Secure Routing in Wireless Sensor Networks via POMDPs

Athirai A. Irissappane[1], Jie Zhang[1], Frans A. Oliehoek[2], Partha S Dutta[3]
[1]Nanyang Technological University, Singapore, [3]Rio Tinto, Singapore
[2]University of Liverpool, UK, University of Amsterdam, The Netherlands
[1]{athirai001,zhangj}@ntu.edu.sg, [2]frans.oliehoek@liverpool.ac.uk, [3]partha.dutta@riotinto.com

Abstract

Wireless sensor networks are being increasingly used for sustainable development. The task of routing in these resource-constraint networks is particularly challenging as they operate over prolonged deployment periods, necessitating optimal use of their resources. Moreover, due to the deployment in unattended environments, they become an easy target for attackers. In this paper, we propose a hierarchical POMDP based approach to make routing decisions with partial/limited information about the sensor nodes, in a secure and energy-efficient manner. We demonstrate in a large-scale simulation that the approach provides a better energy/packet delivery tradeoff than competing methods, and also validate these conclusions in a real-world testbed.

1 Introduction

Sustainable development involves detailed analysis, design and modeling of complex systems used in making decisions about managing natural resources. Gathering data at minimal cost is important to study these complex systems for which Wireless Sensor networks (WSNs) are used [Dyo *et al.*, 2010]. WSNs are composed of spatially distributed sensor nodes to cooperatively monitor environmental changes over time. Sensors sense the data and transmit it to the sink (gateway between sensor nodes and end users) through multihop routing. A key challenge in sustainable development is that the (resource-constraint [Mac Ruairí 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 strong security provisioning, as the unreliable wireless channels and unattended operations make it very easy to compromise/capture the nodes. One such scenario is the monitoring of oil/chemical leaks from an industry in, say a nearby river. The monitors (government) will want to assess the true impact of the leaks and the industry might have an incentive to corrupt the measurements.

The use of trust systems for defending against (internal) security attacks and performing routing has been extensively studied in literature [Román *et al.*, 2009; Yu *et al.*, 2012]. Internal attackers [Liu *et al.*, 2007] are malicious nodes within the network, who can easily bypass traditional cryptographic mechanisms. Trust schemes can identify such nodes, as they can predict a node's behavior (quality) both directly, via evaluation based on its past actions, and indirectly, using recommendations (opinions) from other nodes. However, many trust schemes cannot effectively handle attacks targeting trust systems themselves [Sun *et al.*, 2006] i.e., they are heavily affected by malicious nodes deliberately providing misleading opinions (unfair ratings) about other nodes. Also, existing trust schemes fail to address the energy constraints of WSNs, as they focus on accurately determining node quality but ignore the overhead of such computation by relentlessly seeking recommendations from all the other sensor nodes.

This paper presents a Partially Observable Markov Decision Process (POMDP) based trust scheme [Irissappane *et al.*, 2014] to simultaneously address security issues and energy constraints while routing in WSNs. The POMDP model [Kaelbling *et al.*, 1998] provides a principled approach for decision making under uncertainty, making it an ideal choice for nodes in WSNs that need to choose a suitable next-hop neighbor to route packets with only limited information. 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, we model the routing problem for each node using a *hierarchical* POMDP (called *Secure Routing* POMDP (SRP)). We also employ *factored* representations 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.

2 Related Work

RFSN [Ganeriwal *et al.*, 2008] determines the quality of a node using the Beta distribution on the co-operation information collected from a watchdog [Marti *et al.*, 2000] mechanism as well as from recommendations given by other nodes. TARP [Rezgui and Eltoweissy, 2007] avoids routing through malicious nodes by assessing their forwarding ratio using both direct evaluation 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. CONFIDANT [Buchegger and Le Boudec, 2002] uses 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. [Nurmi, 2007] proposes a POMDP based routing scheme that estimates its state composed of neighbor nodes' local parameters (selfishness and energy). However, it uses gradient techniques to determine policies which (as we show empirically), can be far from optimal. Also, it does not use recommendation information from other sensor nodes.

Hierarchical POMDP based approaches have been studied in literature to harness the curse of dimensionality and history in solving POMDPs [Zhang and Sridharan, 2012; Pineau and Thrun, 2002; Theocharous, 2002; Foka and Trahanias, 2007], using 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

We consider a network $N = \{n_i | i = 1 \cdots |N|\}$ of sensor nodes. The neighborhood of each node n_i consists of sensors reachable within the transmission radius r. Every node independently optimizes its routing behavior and chooses a next-hop neighbor (using the SRP model described in Sec. 5) to route packets to the sink. The following paragraphs describe the important aspects involved in this decision problem.

Fitness (for routing purposes) For a node n_i, a next-hop neighbor n_j is chosen based on its fitness ($f_j \in \{good, bad\}$) in routing packets, calculated using the fitness factors: residual-energy, distance and routing behavior.

Residual-Energy: $f.e_j \in \{high, low\}$ denotes the remaining energy in n_j, to route packets. We use [Heinzelman *et al.*, 2000] to determine n_j's actual energy $e(n_j)$ and then discretize[1] it (to use standard POMDP solvers): $f.e_j$=$high$, if $e(n_j)$ is greater than half its initial value and low, otherwise.

Distance: Distance $D(n_j, sink)$ of n_j from the sink is determined by broadcasting HELLO $\langle source, hopCount \rangle$ messages. Initially, $\langle source = sink, hopCount = 0 \rangle$ is broadcast from the sink. The neighboring nodes of the sink receive this message and determine their distance by incrementing $hopCount$. The new $hopCount$ is then re-broadcast to each node's neighbors. To discretize[1] the distance values, for node n_j, $f.d_j$=$near$, if $D(n_j, sink) < D(n_i, sink)$, else $f.d_j$=far.

[1]Though a larger number of behaviors is possible, in order to find a good POMDP strategy it may suffice to consider a moderate number of values (see 'robustness' experiments in Fig. 5(a)).

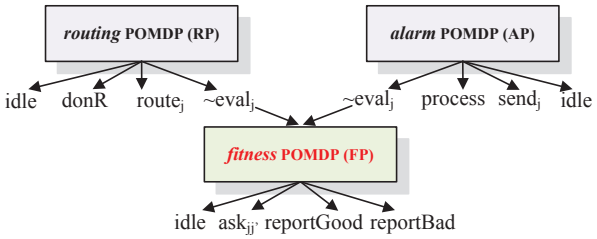


Figure 1: Secure Routing POMDP (SRP) action hierarchy.

Routing Behavior: Node n_j can forward the packets sent to it i.e., $f.rb_j$=$forward$, or drop packets $f.rb_j$=$drop$, exhibiting network based attacks [Karlof and Wagner, 2003], such as: 1) black-hole attack, where node n_j drops packets with probability p_d=1, always; 2) on-off attack, where p_d=1 only during specific intervals of time, and p_d=0 otherwise, etc.

Message Protocols To determine the above fitness factors of f_j, node n_i can request opinions (query action) about node n_j from another neighbor $n_{j'}$ using a QUERY $\langle n_i, n_{j'}, n_j \rangle$ message[2] . A REPLY $\langle n_{j'}, n_i, n_j, f.e_j, f.d_j, f.rb_j \rangle$ message is then sent by $n_{j'}$ about the fitness factors of n_j. Node n_i can also opt to route packets (route action) to n_j, to determine its fitness factors. Once n_i routes packets to n_j, n_j sends an acknowledgement ACK $\langle n_j, n_i, f.e_j, f.d_j \rangle$ message[2], informing n_i about its residual-energy and distance values. Node n_i also employs a watchdog mechanism [Marti *et al.*, 2000] to peek n_j's transmission packets and monitor its routing behavior $f.rb_j$, whether it actually forwards/drops the sent packets. Thus, the actual values of $f.e_j, f.d_j$ and $f.rb_j$ can be determined by routing packets to n_j. Node n_i can also send/receive ALARM $\langle n_i, n_j, f.e_j, f.d_j, f.rb_j \rangle$ messages, carrying information about a malicious node n_j.

Unfair Ratings When sending a REPLY to a QUERY message, node $n_{j'}$ can be unfair by providing misleading opinions about n_j. We use a variable $r_{j'}$ to denote the trustworthiness of $n_{j'}$ in its rating behavior, when providing opinions about other nodes. $n_{j'}$ can be truthful ($r_{j'}$=$true$) or provide unfair ratings about n_j, exhibiting any of the following attacks [Jiang *et al.*, 2013]: 1) random ($r_{j'}$=$rand$), where $n_{j'}$ randomly provides fair and unfair ratings; 2) adversarial ($r_{j'}$=adv), where $n_{j'}$ always provides unfair ratings; 3) camouflage ($r_{j'}$=cam), where $n_{j'}$ is honest in the beginning and provides unfair ratings after ϕ packet transmissions; 4) collusive-unfair ($r_{j'}$=$coll$), where attackers form the majority in the system and always promote their neighbors.

Overall Goal Given that n_i can use query and route actions to determine the fitness factors of its neighbors, 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, we adopt a Partially Observable Markov Decision Process (POMDP) model, which selectively queries for information

[2]In case of no response (REPLY/ACK) from a node, it can be queried again max times, after which it will be deemed malicious.

to select a suitable next-hop neighbor to successfully route packets (and send alarms, if necessary), thereby minimizing energy consumption and maximizing lifetime of the sensor nodes. Sec. 4 gives a brief description of a POMDP model.

4 POMDP

A POMDP [Kaelbling *et al.*, 1998] can be described by the tuple $\langle \mathcal{S}, \mathcal{A}, T, R, \Omega, O \rangle$: given that the environment has a state $s \in \mathcal{S}$, the POMDP agent takes some action $a \in \mathcal{A}$, causing a state transition from s to s', using T, the transition model that specifies probabilities $\Pr(s'|s,a)$. The agent also receives observations ($o \in \Omega$) based on the observation model O, specifying the probabilities $\Pr(o|a,s')$. For a transition, the agent receives a reward $R(s,a,s')$. We also assume an infinite horizon problem. The POMDP agent maintains a *belief* $b \in \mathcal{B}$, i.e., a probability distribution over states via Bayes' rule. If $b(s)$ specifies the probability of s ($\forall\ s$), the updated belief b' after taking action a and receiving observation o is given by,

$$b'(s') = \frac{\Pr(s', o|b, a)}{\Pr(o|b, a)} = \frac{\Pr(o|a, s')}{\Pr(o|b, a)} \sum_s Pr(s'|s, a)b(s) \quad (1)$$

A POMDP policy maps beliefs to actions and is associated with a value function $V_\pi(b)$ that specifies the expected total reward of executing policy π starting from b. The objective of a POMDP agent is to find an optimal POMDP policy π^*, which maximizes the expected total reward. Unfortunately, the routing problem in WSNs is too large to be modeled as a single POMDP (finding the optimal policy is intractable, PSPACE complete). As factored solvers [Poupart, 2005] do not scale sufficiently to large problems and online planning algorithms cannot be applied to the energy-constraint WSNs, we propose an hierarchical approach, while still using factored solvers to solve the individual hierarchical components.

5 Secure Routing POMDP (SRP)

Here, we propose a hierarchical formulation (as shown in Fig. 1) for the secure routing problem, exploiting the fact that this problem admits a natural decomposition in sub-tasks. In particular, at the higher level, we select a next-hop neighbor based on its fitness, performed by the *routing POMDP (RP)*. At the lower level, we evaluate the fitness of the nodes based on more detailed observations and trust propagation mechanisms, handled by the *fitness POMDP (FP)*. Additionally, we investigate whether adding functionality to raise alarms about potentially unfit neighbors can improve the overall WSN performance. This alarm functionality is isolated in a separate component: the *alarm POMDP (AP)*.

The hierarchy functions as follows: every time a node receives a packet to route, an episode of RP is activated. This episode can trigger multiple episodes of FP by using the 'evaluate' ($eval_j$) actions to evaluate the fitness of its neighbors. As such, each episode of FP has as its goal to provide an estimation of the fitness of a specific node (say n_j). To this end, FP can ask different nodes about the various fitness factors of n_j (implemented using QUERY messages, described in Sec. 3). When FP is certain enough, it reports that n_j is 'good' (g) or 'bad' (b), thereby ending the FP episode. At this point, control is transferred back to RP which receives the report (g or b) as an observation, using which it updates its high-level belief about the fitness of the node. Each episode of RP ends by selecting a neighbor to route packets.

The AP (if present) is activated after each packet is handled by RP. The goal of AP is to send alarms about nodes that seem unfit (if no alarm has been sent about them yet). For this, AP can also call FP to perform node evaluations. AP is also activated when an alarm is received from one neighbor about another neighbor. In this case, AP should decide whether or not to process the alarm [Buchegger and Le Boudec, 2002], which may/may not lead to a belief update.

5.1 Routing POMDP (RP)

The main goal of RP is to determine a fit next-hop neighbor to route packets. The states, actions, observations and rewards for RP are described in Table 1. RP maintains a belief over the overall fitness f_j of each neighboring node (n_j) i.e., $f_j = good(g)$, if n_j is fit to route packets and $f_j = bad(b)$, if n_j is unfit. To determine f_j, it uses the evaluate action $\sim eval_j$ and decides a next-hop neighbor n_j using $route_j$ action. The $donR$ (do not route to any neighboring node) action is taken when no nodes are fit to route packets. There is a cost associated with the $\sim eval_j$ action. We assume that selecting an unfit neighbor to route packets will fail to transmit packets to the sink and thus a penalty $R(f'_j{=}b, route_j)$ is levied. Similarly, a reward $R(f'_j{=}g, route_j)$ is given for selecting a fit node which can successfully transmit packets to the sink. We can also see that $\sim eval_j$ is an abstract action which in turn calls FP (action hierarchy for SRP is shown in Fig. 1).

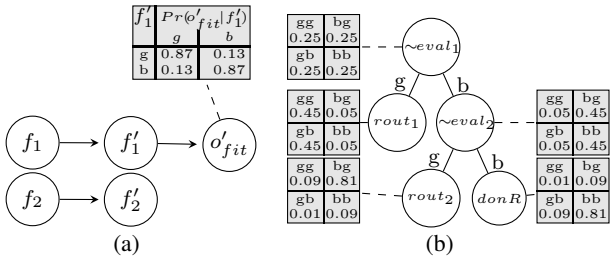


Figure 2: RP: (a) DBN for $\sim eval_1$ action; (b) (Partial) Policy.

To specify the transition and observation functions, we describe the model for a two-node neighborhood ($\in \{n_1, n_2\}$). We follow the factored representation [Poupart, 2005], which uses Dynamic Bayesian Networks (DBNs) with Conditional Probability Tables (CPTs) to compactly represent the state spaces. The transition and observation probabilities for the $\sim eval_{j=1}$ action is shown in Fig. 2(a). Here, transitions are static and observation o'_{fit} depends on fitness f'_1 of n_1. The transition and observation probabilities for the higher level POMDPs (RP, AP) can be learned from the behavior of FP (similar to layered learning [Stone and Veloso, 1998]). Thus, probabilities for o'_{fit} in Fig. 2(a), are learned based on the policy of FP (by conducting experiments as described in Sec. 6).

To briefly illustrate the belief updating process in RP, Fig. 2(b) shows a (partial) routing policy for the two-node ($\in \{n_1, n_2\}$) neighborhood. The beliefs prior to taking the

POMDP	States	Observations	Actions	Rewards
routing	$f_j \in \{g, b\}$	$o_{fit} \in \{g, b\}$	$\sim eval_j$	$R(\sim eval_j) = -10$
			$route_j$	$R(f'_j = g, route_j) = 150$
			$donR$	$R(f'_j = b, route_j) = -200$
			$idle$	$R(idle) = -3$
alarm	$f_j \in \{g, b\}$	$o_{fit} \in \{g, b\}$	$\sim eval_j$	$R(\sim eval_j) = -10$
	$alarmFrom \in \{1, \cdots k\}$	$o_{from} \in \{1, \cdots k\}$	$process$	$R(f'_{from} = g/b, process) = 50/-100$
	$alarmAbout \in \{1, \cdots k\}$	$o_{about} \in \{1, \cdots k\}$	$send_j$	$R(f'_j = b/g, send_j) = 50/-100$
	$alarmSent_j \in \{yes, no\}$	$o_{sent} \in \{1, \cdots k\}$	$idle$	$R(idle) = -3$
fitness	$f.e_j \in \{high, low\}$	$o_{ener} \in \{high, low\}$	$ask_{jj'}$	$R(ask_{jj'}) = -10$
	$f.d_j \in \{near, far\}$	$o_{dist} \in \{near, far\}$	$reportGood$	$R(f'_{node} = g, reportGood) = 150$
	$f.rb_j \in \{forward, drop\}$	$o_{rb} \in \{forward, drop\}$	$reportBad$	$R(f'_{node} = b, reportGood) = -200$
	$r_j \in \{true, rand, adv, cam_{sleep}, cam_{act}, coll\}$	$o_r \in \{true, lie\}$		$R(f'_{node} = bad, reportBad) = 150$
	$evaluation\_node \in \{1, \cdots k\}$	$o_{node} \in \{1, \cdots k\}$		$R(f'_{node} = g, reportBad) = -200$
	$f_j \in \{g, b\}$		$idle$	$R(idle) = -3$

$^*rand - random, adv - adversarial, cam - camouflage, act - active, coll - collusive\text{-}unfair$

Table 1: State, action, observation variables, rewards specified in SRP. Variable j is a vector of k=no. of neighbors elements.

actions (represented by nodes in Fig. 2(b)) are shown using tables associated with the nodes. Here, we present the marginal beliefs on the fitness of nodes $f_1, f_2 \in \{g, b\}$. As a sample state, gg represents that both n_1 and n_2 are fit to route packets. Initially, RP assumes uniform quality levels for f_1, f_2 (0.5 g, 0.5 b). On taking action $\sim eval_1$, when observation g is received (traversing through left child of the tree), beliefs are updated (using Eqn. 1) such that more weights are given to states where f_1 is good ($gg = 0.45$, $gb = 0.45$) and $route_1$ action is taken. The beliefs when observation is b, are shown using right children of the tree, resulting in $donR$ action.

5.2 Alarm POMDP (AP)

There are two modes in which AP can operate: 1) 'periodic' mode after every packet transmission by node n_i, to send alarms about malicious (unfit) nodes; 2) 'triggered' mode after receiving an incoming alarm, to decide whether to process alarms received from other nodes. AP maintains beliefs over the fitness $f_j \in \{g, b\}$ of each neighboring node and uses the $\sim eval_j$ action (as shown in Table 1) to determine them. It can broadcast alarms about a malicious node n_j using the $send_j$ action. To avoid re-broadcasting the same alarms, the alarms sent are tracked using $alarmSent_j \in \{yes, no\}$ variable. When AP is triggered on receiving an alarm from $alarmFrom$ about node $alarmAbout$, it decides either to $process$ these alarms, leading to a subsequent belief update about the fitness of node $alarmAbout$ or simply ignore them. There is a reward/penalty $R(f'_j$=$b/g, send_j)$ for broadcasting truthful/false alarms about n_j. There is also a reward/penalty $R(f'_{from}$=$g/b, process)$ for processing truthful/false alarms from good/bad nodes, respectively.

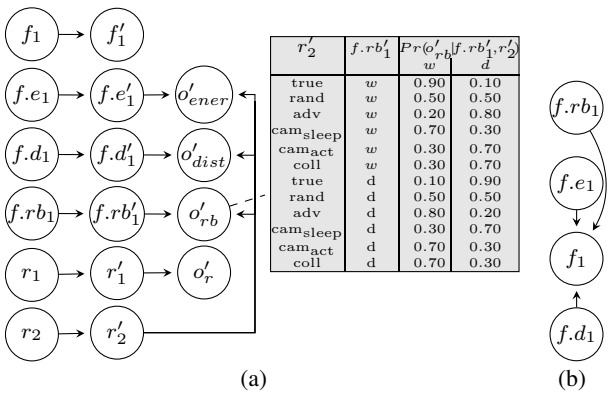
The DBN and CPT[3] for $\sim eval_j$ action are similar to Fig. 2(a). For $process$ action, observation o'_{fit} depends on the fitness of nodes $alarmFrom$ and $alarmAbout$. To deal with both 'periodic' and 'triggered' functionalities in the same POMDP, we introduce an auxiliary (dummy) first time step in AP: the initial belief is an uniform random distribution over the $alarmFrom$, $alarmAbout$, $alarmSent_j$ variables, but during the second time step, the agent receives perfect observations about these factors, leading to beliefs that only have positive probabilities for the states that are consistent with the required functionality. Transition for $alarmSent_j$ is such that $alarmSent_j$=yes on sending an alarm about n_j. We also add some randomness to model variable delay forgetting.

5.3 Fitness POMDP (FP)

The main goal of FP is to evaluate the fitness of a specific node, and report it to the higher level POMDPs. Apart from maintaining beliefs on the fitness of each neighbor n_j i.e., $f_j \in \{g, b\}$, FP also maintains beliefs over the individual factors that make up f_j, i.e., residual-energy $f.e_j$, distance $f.d_j$, and routing behavior $f.rb_j$. The rating behavior of n_j in providing opinions is also maintained using variable r_j. Whenever the $\sim eval_j$ action is taken by RP or AP, FP is activated. To determine fitness of n_j ($evaluation\_node$ in Table 1), FP can query $n_{j'}$ using $ask_{jj'}$ action, which translates to QUERY and REPLY messages (see Sec. 3). The $reportGood$/$reportBad$ action is then taken, determining fitness of n_j to be g/b, respectively, which are then reported as observations to update the high-level beliefs about the fitness of node n_j in RP or AP. There is a reward/penalty for correctly/incorrectly determining the node's quality.

Fig. 3(a) shows the DBN and CPT of the ask_{12} (query n_2 about fitness of n_1) action. Apart from the state variables for node n_1, the DBN also contains a variable r_2, which denotes the rating behavior i.e., trustworthiness of n_2 in providing truthful opinions about n_1. The observation probabilities for $o'_{rb} \in \{forward(w), drop(d)\}$ depend on the routing behavior $f.rb'_1$ of n_1 and rating behavior r'_2 of n_2. o'_{ener} and o'_{dist} have similar probability values[3] as o'_{rb}. The observation probabilities for o'_{rb} encode that asking a trustworthy node ($r'_2 = true$) gives more accurate observations (with 90% probability) than untrustworthy nodes. Specifically, $random$, $adversarial$ nodes provide unfair ratings with 50%, 80% probability, respectively. $camouflage$ nodes, are honest in the beginning, then provide unfair ratings with 70% probability. To model such change in behavior, we consider two state variables i.e., cam_{sleep} (to represent the honest phase) and $cam_{act(ive)}$ (to represent the unfair rating phase), such that, nodes in the cam_{sleep} state can transition to cam_{act} state with a 50% probability. $collusive{-}unfair$ are groups of nodes, providing unfair ratings with 70% probability. Though the above probabilities may not represent the true node behaviors, they are still effective in identifying the unfair raters as

[3]Figures are not shown due to space constraints.



r'_2	$f.rb'_1$	$Pr(o'_{rb}\|f.rb'_1,r'_2)$ w	d
true	w	0.90	0.10
rand	w	0.50	0.50
adv	w	0.20	0.80
$\text{cam}_{\text{sleep}}$	w	0.70	0.30
cam_{act}	w	0.30	0.70
coll	w	0.30	0.70
true	d	0.10	0.90
rand	d	0.50	0.50
adv	d	0.80	0.20
$\text{cam}_{\text{sleep}}$	d	0.30	0.70
cam_{act}	d	0.70	0.30
coll	d	0.70	0.30

Figure 3: (a) DBN for ask_{12} action of FP. State variables without a CPT are 'static', preserving the previous value with probability 1; (b) Initial state distribution for fitness f_1.

demonstrated by our experiments in Sec. 6.

However, in Fig. 3(a), there is no observation for the fitness f_1 of n_1. Instead, we specify in the initial state distribution of FP, how the different fitness factors ($f.e_1, f.d_1, f.rb_1$) contribute to f_1 (Fig. 3(b), Eqn. 2). The specific probabilities are given in Eqn. 3 and 4, where N_{good} is the number of positive fitness factors i.e., number of times $f.e_1$=$high$, $f.d_1$=$near$, $f.rb_1$=$forward$ and N_{bad} is the number of negative fitness factors i.e., number of times $f.e_1$=low, $f.d_1$=far, $f.rb_1$=$drop$. Thus, even though we do not get direct observations about f_1, now it will be updated as part of the overall belief update. $reportGood$, $reportBad$ actions merely notify about fitness of a node and do not receive any observations.

$$b^0(s)=b^0(f_1|f.e_1,f.d_1,f.rb_1)\, b^0(f.e_1)\, b^0(f.d_1)\, b^0(f.rb_1)\cdots \quad (2)$$

$$b^0(f_1 = g|f.e_1, f.d_1, f.rb_1) = \frac{N_{good}}{N_{good} + N_{bad}} \quad (3)$$

$$b^0(f_1 = b|f.e_1, f.d_1, f.rb_1) = \frac{N_{bad}}{N_{good} + N_{bad}} \quad (4)$$

The belief updating process[3] in FP follows a similar approach as that of RP (shown in Fig. 2(b)).

5.4 Parallel Belief Update

The belief update takes place separately for RP, AP and FP. For a routing task, RP and FP are active i.e., when RP takes the $\sim eval_j$ action, FP is called and based on the $reportGood/reportBad$ action of FP, observation $good/bad$ is received by RP. However, in this case the beliefs about n_j are updated only in RP and left outdated in AP. To update the knowledge about n_j even in AP, we introduce the $idle$ action (in each POMDP). Whenever the $\sim eval_j$ action is called in RP and beliefs are updated about n_j, the $idle$ action of AP is also called to update the beliefs on n_j. Similarly, when $\sim eval_j$ action is called by AP, $idle$ action is taken in RP.

Whenever the actual fitness factors i.e., $f.e_j$, $f.d_j$ and $f.rb_j$ of n_j are determined after the routing process (as described in Sec. 3), the $idle$ action is taken by FP to update the beliefs about n_j, 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_j.

6 Performance Evaluation

We conduct experiments in a simulated environment as well as a real-world testbed and compare the performance of SRP with RFSN [Ganeriwal *et al.*, 2008], CONFIDANT [Buchegger and Le Boudec, 2002] and Nurmi [Nurmi, 2007]. To show the usefulness of AP, we compare the results of SRP with and without AP denoted by SRP and SRP-NAP, respectively. To verify the usefulness of the hierarchical structure, we implemented SRP 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. We measure 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.

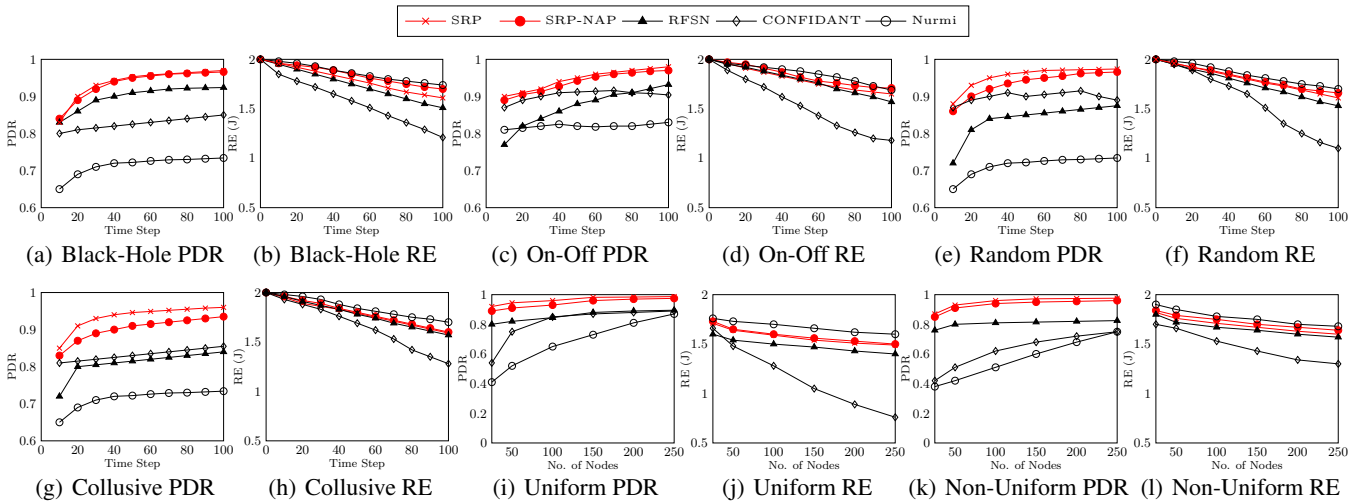
To learn the observation probabilities for $\sim eval_j$ action of RP (AP) based on the policy of FP (for which we manually specify the probabilities), we use a maximum likelihood approach. Using (offline) simulations, we run SRP and randomly select 500 actions of RP (AP) which invoke FP. We measure the number of instances where FP correctly identifies a node's quality and determine the probability of receiving a correct/incorrect observation for $\sim eval_j$ action to be $0.87/0.13$, respectively. Also, the RP, AP and FP policies are computed (using Symbolic Perseus [Poupart, 2005]) offline, assuming a maximum neighborhood size M[4].

For simulation, we use the SWANS Simulator [Barr *et al.*, 2005]. We consider 100 stationary nodes, uniformly randomly distributed within a $1000m \times 1000m$ square, with the sink at its right end. The transmission radius is $100m$ and M=5. 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 $100\ pJ/bit/m^2$ for the transmitter amplifier. We consider 20% of the nodes 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. 4(a), under black-hole attack, SRP and SRP-NAP achieve the highest PDR (97% after 100 time steps). SRP performs slightly better than SRP-NAP, especially in the beginning, as it identifies malicious nodes earlier by propagating alarms. SRP, SRP-NAP, 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. 4(b), SRP obtains a lower residual-energy ($1.61J$) than SRP-NAP ($1.70J$), 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. 4(c-d) show similar results, where on-off attackers drop packets every 5 time steps.

In Fig. 4(e-h), the 20% compromised nodes (black-hole attackers) also target the trust system by providing unfair rat-

[4]Non-existent nodes will be considered malicious.



(a) Black-Hole PDR (b) Black-Hole RE (c) On-Off PDR (d) On-Off RE (e) Random PDR (f) Random RE

(g) Collusive PDR (h) Collusive RE (i) Uniform PDR (j) Uniform RE (k) Non-Uniform PDR (l) Non-Uniform RE

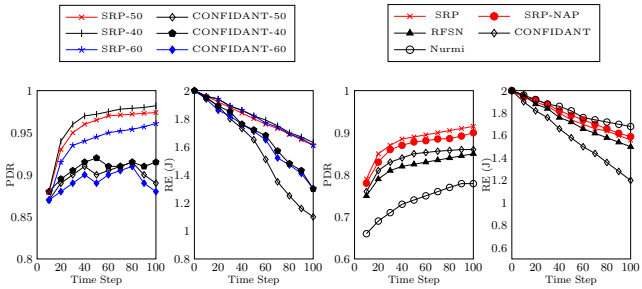
Figure 4: Performance of the routing schemes in terms of Packet Delivery Ratio (PDR) and Residual-Energy (RE), in a simulated environment: (a-h) different attacking scenarios; (i-l) different load characteristics.

ings (showing $random$, $collusive$-$unfair$ behavior)[5]. SRP and SRP-NAP can effectively identify unfair raters as they model such behaviors as a part of their POMDP states. In Fig. 4(e-f), under random attack, SRP, SRP-NAP achieve high performance. In Fig. 4(g-h), under collusive-unfair attack, unfair raters are increased to 60%, forming the majority. SRP (PDR 96%) performs better than SRP-NAP as it easily identifies attackers by propagating alarms, while SRP-NAP (PDR 93%) initially obtains misleading opinions from the colluders, thereby routing through malicious nodes, until their actual behavior is identified after routing. Further Fig. 4(a-h) also show that AP indeed improves the performance of SRP (PDR of SRP is always greater than SRP-NAP, though AP involves additional energy drain, in some cases).

Fig. 4(i-l) show the results (under collusive-unfair attack[5]), when network environment changes. SRP performs better under uniform load (λ=1 per node) as well as under non-uniform load ($\lambda \in [0, 1]$ 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. Since most probabilities in SRP are manually specified (e.g., $random$ nodes provide unfair ratings with probability p_u=50%), we analyze the robustness of SRP to the specification of such values when the actual behaviors of $random$ nodes change: 1) SRP-60, where p_u=60% instead of 50% (as assumed in FP); 2) SRP-40, where p_u=40%; 3) SRP-50 for perfectly random nodes with p_u=50%. Fig. 5(a) show that even when interacting with advisors that act differently than those assumed in FP, performance of SRP is robust (relatively better than its CONFIDANT counterpart).

We also conduct experiments on a real testbed, consisting of 10 sensor nodes placed randomly in a 5×10 m^2 in-door space. Each node is built on an Arduino Due board, with a XBee- Pro RF Module for radio transmission, a SHT21 Digital Sensor Module for temperature sensing and a TOL-10617 LiPo Fuel Gauge for measuring energy. The neighborhood radius is $5m$, size of packets is 62 $bytes$, initial node energy is $2J$ and λ=1. Fig. 5(b), under collusive-unfair attack[5], show that SRP (PDR 91.5%) outperforms other schemes.

[5]Experiments on other attacks exhibit similar conclusion and are not shown here due to space constraints.



(a) Random (b) Collusive

Figure 5: Experiments on: (a) Robustness; (b) Real testbed.

7 Conclusion and Future Work

We present the Secure Routing POMDP (SRP) approach, to select suitable next-hop neighbors and successfully route packets to the sink. SRP 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. We model SRP using hierarchical and factored representations to address the complexity in solving POMDPs. Experiments show that SRP consistently achieves higher packet delivery rates by coping with various attacks, while still maintaining high residual-energy. Thus, we guarantee reliable, energy-efficient routing in WSNs, which are key factors in sustainable development.

While we established that SRP is robust against the choice of parameters for transition and observation models, an inter-

esting direction of future work is to automatically optimize these. We will also investigate using finite-state controllers, which can be more energy-efficient [Grześ *et al.*, 2013].

Acknowledgments

This work is supported by the A*STAR SERC grant (1224104047) awarded to Dr. Jie Zhang, NWO Innovational Research Incentives Scheme Veni #639.021.336 awarded to Dr. Frans A. Oliehoek and the Institute for Media Innovation at Nanyang Technological University.

References

[Barr *et al.*, 2005] Rimon Barr, Zygmunt J Haas, and Robbert Van Renesse. 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, 2005.

[Buchegger and Le Boudec, 2002] Sonja Buchegger and Jean-Yves Le Boudec. Performance analysis of the CONFIDANT protocol (Cooperation of nodes: Fairness in dynamic ad-hoc networks). In *MobiHoc*, 2002.

[Dyo *et al.*, 2010] Vladimir Dyo, Stephen A Ellwood, David W Macdonald, Andrew Markham, Cecilia Mascolo, Bence Pásztor, Salvatore Scellato, Niki Trigoni, Ricklef Wohlers, and Kharsim Yousef. Evolution and sustainability of a wildlife monitoring sensor network. In *SenSys*, 2010.

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

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

[Grześ *et al.*, 2013] Marek Grześ, Pascal Poupart, and Jesse Hoey. Controller compilation and compression for resource constrained applications. *Algorithmic Decision Theory*, pages 193–207, 2013.

[Heinzelman *et al.*, 2000] Wendi Rabiner Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *HICSS*, 2000.

[Irissappane *et al.*, 2014] Athirai A Irissappane, Frans A Oliehoek, and Jie Zhang. A POMDP based approach to optimally select sellers in electronic marketplaces. In *AAMAS*, 2014.

[Jiang *et al.*, 2013] Siwei Jiang, Jie Zhang, and Yew-Soon Ong. An evolutionary model for constructing robust trust networks. In *AAMAS*, 2013.

[Kaelbling *et al.*, 1998] Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1):99–134, 1998.

[Karlof and Wagner, 2003] Chris Karlof and David Wagner. Secure routing in wireless sensor networks: Attacks and countermeasures. *Ad hoc networks*, 1(2):293–315, 2003.

[Liu *et al.*, 2007] Fang Liu, Xiuzhen Cheng, and Dechang Chen. Insider attacker detection in wireless sensor networks. In *INFOCOM*, 2007.

[Mac Ruairí and Keane, 2007] Rónán Mac Ruairí and Mark T Keane. An energy-efficient, multi-agent sensor network for detecting diffuse events. In *IJCAI*, 2007.

[Marti *et al.*, 2000] Sergio Marti, Thomas J Giuli, Kevin Lai, and Mary Baker. Mitigating routing misbehavior in mobile ad hoc networks. In *MobiCom*, 2000.

[Nurmi, 2007] Petteri Nurmi. Reinforcement learning for routing in ad hoc networks. In *WiOpt*, 2007.

[Pineau and Thrun, 2002] Joelle Pineau and Sebastian Thrun. An integrated approach to hierarchy and abstraction for POMDPs. *Carnegie Mellon University Technical Report CMU-RI-TR-02-21*, 2002.

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

[Rezgui and Eltoweissy, 2007] Abdelmounaam Rezgui and Mohamed Eltoweissy. Tarp: A trust-aware routing protocol for sensor-actuator networks. In *MASS*, 2007.

[Román *et al.*, 2009] Rodrigo Román, Carmen Fernandez-Gago, Javier López, and Hsiao Hwa Chen. Trust and reputation systems for wireless sensor networks. *Security and Privacy in Mobile and Wireless Networking*, pages 105–128, 2009.

[Ross *et al.*, 2008] Stéphane Ross, Joelle Pineau, Sébastien Paquet, and Brahim Chaib-draa. Online planning algorithms for POMDPs. *Journal of Artificial Intelligence Research*, 32(1):663–704, 2008.

[Stone and Veloso, 1998] Peter Stone and Manuela Veloso. Layered approach to learning client behaviors in the robocup soccer server. *Applied Artificial Intelligence*, 12(2-3):165–188, 1998.

[Sun *et al.*, 2006] Yan Lindsay Sun, Zhu Han, Wei Yu, and KJ Ray Liu. A trust evaluation framework in distributed networks: Vulnerability analysis and defense against attacks. In *INFOCOM*, 2006.

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

[Yu *et al.*, 2012] Yanli Yu, Keqiu Li, Wanlei Zhou, and Ping Li. Trust mechanisms in wireless sensor networks: Attack analysis and countermeasures. *Journal of Network and Computer Applications*, 35(3):867–880, 2012.

[Zhang and Sridharan, 2012] Shiqi Zhang and Mohan Sridharan. Active visual sensing and collaboration on mobile robots using hierarchical POMDPs. In *AAMAS*, 2012.