Distributed Reputation-based Secure Localization in Sensor Networks

Avinash Srinivasan, Jie Wu, and Joshua Teitelbaum
Department of Computer Science and Engineering
Florida Atlantic University
Boca Raton, FL 33431
Email: {asriniva@, jie@cse., jteitel2@}fau.edu

***Abstract*— Wireless Sensor Networks (WSNs) have critical applications in diverse domains like environmental monitoring and military operations like target tracking and land-mine detection where accurate location of sensors is vital. One common method of localization in literature uses a set of specialty nodes known as beacon nodes (BNs) that assist other sensor nodes (SNs) to determine their location. In BN-based models, it is critical that malicious BNs be discouraged from providing false location information. This paper proposes a novel reputation-based scheme called Distributed Reputation-based Beacon Trust System (DRBTS) for excluding malicious BNs to ensure secure localization of SNs. To the best of our knowledge, DRBTS is the first localization model to use the concept of reputation for excluding malicious BNs. In DRBTS, every BN monitors its 1-hop neighborhood for misbehaving BNs and accordingly updates the reputation of the corresponding BN in the Neighbor-Reputation-Table (NRT). Each BN then publishes its NRT in its 1-hop neighborhood which the remaining BNs use as second-hand information, after it qualifies a deviation test, for updating the reputation of their neighbors. On the otherhand, SNs use the information in NRT to determine whether or not to use a given beacon's location information based on a simple majority voting scheme.**

***Index Terms*— Localization, quorum, reputation, security, sensor networks, trust.**

I. Introduction

The core function of wireless sensor networks (WSNs) is to detect and report events. Many protocols have been devised over the last few years to enable the location discovery process to function independent of GPS and other manual techniques [2], [3], [4], [5], [7], [8], [9]. In all these literatures, the focal point of location discovery has been a set of specialty nodes known as beacon nodes (BNs). These BNs are capable of determining their location and then providing this information to other sensor nodes (SNs) lacking this ability. Figure 1 shows a WSN with SNs and BNs. BNs are represented as shaded nodes (big circles). SNs, on receiving a sufficient number of location references from their beacon neighbors, determine their location. Any feature of the received beacon signal, like the Angle of Arrival (AoA), Received Signal Strength Indicator (RSSI), Time Difference of Arrival (TDoA), etc may be used by SNs to compute their location [5], [9]. We will not be discussing the actual localization technique used but rather focus on securing the localization process.

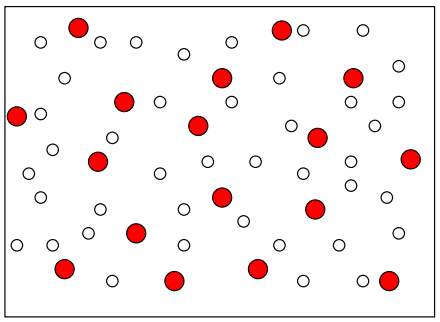


Fig. 1. A network with sensor and beacon nodes. Sensor nodes are represented by hollow circles and beacon nodes are represented by shaded circles.

WSNs are often deployed in unattended or even hostile environments, which would allow an adversary to capture and compromise one or more sensors. This would allow an adversary to launch attacks from within the system, bypassing encryption and password security systems, as the adversary would have access to all the information that the compromised node held. This problem has been extensively studied in wireless networks, but the introduction of BNs creates a new challenge. Most distributed reputation-based systems require that a node be able to interact personally with its neighbors to judge for itself their trustworthiness. Since SNs are not capable

This work was supported in part by NSF grants ANI 0073736, EIA 0130806, CCR 0329741, CNS 0422762, CNS 0434533, CNS 0531410, and CNS 0626240.

of determining their own location, they have no way of determining which BNs are being honest in giving their true location information. This information asymmetry[1] has not been considered by previous works, and as such complicates their implementation in this environment. To solve this problem, we propose a novel Distributed Reputation-based Beacon Trust System (DRBTS).

DRBTS is a distributed security protocol aimed at providing a method by which BNs can monitor each other and provide information so that SNs can choose who to trust, based on a quorum voting approach. In order to trust a BN's information, a sensor must get votes for its trustworthiness from at least half of their common neighbor which is explained in detail in sections IV and V-A. We will show that this allows a sensor to accurately guess the misbehaving/non-misbehaving status of a given BN, given a certain assumption about the level of corruption in the system. We show that our system grows in robustness as node density increases, and show through simulations the effects of different system parameters on robustness. This distributed model not only alleviates the burden on the base station to a great extent, but also minimizes the damage caused by the malicious nodes by enabling sensor nodes to make a decision on which beacon neighbors to trust, on the fly, when computing their location. In summary, our contributions in this paper are as follows

- DRBTS is the first reputation-based approach for secure localization in WSNs.
- We introduce information asymmetry in sensor localization for the first time.
- Simple majority principle has been applied for the first time in sensor localization.
- The proposed model is a distributed approach for sensor localization.
- DRBTS can be easily extended to support beacon based routing.
- We confirm the robustness of our model through simulation and analysis.

The rest of this paper is organized as follows. Section II presents related work. In Section III we give a formal definition of the problem addressed in this paper along with the assumptions made. Section IV discusses the working of our DRBTS model in detail and in Section V we analyze our DRBTS scheme. Section VI discusses the simulation environment and the results. In Section VII we conclude our paper with directions for future work.

[1]When two or more entities involved in a transaction do not have the same amount of information, it is know as information asymmetry.

II. Related Work

Security in sensor networks and mobile ad-hoc networks has become a major focus of research in recent years. In particular, secure localization has been a key research area. Numerous key pre-distribution techniques have been developed [23], [24], [25] to provide efficient key management, a requirement for secure node-to-node communication. Though several researchers have addressed the issue of accurate and efficient localization, very few have addressed it from a security perspective. Savvides, Han, and Srivastava [5] present a novel approach for localization of sensors in an ad-hoc network called AHLoS (Ad-Hoc Localization System) that enables SNs to discover their locations using set distributed iterative algorithms. An extension to this was presented in [6]. Lazos and Poovendran [11] have addressed the problem of enabling sensors of WSNs to determine their location in an un-trusted environment and have proposed a range independent localization algorithm called SeRLoc (Secure Range-Independent Localization) which is a distributed algorithm and does not require any communication among sensors. In [12], Sastry, Shankar, and Wagner introduced the concept of secure location verification, and show how it can be used for location-based access control. They have also presented an extremely lightweight protocol, the Echo protocol, which is a simple method for secure location verification.

Two techniques for improving throughput in ad-hoc networks were presented in [13]. One is the *watchdog* that identifies misbehaving nodes and the other is the *pathrater* that helps routing protocols to avoid these misbehaving nodes. The watchdog system has often been used as the prototypical promiscuous monitoring system in subsequent research.

Michiardi and Molva [14] proposed CORE that has a watchdog along with a reputation mechanism to distinguish between subjective, functional and indirect reputation, all of which are weighted to get the combined reputation of a node. Here, nodes exchange only positive reputation information. The authors argue that this prevents the false-negative or badmouthing[2] attacks, but do not address the issue of collusion of malicious nodes to create false praise. Another interesting feature of CORE is that its members have to contribute on a continuing basis to remain trusted. Otherwise, they will find their reputation deteriorating until they are excluded. Buchegger and Boudec [18] have presented CONFIDANT with predetermined trust, and later improved it with adaptive

[2]Badmouthing is an attack in which a malicious node falsely accuses a benign node of misbehavior.

bayesian reputation and trust system and an enhanced passive acknowledge mechanism (PACK) in [19] and [20] respectively.

Mundinger and Boudec [22] have presented a two-dimensional reputation system for protecting the system from liars to ensure cooperation and fairness in mobile ad-hoc networks. This system works based on a simple deviation test, i.e., a node will accept second-hand information only if it does not deviate too much from its own reputation value. Ganeriwal and Srivastava [21] proposed a reputation-based framework for sensor networks where nodes maintain reputation for other nodes and use it to evaluate their trustworthiness. They show that their framework provides a scalable, diverse and a generalized approach for countering all types of misbehavior resulting from malicious and faulty nodes. Like CORE, the authors have chosen only to disseminate positive interactions, in order to block false-negative attacks, but have shown interest in extending their work with a second metric similar to the one used by Mundinger and Boudec. For readers interested in reputation and trust-based systems, Srinivasan et al have presented a detailed discussion on reputation and trust-based systems for ad hoc and sensor networks in [16].

Finally, in [10], Liu, Ning, and Du have presented a suite of techniques that detect malicious beacon signals, identify malicious BNs, avoid false detection, detect replayed beacon signals, and revoke malicious BNs. Their revocation scheme works on the basis of two counters maintained for each BN, namely attack counter A_c and report counter R_c. It is, to the best of our knowledge, the only current work to address the specific BN model with respect to WSNs. Our paper aims to extend [10] by introducing a reputation-based scheme. Table I is used as an index to the acronyms used throughout this paper.

TABLE I
NOTATION INDEX

s_i, b_j	Sensor Node i / Beacon Node j
TBN_{s_i}	Trusted Beacon Neighbor of s_i
NRT_{b_j}	Neighbor Reputation Table of b_j
$R_{i,j}$	Reputation of b_j from b_i's perspective
$N(s_i), N(b_j)$	Beacon Neighbor Set of s_i / b_j
$C(s_i, b_j)$	Common neighbor set of (s_i, b_j)
$NbrHood(s_i, b_j)$	$C(s_i, b_j) \bigcup b_j$
RNG_{s_i}, RNG_{b_j}	Communication Range of s_i / b_j
$CLoc_{b_j}$	Computed Location of b_j
$ALoc_{b_j}$	Actual Location of b_j
$TLoc_{b_j}$	Transmitted Location of b_j
$+ve_{b_j}$	Votes for b_j
$-ve_{b_j}$	Votes against b_j
TH_{LD}	Threshold for Location deviation
TH_{OD}	Threshold for Opinion deviation
TH_{REP}	Trust Threshold for SN to accept $TLoc_{b_j}$
TO_{res}	Timeout for BN response

III. PROBLEM DEFINITION

Formally, the problem addressed in this paper can be stated as, *"Given a network with BNs and SNs, how can one exclude malicious BNs that provide SNs with incorrect location information?"*

A. Assumptions

DRBTS is developed with the following underlying assumptions:

1) We are considering only dense and static networks. A neighborhood is resilient to k malicious BNs only if there are at least $2k + 1$ BNs in that neighborhood.
2) Location information is broadcast to the requesting SN by the BN, unlike [10], where it is unicasted.
3) Location information is not encrypted using a pair-wise key, unlike [10]. We instead assume a network-wide group key for encryption, to allow promiscuous observation in the network, while preventing outsiders from eavesdropping.
4) We assume an ideal environment, such that transmissions are not lost due to collision or background noise. If two nodes are within each others' transmission range, they will always be able to communicate.

IV. REPUTATION

Reputation is the opinion of one entity about another. In an absolute context, it is the trustworthiness of an entity. The foremost difficulty in adapting standard watchdog mechanisms to systems involving location-aware BNs is that the SNs do not have any first-hand experience to compare second-hand information provided to them by the BNs. To put it in another perspective, in the real world, we base our opinion of a person on personal experience first, and then on what we have been told of them. When we have no personal experience available, we must use our trust in a second party to weigh their testimony about a person. Without any assumptions, there is no way to determine who can be trusted. One logical method is to assume that the majority of people are honest. From there, one can use a simple majority[3] principle to determine the truth.

[3]In a neighborhood of n members, when at least $\lfloor \frac{n}{2} \rfloor + 1$ members have the same opinion it is known as simple majority.

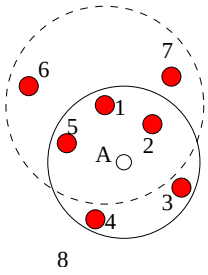


Fig. 2. $N(s_A) = \{b_1, b_2, b_3, b_4, b_5\}$; $N(b_1) = \{b_2, b_5, b_6, b_7\}$; $C(s_A, b_1) = N(s_A) \bigcap N(b_1) = \{b_2, b_5\}$; $NbrHood(s_A, b_1) = C(s_A, b_1) \bigcup b_1 = \{b_1, b_2, b_5\}$

A. Overview

Sensors in DRBTS operate on the aforementioned simple majority principle. Each BN is responsible for monitoring its neighborhood. When a sensor within its range asks for location information, it responds with its location, as do all other beacon nodes within the range of the requesting node. Due to the promiscuity of broadcast transmissions, a BN b_j can overhear the responses of other BNs that are within its communication range. b_j can then compute its location $CLoc_{b_j}$ using this transmitted location $TLoc_{b_i}$ of each BN $b_i \in RNG_{b_j}$ independently. In this paper, however, our primary focus is not on the localization technique. We assume b_j employs one of the several techniques in literature for localization that make use of one or more features of a received beacon signal. b_j then compares each $CLoc_{b_j}$ against its actual location $ALoc_{b_j}$. If the difference is less than a predetermined threshold, TH_{LD} then the corresponding b_i is considered benign and its reputation is increased. Otherwise, the corresponding b_i is considered malicious and its reputation is decreased. Since our main goal is to enforce security such that location information from only benign BNs are accepted, it is necessary to use the BNs as scape goats to do all the computation so that the resource constrained SNs, that actually sense events, gather data, and send reports, can use their energy efficiently solely for the purpose they are deployed.

A decision must be made as to the status of a BN's reputation at time $t = 0$. The system can either assume that all nodes are good nodes until they do something bad, or it can assume that all nodes are bad until they prove themselves as good nodes. The benefit of the former is that the system needs no initial setup time. However, there is a downside to this system. It not only allows, but encourages nodes that have bad reputation to simply spoof a new ID, and re-enter the system with good reputation. This problem can be solved by using the latter technique in which an unknown beacon is untrusted, i.e., whenever a BN hears another BN responding for the first time, it sets the new BN's reputation to 0 before evaluating its transmission. Thus there is no incentive for identity spoofing. The drawback of this system is that no one trusts anyone else at the beginning. Hence, SNs cannot decide with confidence on which beacon locations to include and which to exclude. Our system, DRBTS, assumes the latter untrusted model, but adds a method for BNs to bootstrap reputation. Each BN is given a small number of pseudo-sensor IDs [10]. In periods of low network activity, a BN can use one of these IDs to disguise itself as a SN and request location information, triggering responses from BNs in its neighborhood. This allows reputation to build, even in the absence of network traffic. It should be noted, though, that there will still be a period of uncertainty as the bootstrap mechanism initializes the network to a stable state.

While this bootstrap mechanism allows for network traffic to be created where it is lacking, the reputation values can still take a considerable time to build. The commonly recognized solution to this problem is to allow neighbors to share their experiences. This allows for much more rapid buildup of information, but comes at the expense of security, as it makes the system vulnerable to false praise and slander. Another benefit of sharing information is that it tends to lead to a more consistent local view. In most systems, nodes will publish reputation information to their neighbors at certain time intervals. We have chosen to couple the publishing to dissemination of location information. In this way, the rate at which reputation information is published is directly coupled to the rate at which the reputation changes.

So, when a SN sends out a broadcast requesting location information, each BN will respond with a single broadcast. In this broadcast is both the location it is reporting, and its reputation values for each of its neighbors. Other BNs within the 1-hop neighborhood will overhear this transmission and check the correctness of location information as explained earlier and accordingly update the corresponding BNs reputation entry in their Neighbor Reputation Table (NRT). They also evaluate the reputation values reported in the reporting nodes NRT in light of their own using a deviation test [22], and incorporate the reported values as explained in Section IV-C. Meanwhile, the SN will also receive the transmission consisting of location information and the reputation values. The SN first examines the the reported reputation values to form an opinion of its neighborhood on who to trust. It then uses the location information from trustable BNs to compute its location.

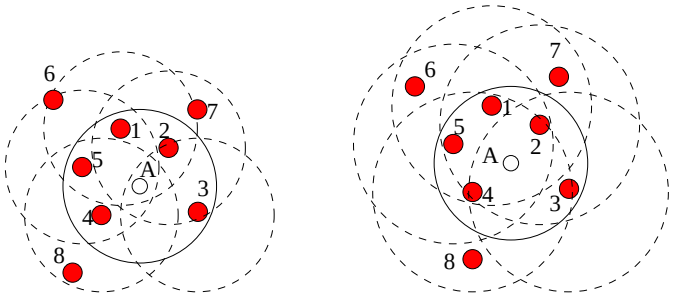


Fig. 3. (a)$RNG_{b_j} = RNG_{s_i}$ (b)$RNG_{b_j} = 1.5 \times RNG_{s_i}$

We shall now discuss in detail how a SN forms an opinion of its neighborhood. For discussion, consider a neighborhood consisting two BNs b_i and b_j and a SN s_k. Now, if b_i reports a reputation value for b_j that is over s_k's trust threshold TH_{REP}, then s_k counts that as a positive vote from b_i to b_j. For b_j to be trusted by s_k, it must have votes of trust from at least $\lfloor \frac{|NbrHood(s_k,b_j)|}{2} \rfloor$ BNs in the common neighbor set $C(s_k, b_j)$, where $NbrHood(s_k, b_j) = C(s_k, b_j) \bigcup b_j$. In other words, s_k considers b_j to be malicious if it receives votes of distrust from $\lceil \frac{|NbrHood(s_k,b_j)|}{2} \rceil$ or more BNs in $C(s_k, b_j)$. For example, in Figure 2, s_A's neighborhood includes five BNs, i.e., $N(s_A) = \{b_1, b_2, b_3, b_4, b_5\}$ and b_1's neighborhood includes four BNs, i.e., $N(b_1) = \{b_2, b_5, b_6, b_7\}$, $C(s_A, b_1) = N(s_A) \bigcap N(b_1) = \{b_2, b_5\}$, and $NbrHood(s_A, b_1) = C(s_A, b_1) \bigcup b_1 = \{b_1, b_2, b_5\}$. Now, in order for s_A to trust b_1, as explained earlier, at least $\lfloor \frac{|NbrHood(s_A,b_1)|}{2} \rfloor$ BNs in $C(s_A, b_1)$ must trust b_1, which in this example is equal to 1. This will be a correct assessment, assuming there are no more than $\lfloor \frac{|C(s_A,b_1)|}{2} \rfloor$ malicious BNs in $NbrHood(s_A, b_1)$ which again in this example is equal to 1.

B. Simple Majority

It is important to explain why the simple majority system works and is necessary. As mentioned in Section III-A, we assume that there are malicious BNs in the system, and that they can cooperate with each other. For discussion, refer to Figure 2. Assume that nodes b_1 and b_5 are malicious. If a simple system of *"If anyone trusts him, I trust him"* was used, b_5 could falsely praise of b_1, and b_1 could falsely praise b_5. Similarly, if a system of *"If anyone distrusts him, I distrust him"* was used, b_1 and b_5 could falsely accuse b_2 of being malicious, and have her location information discarded. In our system, though, we can illustrate the simple majority rule's validity.

Let us look at $NbrHood(s_A, b_2)$ in Figure 3(b). Here, $N(s_A) = \{b_1, b_2, b_3, b_4, b_5\}$, $N(b_2) = \{b_1, b_3, b_4, b_5, b_7\}$, $C(s_A, b_2) = \{b_1, b_3, b_4, b_5\}$, and $NbrHood(s_A, b_2) = C(s_A, b_2) \bigcup b_2 = \{b_1, b_2, b_3, b_4, b_5\}$. Since there are 4 nodes in $C(s_A, b_2)$, we can survive a collusion of up to $k = \lfloor \frac{|C(s_A,b_2)|}{2} \rfloor = 2$ malicious nodes in $NbrHood(s_A, b_2)$.

Here, we have two scenarios to investigate. In the first scenario, assume b_2 is benign. Then the two malicious nodes in $NbrHood(s_A, b_2)$ are both in $C(s_A, b_2)$, i.e., two of the nodes in $\{b_1, b_3, b_4, b_5\}$ are malicious. This means, in the worst case b_2 can get up to 2 negative votes and 2 positive votes. In the best case, however, b_2 can get up to 4 positive votes. This is because a malicious node's vote can vary. It can cast either a negative vote or a positive vote on a benign node and similarly it can cast either a negative vote or a positive vote on a malicious node. However, a benign node's vote will always be positive for another benign node and negative for a flagged malicious node. Therefore, when b_2 is benign, the positive votes can vary anywhere between 2 to 4. In this example, assuming both the malicious nodes cast a negative vote on the benign node b_2, it still does not affect the status of b_2, since for s_A to consider b_2 as malicious it should receive at least $\lceil \frac{|NbrHood(s_A,b_1)|}{2} \rceil$ negative votes, which in this example is 3 and we have already shown that b_2 can receive no more that 2 negative votes in this example.

In the second scenario, assume b_2 is malicious in which case there is only one malicious node in $C(s_A, b_2)$, i.e., only one of the nodes in $\{b_1, b_3, b_4, b_5\}$ is malicious. As a result, in the best case b_2 can receive 1 positive vote and will receive a minimum of 3 negative votes under all circumstances. This is because there are 3 benign nodes in $C(s_A, b_2)$ and they all will vote negative for b_2. In the best case, however, b_2 can receive upto 4 negative votes assuming that even the malicious node casts a negative vote on b_2. For this example, let us assume that the two malicious nodes are colluding and the other malicious node in $C(s_A, b_2)$ casts a positive vote for b_2. Now, since s_A receives $\lceil \frac{|NbrHood(s_A,b_1)|}{2} \rceil = 3$ negative votes against b_2, it considers b_2 as malicious and the one false vote from the other colluding malicious node cannot elevate b_2 to be trusted by s_A. Therefore, through this example, it is clear that even if they are lying all the time, the malicious nodes can neither exclude a benign node nor falsely elevate a malicious node. Therefore, our simple majority scheme works and is acceptable.

C. Algorithms

We recognize two classifications of information available to the reputation system for updating the reputation values.

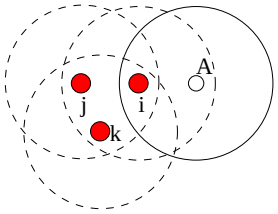


Fig. 4. b_i, b_j, and b_k are one hop neighbors. Only b_i is in range of s_A.

1) *First-hand information*: It is the information available by virtue of personal experience or direct observation. In DRBTS, a BN overhears location information transmitted by other BNs, in its communication range, in response to a location request. This is regarded as direction observation. On the otherhand, during periods of low network activity, a BN can use its pseudo-sensor IDs to disguise itself as a SN and request location information. This is regarded as personal experience.
2) *Second-hand information*: It is the information available when peers share their experiences. In DRBTS, BNs share their experiences by publishing their gathered reputation information in their 1-hop neighborhood.

Both these types of information are utilized by BNs to update the reputation of their neighbors. For illustration of the incorporation of first-hand information, refer to Algorithm 2A. Now consider a network setup in which BNs b_i, b_j, and b_k are 1-hop neighbors as shown in Figure 4. To simplify the example, we have considered a SN s_A that is within the range of b_i but outside of b_j and b_k's range. When s_A requests location information, b_i responds by broadcasting its location $TLoc_{b_i}$ and NRT_{b_i}. b_j and b_k overhear this broadcast transmission.

They then determine, using their own location, if they believe b_i is lying. Let us examine this in more detail using b_k as the example. b_k uses $TLoc_{b_i}$ to compute $CLoc_{b_k}$ and performs the following location deviation test:

$$|(ALoc_{b_k} - CLoc_{b_k})| < TH_{LD} \tag{1}$$

b_k then update its reputation entry for b_i as follows:

$$R_{k,i}^{New} = \mu_1 \times R_{k,i}^{Current} + (1 - \mu_1) \times \tau \tag{2}$$

If the location $TLoc_{b_i}$ was deemed to be truthful, i.e., the deviation test in Equation 1 was positive, then $\tau = 1$, otherwise $\tau = 0$. Similarly, b_j will update its reputation entry for b_i. Here, μ_1 is a factor used to weigh previous experience against current information. This factor can be varied according to system requirements. The value of TH_{LD} is set such that an intelligent adversary cannot lie in a deterministic manner and qualify the deviation test in Equation 1.

This is a simplified method of recording reputation, and more complicated representations, such as a Beta distribution [19], [21], may yield better performance. We have chosen a simplistic system for clarity, and how more complicated systems influence performance is on the agenda for our future work.

For illustration of the incorporation of second-hand information, published in NRT, refer to Algorithm 2B and return to the previous example where BNs b_i, b_j, and b_k are 1-hop neighbors and SN s_A is within the range of b_i but outside of b_j and b_k's range (Figure 4). Now, assume that b_k is the publishing node. b_j receives the

Algorithm 1

Construct $N(s_i)$

$N(s_i) \leftarrow \emptyset$
while ! TO_{res} **do**
  for each b_j that responds to s_i's location request **do**
    $N(s_i) \leftarrow N(s_i) \bigcup b_j$;
  end for
end while

Algorithm 2 Reputation Update

A: Firsthand Reputation:

for each location response $TLoc_{b_i}$ received **do**
  if $b_i \notin NRT_{b_k}$ **then**
    $NRT_{b_k} \leftarrow NRT_{b_k} \bigcup b_i$;
    $R_{k,i} \leftarrow 0$;
  end if
  compute $CLoc_{b_k}$ using $TLoc_{b_i}$;
  if $|ALoc_{b_k} - CLoc_{b_k}| < TH_{LD}$ **then**
    $R_{k,i}^{New} = \mu_1 \times R_{k,i}^{Current} + (1 - \mu_1)$;
  else
    $R_{k,i}^{New} = \mu_1 \times R_{k,i}^{Current}$;
  end if
end for

B: Secondhand Reputation:

for each NRT_{b_k} published **do**
  for each receiving $b_j \in N(b_k) : j \neq k$ **do**
    for each $b_i \in N(b_k) \bigcap N(b_j) : i \neq j \neq k$ **do**
      if $|R_{j,i}^{Current} - R_{k,i}^{Current}| < TH_{OD}$ **then**
        $R_{j,i}^{New} = \mu_2 \times R_{j,i}^{Current} + (1 - \mu_2) \times R_{k,i}^{Current}$
      else
        $R_{j,k}^{New} = \mu_3 \times R_{j,k}^{Current}$
      end if
    end for
  end for
end for

ID	$R_{i,j;\forall j \neq i}$
j	0.83
k	0.47
l	0.93

TABLE II
A SAMPLE NRT_{b_i}

ID	+ve Vote	-ve Vote
j	2	1
k	1	2
l	0	3

TABLE III
A SAMPLE TBN_{s_A}

published NRT_{b_k} which has reputation entries $R_{k,i}$ and $R_{k,j}$ for nodes b_i and b_j respectively which are beacon neighbors of b_k. A sample NRT is presented in Table II. Note that no BN will have an entry for itself in its NRT, i.e., $\forall i, R_{i,i} \notin NRT_{b_i}$. Therefore, b_j discards $R_{k,j}$ since $R_{k,j}$ is the reputation value of b_j from b_k's perspective and there is no way for b_j to incorporate it.

However, b_j considers $R_{k,i}$ but before incorporating $R_{k,i}$ into $R_{j,i}$, b_j first performs a simple opinion deviation test as follows:

$$|R_{j,i}^{Current} - R_{k,i}^{Current}| < TH_{OD} \tag{3}$$

If the above deviation test is positive, then the information published in NRT_{b_k} is considered to be compatible with b_j's and is accepted. b_j then updates $R_{j,i}$ in NRT_{b_j} as follows:

$$R_{j,i}^{New} = \mu_2 \times R_{j,i}^{Current} + (1 - \mu_2) \times R_{k,i}^{Current} \tag{4}$$

However, if the deviation test in equation 3 is negative, then the published information in NRT_{b_k} is considered to deviate too much from that of b_j's and is disregarded as incompatible information. In order to discourage nodes from publishing such false, incompatible information, the *lying* node's reputation is decremented as follows:

$$R_{j,k}^{New} = \mu_3 \times R_{j,k}^{Current} \tag{5}$$

Note that this is equivalent to b_j overhearing b_k giving false location information to s_A. In both cases, b_k's reputation is reduced for providing false information, but in different amounts. Also, in DRBTS, we don't have to worry about a node that first detects a misbehaving node getting punished for its findings deviating from the public opinion. This is because, location information and NRT is broadcast locally and can be detected by all the nodes in the neighborhood simultaneously. Hence, a benign node's findings will never deviate too much from the honest public opinion which, under our model constraints, will always have at least simple majority.

A SN, after broadcasting a location information request, waits untill a time-out TO_{res}, within which all beacon neighbors are expected to respond. Any BN that responds after TO_{res} is ignored and its location information is excluded from the localization process. Returning to the previous example, the SN s_A, receives the location information $TLoc_{b_j}$ and NRT_{b_j} broadcast by its beacon neighbor b_j, and continues to wait untill TO_{res} occurs. s_A then tabulates the results as follows. s_A first constructs $N(s_A)$ using Algorithm 1. Then, for each b_j in $N(s_A)$, it counts the number of $+ve$ votes and the number of $-ve$ votes, storing them in a table, similar to NRT, called Trusted Beacon Neighbor (TBN_{s_A}). A sample TBN is presented in Table III. Now, using Algorithm 3, location information from any BN that received more negative votes than positive votes is discarded. Then, finally location information from remaining beacon neighbors with at least $\lfloor \frac{|NbrHood(s_A,b_1)|}{2} \rfloor$ positive votes is used by s_A to calculate its location. Once the location is computed, TBN_{s_A} is flushed to free up memory, since the BNs are already keeping track of the long term reputation.

Algorithm 3 Create TBN_{s_i} and Exclude malicious b_j

```
Construct N(s_i)
for each b_j ∈ N(s_i) do
    for each b_k ∈ C(s_i, b_j) do
        if R_{k,j} > TH_REP then
            +ve_j ← +ve_j + 1;
        else
            −ve_j ← −ve_j + 1;
        end if
    end for
    if −ve_j > +ve_j then
        discard TLoc_{b_j}
    end if
end for
```

Parameters μ_1, μ_2, and μ_3 in equations 2, 4, and 5 respectively are system dependant parameters and are each range bound between 0 and 1. They each decide the extent to which past history can be discounted and substituted with the most recent behavior.

V. ANALYSIS

A. Common Neighbor Set Requirement

The extent to which SNs can withstand collusion depends on the size of their common neighbor set. For a SN s_A to withstand a collusion of k malicious BNs in its neighborhood, its common neighbor set with any beacon neighbor b_j should have at least $2k$ BNs. This essentially means that $|NbrHood(s_A, b_j)| = 2k + 1$. Therefore, the robustness and performance of DRBTS depends on the size of the common neighbor set. Any violation of

the size requirement of the common neighbor set will result in degraded system performance and may result in breach of security.

DRBTS has been modeled as an undirected graph. For details please refer to section VI-A. The set of vertices $V = v_1 \bigcup v_2$. Here, $v_1 = \{s_1, s_2, ..., s_n\}$ and $v_2 = \{b_1, b_2, ..., b_m\}$ where n and m are system dependant parameters and represent the number of SNs and BNs respectively. The sensitivity of the system and its performance for different values of n and m have been studied through simulations and results are presented in section VI-B. Let $N(s_A)$ and $N(b_j)$ represent the neighbor set of SN s_A and BN b_j respectively. When we say neighbor set, we are always referring to the beacon neighbor set as beacon nodes are of our interest. We will now look at how the size of $C(s_A, b_j)$ affects the performance of the system and the minimum size of $C(s_A, b_j)$ needed to defeat a collusion of k malicious nodes. We know that

$$C(s_A, b_j) = N(s_A) \bigcap N(b_j) \tag{6}$$

The value of $|C(s_A, b_j)|$ determines the extent to which the system is robust against collusion. For a network with k malicious BNs that can potentially collude, the worst case scenario occurs when all the k malicious nodes are in the same neighborhood $N(s_A)$. However, the chance that this scenario occurs is very unlikely. In any event, the system must have, on average, a minimum of $2k$ beacon nodes in every $C(s_A, b_j)$ for a completely robust system. Therefore, the equations below give the necessary and sufficient conditions for a robust system:
$\forall[s_A \in v_1; b_j \in v_2; b_j \in N(s_A)]$,

$$|C(s_A, b_j)| = 2k \tag{7}$$

$$NbrHood(s_A, b_j) = C(s_A, b_j) \bigcup b_j = 2k + 1 \tag{8}$$

$$Maximum\ Malicious\ Nodes = \lfloor \frac{|C(s_A, b_j)|}{2} \rfloor = k \tag{9}$$

$$Minimum\ -ve\ Votes \geq \lceil \frac{|NbrHood(s_A, b_j)|}{2} \rceil \tag{10}$$

$$Minimum\ +ve\ Votes \geq \lfloor \frac{|NbrHood(s_A, b_j)|}{2} \rfloor \tag{11}$$

Equation 9 gives the maximum permissible number of malicious nodes in a neighborhood. Equation 10 gives the minimum number of negative votes needed to flag a node as malicious while Equation 11 gives the minimum number of positive votes needed to flag a node as trusted.

There are two scenarios here that we need to analyze. First, if $\lfloor |C(s_A, b_j)| \rfloor$ is odd, then there will be no tie between positive and negative votes. But, in the second scenario, when $\lfloor |C(s_A, b_j)| \rfloor$ is even, then there can be a tie between the number of positive and negative votes. When there is a tie, however, Equaltion 11 will always win over Equation 10. This clearly indicates the the BN in question is benign and the colluding malicious nodes cannot affect its status since the minimum number of negative votes needed to flag a node as malicious is $\lceil \frac{|NbrHood(s_A, b_j)|}{2} \rceil$ and when $\lfloor |C(s_A, b_j)| \rfloor$ is even it is equal to $\lfloor \frac{|C(s_A, b_j)|}{2} \rfloor + 1$. This, however, is greater than the permissible number of malicious nodes as per Equation 9.

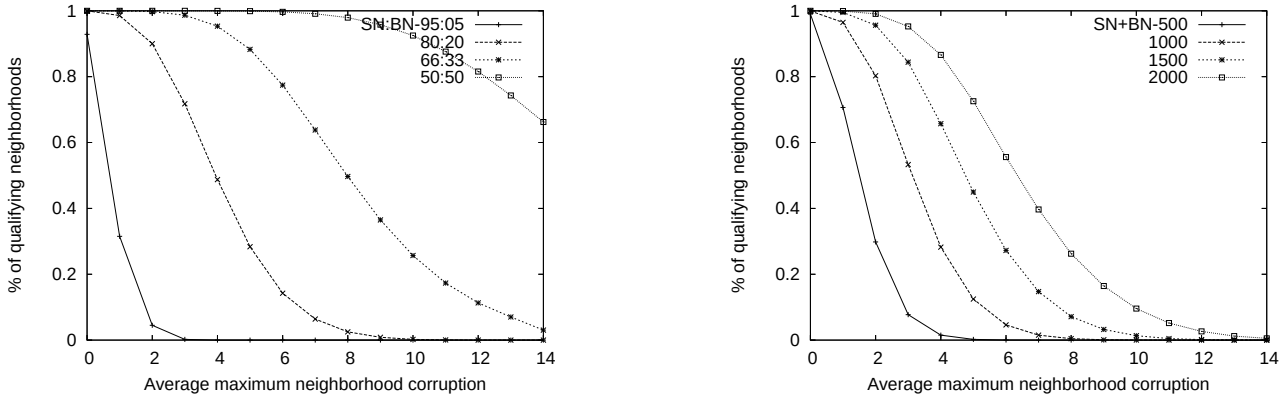
B. Extended Communication Range for Beacon Nodes

In this paper, for the most part, we have assumed uniform communication ranges for both SNs and BNs, i.e., $RNG_{b_j} = RNG_{s_A}$. Providing the BNs with extended communication range helps each SN s_A to get more input on the trustworthiness of all BNs in its neighbor set $N(s_A)$. In Figure 3(a), $RNG_{b_j} = RNG_{s_A}$ and $|C(s_A, b_2)| = |\{1, 3\}| = 2$ and in Figure 3(b), $RNG_{b_j} = 1.5 \times RNG_{s_A}$ and $|C(s_A, b_2)| = |\{1, 3, 4, 5\}| = 4$. Hence, it is clear that when $RNG_{b_j} = RNG_{s_A}$, even though b_4 and b_5 are in $N(s_A)$, it does not get any input from b_2 about their trustworthiness. But by merely increasing RNG_{b_j} to $1.5 \times RNG_{s_A}$, $N(s_A)$ is completely covered. Hence, it is beneficial for the system to have BNs with higher communication range than the SNs. Consequently, s_A can now decide whether or not to include the location information of any given BN in $N(s_A)$, with higher confidence, while computing its own location information.

C. Beacon-Node-Based Routing

Efficient routing is one of the most critical functions in WSNs due to severe energy restriction of sensors. It is made more complicated by the lack of an infrastructure. If the routing burden on SNs can be mitigated, then the effective lifetime of sensors and eventually the entire network can be prolonged. One way to achieve reduced routing overhead is to use BNs as relay nodes to forward packets. Since watchdogs are already being used for monitoring the behavior of BNs, they can easily be adapted to also make a decision on which nodes to choose for routing.

When used for routing, BNs will take on the function of cluster heads. A SN defaults to either the BN



(a) Varying SN:BN, fixed n

(b) Fixed SN:BN, varying n

Fig. 5. (a)Impact of varying SN:BN ratio with n=1000 on robustness. (b) Impact of varying n with SN:BN ratio fixed at 80:20 on robustness.

closest to it or the BN with the highest reputation value, among the nodes that satisfy equation 11, as its cluster head, keeping that BN's location in memory. The choice between these two methods for selecting cluster head has an inevitable tradeoff between security and energy efficiency. By choosing a BN that is physically the closest among the qualifying nodes only ensures a basic level of security but is more energy efficient. On the otherhand, by choosing the node with the highest reputation value among the qualifying nodes, the physical distance between the SN and the chosen cluster head BN may be larger which consumes higher energy. However, it ensures a high degree of security and reliable routing and delivery. In any event, we cannot exclude the possibility of a BN cluster head starting out as benign but getting corrupt at a later stage. To address this issue, in DRBTS, SNs regularly check for the best BN cluster head, either in terms of shortest distance or maximum accumulated reputation, depending on the methodology used. Using BN as routing backbone has many advantages as listed below.

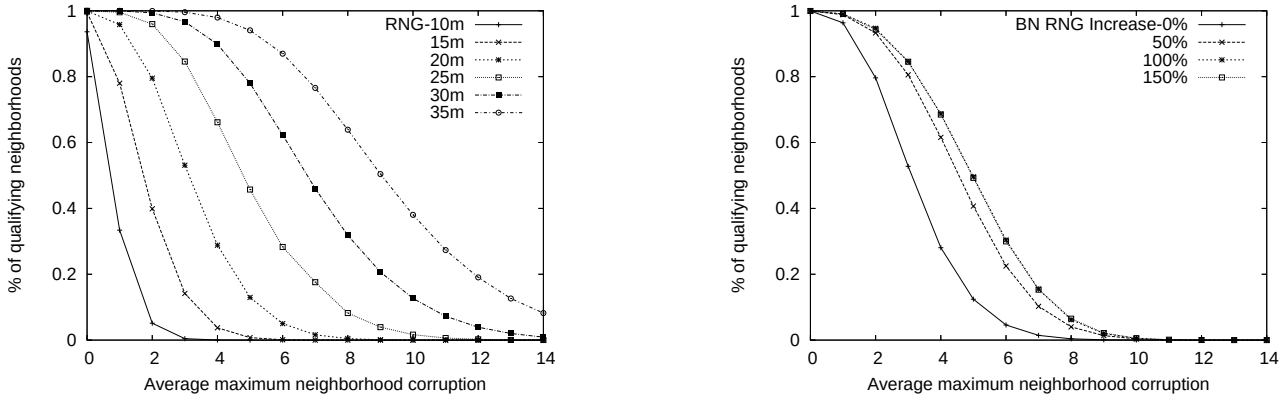
- It results in energy savings for SNs since they are excused from their routing obligations thereby prolonging their lifetime and consequently the lifetime of the entire network.
- The burden on a SNs memory is completely eliminated since they need to store only their cluster head BN location.
- Since the number of BNs is relatively small, it reduces the size of the routing table rendering route updates and maintenance faster and easier.
- By extending BNs transmission range, the number hops needed to deliver a message from any source node to any destination node can be controlled.
- Additionally, since the BNs are used as routing backbone, it mitigates the redundancy of exchanged messages avoiding unnecessary network traffic. Thus it results in reduced contention and collision in the network.

Nonetheless, there is a downside to using BNs as cluster heads. This will have an even greater load placed on them than previously. In systems where beacon nodes do not take part in any other network activities to save energy for their beacon duties, using them as a routing backbone is counterproductive. But, in systems where beacon nodes are more resource rich nodes, using them as a backbone offers to kill two birds with one stone.

We have discussed the benefits of increasing the transmission range of BNs in section V-B. In the case where BNs are used as a routing backbone, this increase in transmission range offers additional benefit, as packets can be forwarded to the destination in fewer hops compared to routing via SNs.

D. Overhead

The proposed DRBTS has some additional overhead. It requires extra memory to store the NRT and TBN. Publishing first-hand information adds to the communication overhead while calculating and updating the reputation of neighboring nodes lends itself to computational overhead. However, by combining the publishing of the NRT with location information in response to a location request, we mitigate the communication overhead to a large extent and almost reduce it to as much in the base model [10]. Also, elimination of pairwise keys used in [10] compensates for the memory overhead introduced by the storage of NRTs and TBNs. Note that TBNs are stored only temporarily and flushed as soon as the SN computes its location. It should be noted that the pairwise keys are eliminated to enable promiscuous observation



(a) SN and BN with same transmission range

(b) Extended transmission range for BN

Fig. 6. (a)Impact of Varying transmission range on robustness. (b) Impact of extending BN transmission range on robustness.

in the neighborhood. However, to ensure integrity of data, we use a network-wide group key. Additionally, because the reputation system is distributed, there is no information bottleneck at the base station and there is less network traffic.

VI. Simulation and Results

A. Setup and Environment

In this paper we consider a WSN consisting of n SNs $s_1, s_2, ..., s_n$ and m BNs $b_1, b_2, ..., b_m$. We model the network as an undirected graph $G = (V, E)$, with the set of vertices $V = v_1 \bigcup v_2$, with v_1 being the set of SNs and v_2 being the set of BNs. E is the set of edges. An edge exists between any two nodes that are in each other's communication range.

Our simulation has been carried out on a custom-built Java simulator. The simulation results show how different variables affect the common neighbor set requirement and consequently the robustness of the system. The higher the average number of neighbors, the greater the corruption the network can withstand. The simulation results help in understanding how various factors affect performance and can allow for informed decision making, based on the expected chance of node corruption, as well as to give an idea of the system's tolerance of failing nodes.

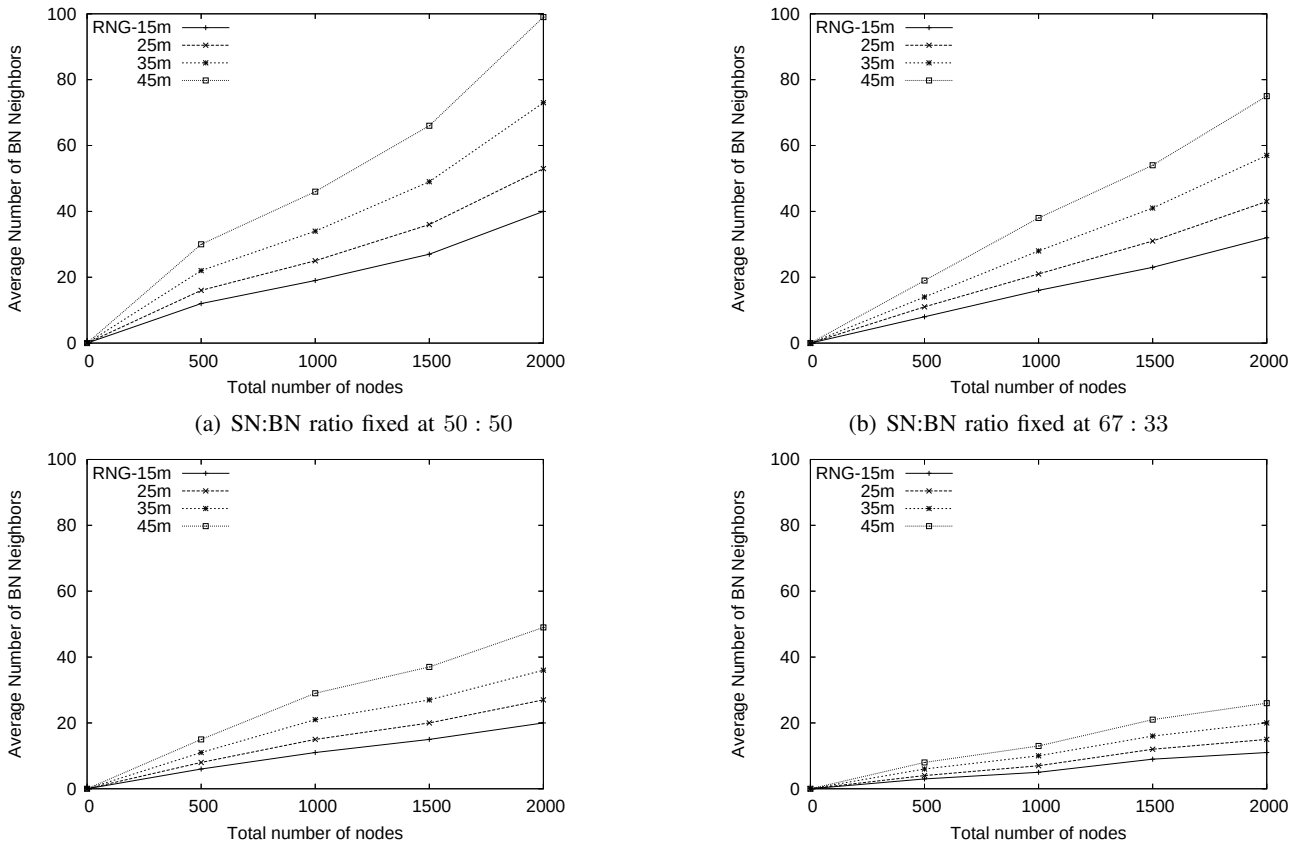
For each trial in the simulation, a field of $500m \times 500m$ was randomly seeded with uniformly distributed SNs and BNs. A trial was examined by taking 200 random sensor-beacon pairs and measuring the size of their common neighbor sets. 500 trials were performed and averaged for statistical stability. We plot the misbehavior density against the percentage of SNs that have sufficient neighbors to withstand collusion in their neighborhood.

B. Results

In Figure 5(a), we examine the effect of varying the ratio of BNs to SNs on the robustness of DRBTS. The network was deployed with 1000 SNs, and the number of BNs was varied to get the appropriate ratios. The transmission range for both SNs and BNs was fixed at $20m$. SN to BN ratios of 95:5, 80:20, 67:33, and 50:50 were tested, and the system performed the best with 50:50 ratio. However, ratios of 80:20 and 67:33 also performed very well. With an 80:20 ratio, 50% of SNs can withstand a collusion of $k = 5$ malicious nodes in their neighborhood where as in 67:33 ratio they can withstand up to $k = 8$. It is evident from the results that, the higher the number of BNs, the more robust DRBTS gets.

We have also studied the impact of the total number of nodes on the robustness of DRBTS. The results are presented in Figure 5(b). In the test runs we have fixed the ratio of SN to BN at 80:20, and vary the total number of nodes in the network from 500 to 2000 in steps of 500. Transmission range remains constant at $20m$. It can be seen that the robustness of the system increases as the number of nodes increases. This is true because as the network becomes more dense, the size of common neighbor set increases and so does robustness.

In Figure 6(a), the effects of transmission range on robustness is presented. In this case, n and $SN{:}BN$ were kept constant at 1000 and 80:20 respectively. The transmission range of SN and BN were uniformly varied from $10m$ to $35m$ in $5m$ step intervals. Once again, not to our surprise, the system tends to get more robust with increasing transmission range which again is due to the fact of growing size of the common neighbor set. A transmission range of 20-25m is sufficient to achieve enough robustness at which point about 80% of SN can



(a) SN:BN ratio fixed at $50:50$

(b) SN:BN ratio fixed at $67:33$

(c) SN:BN ratio fixed at $80:20$

(d) SN:BN ratio fixed at $95:05$

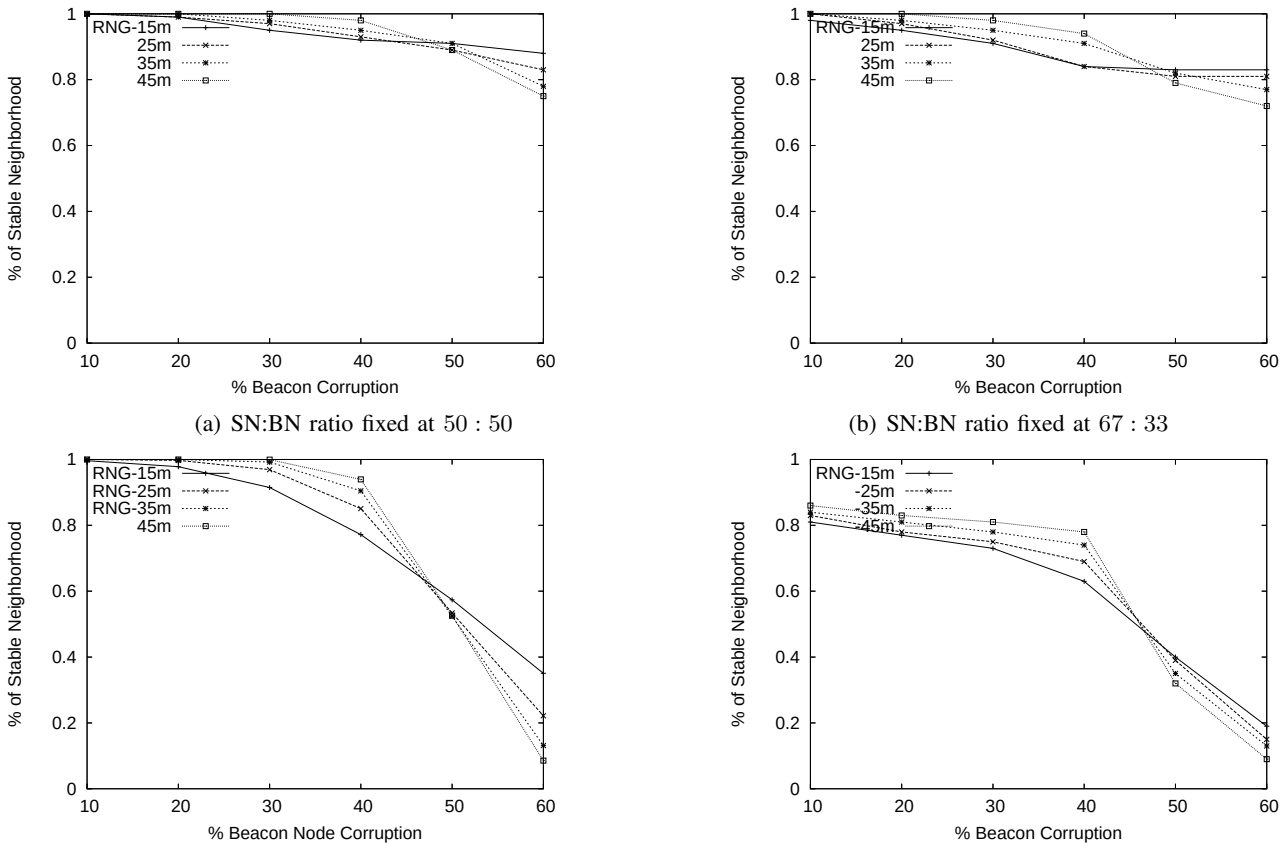
Fig. 7. Average number of BN neighbors with transmission range varied from 15-45m in steps of $10m$.

withstand about $k = 2$ to 4 colluding BNs in their neighborhood. A higher transmission range will increase the robustness but will result in reduced lifetime of the nodes, in particular the resource constrained SNs, as they expend more energy.

We look at a possible extension to DRBTS in which BNs have a larger communication range compared to SNs which is presented in Figure 6(b). This would allow a smaller number of BNs to give the same coverage. As we can see, the system is more robust when the BNs have a higher transmission range compared to SNs. However, beyond a 50% increase in the BN communication range compared to SN, the system shows stability and robustness of the system grows very slowly and beyond 100% it ceases to get any more robust. With a 50% higher transmission range for BNs, about 80% of SNs can withstand up to $k = 3$ colluding BNs in their neighborhood.

We have presented the results for the average number of BN neighbors an SN has. We have simulated different scenarios. Then the simulation was run for $n = 500$, 1000, 1500 and 2000 with the transmission range of both SN and BN uniformly varied from $15m$ to $45m$ in steps of $10m$. Different ratios of SN:BN was examined and results are presented in Figure 7(a), (b), (c), and (d) with SN:BN ratio of 50:50, 67:33, 80:20, and 95:05 respectively. It is clear from the results that the average number of beacon neighbors is sensitive to both total number of nodes as well as the transmission range. However, it is more sensitive to communication range than the total number of nodes. This is one other benefit of increasing the communication range which is discussed in Section V-B.

In our final simulation, we wish to show the total robustness of the system. We take what we believe is a fair system, with 1000 total nodes, an 80:20 ratio of sensors to beacons, and uniformly varied the communication range for both sensors and beacons nodes, from $15m$ to $45m$ in steps of $10m$. We then add varying levels of corruption by giving each beacon node a percentage chance of being corrupt. We then take random sensor-beacon pairs, and compare the number of good nodes to corrupt nodes, and determine if that neighborhood is stable under our simple majority voting system. We have also tested the impact of different SN:BN ratios 50:50, 67:33, and 95:5. The results are displayed in Figure 8(a),



(a) SN:BN ratio fixed at $50:50$

(b) SN:BN ratio fixed at $67:33$

(c) SN:BN ratio fixed at $80:20$

(d) SN:BN ratio fixed at $95:05$

Fig. 8. Percentage of Stable Neighborhood with transmission range varied from 15-45m in steps of $10m$.

(b), (c) and (d) respectively.

It is easy to see that even with a very small communication range, the system exhibits very good stability for up to 20% corruption. The larger the communication range, the slower stability degrades as corruption rises, until a critical point of around 45% is reached. These results can be easily explained by the fact that a sparse neighborhood is more subject to variance, as it is a small sample set. The larger the transmission range, the larger the neighborhood, and the closer the neighborhood cleaves to the system-wide average level of corruption. It is worth noting that even at a 40% corruption rate, the system with $25m$ transmission range and a SN:BN ratio of 80:20 was 85% stable. This best showcases that the robustness of our model is second to none.

VII. Conclusion and future work

In this paper we have presented a novel method for allowing sensor nodes to rely on trusted beacon nodes, based on a simple majority principle, to provide location information. We have shown through simulations that the proposed scheme is robust in dense networks and can be tailored to specific security requirements depending on the application domain. We have also shown that our scheme adds relatively less overhead, compared to similar schemes.

In our future work, we will examine more complex models for reputation, such as the Beta distribution, examine how DRBTS can be adapted to counteract a broader range of malicious behavior, and investigate through simulation the idea of using BNs as cluster heads for routing purposes by extending the proposed DRBTS, which has been introduced in section V-C. This will enhance the life of SNs, thereby enhancing the overall lifetime of the system. We will also conduct more exhaustive simulations to confirm the robustness of our system and devise a neighborhood group-key mechanism such that communication can be secure. The scheme will allow BNs to decrypt messages but only those transmitted by nodes which are in their range. The challenge in devising such schemes lies in deciding dynamically how many groups each node belongs to. Node mobility and topology changes are two other issues on our agenda for future work. We would like to extend DRBTS to address these two issues as well and confirm it through simulations.

REFERENCES

[1] C.-Y. Chong and S.P. Kumar. Sensor networks: Evolution, opportunities, and challenges. *Proceedings of the IEEE*, August 2003.

[2] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. F. Abdelzaher. Range-free localization schemes in large scale sensor networks. *In Proceedings of ACM MobiCom 2003*, 2003.

[3] L. Doherty, K. S. Pister, and L. E. Ghaoui. Convex optimization methods for sensor node position estimation. *In Proceedings of INFOCOM'01*, 2001.

[4] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and D. Tygar. SPINS: Security protocols for sensor networks. *In Proceedings of Seventh Annual International Conference on Mobile Computing and Networks*, July 2001.

[5] A. Savvides, C. Han, and M. Srivastava. Dynamic fine-grained localization in ad-hoc networks of sensors. *In Proceedings of ACM MobiCom '01, pages 166-179*, July 2001.

[6] A. Savvides, H. Park, and M. Srivastava. The bits and flops of the n-hop multilateration primitive for node localization problems. *In Proceedings of ACM WSNA '02*, September 2002.

[7] N. Bulusu, J. Heidemann, and D. Estrin. GPS-less low cost outdoor localization for very small devices. *In IEEE Personal Communications Magazine*, pages 28-34, October 2000.

[8] R. Nagpal, H. Shrobe, and J. Bachrach. Organizing a global coordinate system from local information on an ad hoc sensor network. *In IPSN'03*, 2003.

[9] A. Nasipuri and K. Li. A directionality based location discovery scheme for wireless sensor networks. *In Proceedings of ACM WSNA'02*, September 2002.

[10] D. Liu, P. Ning, and W. Du. Detecting Malicious Beacon Nodes for Secure Location Discovery in Wireless Sensor Networks. *In* 25*th IEEE International Conference on Distributed Computing Systems (ICDCS'05)*, pp. 609-619, 2005.

[11] L. Lazos and R. Poovendran. Serloc: Secure range independent localization for wireless sensor networks. *In ACM workshop on Wireless security (ACM WiSe 2004)*, Philadelphia, PA, October 1 2004.

[12] N. Sastry, U. Shankar, and D.Wagner. Secure verification of location claims. *In ACM Workshop on Wireless Security*, 2003.

[13] S. Marti, T. J. Giuli, K. Lai, and M. Baker. Mitigating Routing Misbehaviour in Mobile Ad Hoc Networks. *In Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking (MobiCom 2000)*.

[14] P. Michiardi and R. Molva. CORE: A COllaborative REputation mechanism to enforce node cooperation in Mobile Ad Hoc Networks. *Communication and Multimedia Security*, September, 2002.

[15] A. Srinivasan, J. Teitelbaum and J. Wu. DRBTS: Distributed Reputation-based Beacon Trust System. *In the* 2*nd IEEE International Symposium on Dependable, Autonomic and Secure Computing (DASC'06)*, Indianapolis, USA, 2006.

[16] A. Srinivasan, J. Teitelbaum, H. Liang, J. Wu, and M. Cardei. Reputation and Trust based System for Ad Hoc and Sensor Networks. *In Algorithms and Protocols for Wireless Ad Hoc and Sensor Networks*, A. Boukerche (ed), Wiley&Sons, 2006.

[17] S. Bansal and M. Baker. Observation-based Cooperation Enforcement in Ad Hoc Networks. *http://arxiv.org/pdf/cs.NI/0307012*, July 2003.

[18] S. Buchegger and J.-Y. Le Boudec. Performance Analysis of the CONFIDANT Protocol (Cooperation Of Nodes-Fairness In Dynamic Ad-hoc NeTworks). *In the Proceedings of MobiHoc 2002*, Lausanne, CH, June 2002.

[19] S. Buchegger and J.-Y. Le Boudec. A Robust Reputation System for Peer-to-Peer and Mobile Ad-hoc Networks. *Proceedings of P2PEcon 2004*, Harvard University, Cambridge MA, U.S.A., June 2004.

[20] S. Buchegger, C. Tissieres, and J.-Y. Le Boudec. A Test-Bed for Misbehavior Detection in Mobile Ad-hoc Networks - How Much Can Watchdogs Really Do? *Proceedings of IEEE WMCSA 2004*, English Lake District, UK, December 2004.

[21] S. Ganeriwal and M. Srivastava. Reputation-based framework for high integrity sensor networks. *In Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks (SASN '04)*, October 2004 pp. 66-77.

[22] J. Mundinger and J.-Y. Le Boudec. Analysis of a Reputation System for Mobile Ad-Hoc Networks with Liars. *In Proceedings of The 3rd International Symposium on Modeling and Optimization*, Trento, Italy, April 2005.

[23] D. Liu and P. Ning. Establishing pairwise keys in distributed sensor networks. *In Proceedings of 10th ACM Conference on Computer and Communications Security (CCS'03)*, pages 52-61, October 2003.

[24] H. Chan, A. Perrig, and D. Song. Random key predistribution schemes for sensor networks. *In IEEE Symposium on Research in Security and Privacy*, pages 197-213, 2003.

[25] L. Eschenauer and V. D. Gligor. A key-management scheme for distributed sensor networks. *In Proceedings of the 9th ACM Conference on Computer and Communications Security*, pages 41-47, November 2002.