DRBTS: Distributed Reputation-based Beacon Trust System

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

***Abstract*— Wireless Sensor Networks (WSNs) have critical applications in diverse domains like environmental monitoring and military operations where accurate location of sensors is vital. One common method of location discovery uses a set of specialty nodes known as beacon nodes (BNs) that assist other sensor nodes (SNs) to determine their location. This paper proposes a novel reputation-based scheme called Distributed Reputation-based Beacon Trust System (DRBTS) for excluding malicious BNs that provide false location information. To the best of our knowledge, DRBTS is the first model to use the concept of reputation for excluding 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). The BNs then publish their NRT in their 1-hop neighborhood. BNs use this second-hand information published in NRT for updating the reputation of their neighbors after it qualifies a deviation test. On the other hand, the SNs use the NRT information to determine whether or not to use a given beacon's location information, based on a simple majority voting scheme.**

***Index Terms*— Beacon node, malicious node, reputation, revocation, security, sensor node, trust.**

I. Introduction

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 of determining their own location, they have no way of determining which BNs are being truthful. This information asymmetry has not been considered by previous works, and as such complicates their implementation in this environment. To solve this problem, we propose 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(s), 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.

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 outlines our DRBTS model. In Section V we present an analysis of our DRBTS scheme. In Section VI we conclude our paper with directions for future work.

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. Savvides et al [1] 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 [2]. Lazos and Poovendran [7] 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. SeRLoc is a distributed algorithm and does not require any communication among sensors. In [8], Sastry et al introduced the concept of secure location verification, and show how it can be used for location-based access control.

Two techniques for improving throughput in ad-hoc networks were presented in [9]. One is the *watchdog*, which identifies misbehaving nodes and the other is the *pathrater*, which helps routing protocols to avoid these nodes. The watchdog system has often been used as the prototypical promiscuous monitoring system in subsequent research.

Michiardi and Molva [10] proposed CORE which 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. Here, nodes exchange only positive reputation information. The authors argue that this prevents a false-negative (badmouthing) attack, but do not address the issue of collusion to create false praise. In CORE, members have to contribute on a continuing basis to remain trusted or they will find their reputation deteriorating until they are excluded.

Buchegger and Boudec [11] have presented CONFIDANT with predetermined trust, and later improved it with an adaptive bayesian reputation and trust system and an enhanced passive acknowledge mechanism (PACK) in [12] and [15] respectively. Mundinger and Boudec [17] 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., nodes accept second-hand information only if it does not deviate too much from the node's reputation value.

Ganeriwal and Srivastava [16] 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.

Finally, in [6], Liu and associates 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 (Ac) and report counter (Rc). This system is a very simple example of a reputation-based system, in which nodes garner negative reputation for misbehavior. 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 [6] by introducing a reputation-based scheme.

III. Problem Definition

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 being the set of SNs and BNs and the set of edges E being the link between them. An edge exists between any two nodes that are in each other's communication range. Formally, the problem addressed in this paper can be stated as, "Given a network with BNs and SNs, how to exclude malicious BNs that provide SNs with incorrect location information?"

DRBTS is developed with the following underlying assumptions:

1) BNs are static after deployment. Malicious BNs can collaborate.
2) We are considering only dense networks. In order for a neighborhood to be resilient to k malicious BNs, there must be at least $2k$ BNs in that neighborhood.
3) Location information is broadcast to the requesting sensor node by the BN, unlike [6] where it is unicasted.
4) Location information is not encrypted using a pairwise key, unlike [6]. We instead assume a network-wide group key for encryption, to allow promiscuous observation in the network, while preventing outsiders from overhearing.
5) 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.

Table 1 is used as an index to the acronyms used throughout this paper.

IV. Reputation

Reputation is the opinion of one entity about another. In an absolute context, it is the trustworthiness of an en-

TABLE I
NOTATION INDEX

s_i, b_j	Sensor Node i / Beacon Node j
SN, BN	Sensor Node / Beacon Node
TBN_{s_i}	Trusted Beacon Neighbor Table of s_i
NRT_{b_j}	Neighbor Reputation Table of b_j
$R_{i,j}$	Reputation of b_j in NRT_{b_i}
$N(s_i), N(b_j)$	Neighbor Set of s_i / b_j
(s_i, b_j)	Sensor Beacon Pair
$C(s_i, b_j)$	Common Neighbor(s) set of (s_i, b_j)
RNG_{s_i}, RNG_{b_j}	Sensing/Transmission Range of s_i / b_j
$CLoc_{b_j}$	Computed Location of b_j
$ALoc_{b_j}$	Actual Location of b_j
$TLoc_{b_j}$	Location Transmitted by b_j
$+ve_{b_j}$	Votes for b_j
$-ve_{b_j}$	Votes against b_j
TH	Threshold below which $TLoc_{b_j}$ is discarded

tity. The foremost difficulty in adapting standard watchdog mechanisms to systems involving location-aware BNs is that the SNs do not have first-hand experience to compare second-hand information provided to them by BNs. Hence, without any assumptions, there is no way to determine who can be trusted. One logical method is to assume that the majority of BNs are honest. From there, one can use a simple majority principle to determine the truth.

Sensors in the DRBTS operate on the aforementioned simple majority scheme. 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 can overhear the responses of other BNs in its area. It can then determine its location using this claimed location of each BN and comparing them against its true location. If the difference is within a certain margin of error, then the corresponding BN is considered benign, and its reputation increases. If the difference is greater than the margin of error, then that BN is considered malicious and its reputation is decreased.

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 (Reputation is 1 at $t = 0$) until they do something bad, or it can assume they are bad nodes (Reputation is 0 at $t = 0$) until they prove themselves. The benefit of the former is that the system needs no initial setup time. The main drawback is that it not only allows, but encourages nodes who have bad reputation to simply spoof a new ID, and re-enter the system with fresh reputation. This is solved by assuming that an unknown beacon is untrusted, i.e., when a BN hears another BN responding for the first time, it sets the new BN's reputation to 0 before evaluating the transmission. Thus, there is no incentive for identity spoofing. The drawback to this is that the system begins at 0. No one trusts anyone else, and SNs cannot get location information. Our system assumes the untrusted model, but adds a method for BNs to bootstrap reputation. Each BN is given a small number of fake IDs, as in [6]. In periods of low network activity, a BN can use one of these IDs to disguise itself as an SN and request location information, triggering responses from nearby BNs. 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 training 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 second-hand 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 tied to the rate at which the reputation changes.

So, when a SN sends a broadcast asking for 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 evaluate these findings in light of their own using a deviation test [17], and incorporate the findings as explained later in this section. Meanwhile, the SNs will also receive these reports, and use them to form an opinion of their neighborhood. If a BN reports a trust value over the SN's trust threshold for another BN, the sensor counts that as a positive vote from the first BN to the second. For a BN b_j to be trusted by a SN s_i, it must have votes of trust from at least half of the BNs in the common neighbor(s) set, $C(s_i, b_j)$, of (s_i, b_j). For example, in Figure 1, s_A's neighborhood includes five BNs, i.e., $N(s_A) = \{1, 2, 3, 4, 5\}$. In order for s_A to trust b_1, at least 1 other BN in $C(s_A, b_1)$ must trust b_1. This will be a correct assessment, assuming there are no more than

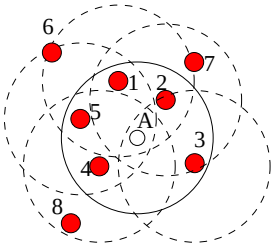


Fig. 1. $N(s_A) = \{1, 2, 3, 4, 5\}$; $N(b_1) = \{2, 5, 6, 7\}$; $C(s_A, b_1) = N(s_A) \bigcap N(b_1) = \{2, 5\}$

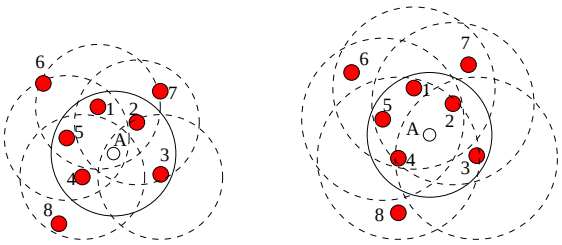


Fig. 2. (a)$RNG_{b_j} = RNG_{s_i}$. (b)$RNG_{b_j} = 1.5 \times RNG_{s_i}$.

$k = 1$ malicious BNs in $C(s_A, b_1)$.

It is important to explain why the simple majority system works and is necessary. As has already been stated, we assume that there are malicious BNs in the system, and that they can cooperate with each other. Consider the Figure 1 above. 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 send false 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 could falsely accuse b_2 of being malicious, and have her location information discarded. In our system, though, we can illustrate the majority rule's validity. Let us look at the sensor-beacon pair (s_A, b_2) in Figure 2(b). Their common neighbor(s) set, i.e., $C(s_A, b_2) = \{1, 3, 4, 5\}$. Since there are 4 nodes in $C(s_A, b_2)$, we can survive a collusion of up to $k = 2$ malicious nodes in $C(s_A, b_2)$. First, if b_2 is good, that means it will get up to 2 negative votes, and 2 positive votes (from the malicious and the good nodes, respectively). If b_2 is bad, then there can be at most 1 other malicious node in this neighbor-set, and so b_2 will receive 3 negative votes, and 1 positive vote. Since a good node will always give a positive vote to another good node, and a negative vote to a discovered malicious node, only the malicious node's votes can vary. But since it is shown that even lying all the time, the malicious nodes cannot exclude a good node, and cannot falsely elevate a malicious node, this is acceptable.

A. Algorithms

Algorithm 1

Construct $N(s_i)$

1: $N(s_i) \leftarrow \emptyset$
2: **for** each b_j that publishes $TLoc_j$ and NRT_{b_j} that s_i can receive **do**
3: $N(s_i) \leftarrow N(s_i) \bigcup b_j$;
4: **end for**

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

1) A BN may overhear location information transmitted by another BN in its communication range. This observation is *first-hand information*.
2) BNs publish their gathered reputation information while responding to a request for location information. This is *second-hand information*.

Both these types of information are used by the BNs to update the reputation of their neighbors. For illustration of the first type of information refer to Algorithm 2A. Consider a network setup in which BNs b_i, b_j, and b_k are 1-hop neighbors. To simplify the example, assume a SN outside of b_j and b_k's range asks for location information. b_i responds by broadcasting its own location. b_j and b_k overhear. They then determine, using their own location, if they believe b_i is lying. They then update their b_i entry as follows, using b_k as the example.

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

If the location was deemed to be truthful, $\tau = 1$, otherwise $\tau = 0$. μ_1 is a factor used to weight previous experience against current information. This factor can be varied according to system requirements. This is a simplified method of recording reputation, and more complicated representations, such as a Beta distribution [12], [16], may yield better performance. We have chosen a simplistic system for clarity, and will examine how more complicated systems influence performance in our future work.

When a node requests location information, every beacon neighbor of the requesting node will publish its Neighbor Reputation Table (NRT) (see Table II) along with its own location. To understand how a BN incorporates second-hand information from a neighboring BN let us refer to Algorithm 2B and return to the previous example. Assume that b_k is the publishing node. b_j receives $R_{k,j}$ $R_{k,i}$. Since b_j does not maintain an entry in its NRT about itself, $R_{k,j}$ is discarded. Before incorporating $R_{k,i}$, b_j first performs a simple deviation

Algorithm 2 Reputation Update

A: Firsthand Reputation:

for each $TLoc_{b_i}$ **do**
  if $b_i \notin NRT_{b_k}$ **then**
    $R_{k,i} \leftarrow 0$
  end if
  compute $CLoc_{b_k}$ using $TLoc_{b_i}$;
  if $CLoc_{b_k} - ALoc_{b_k} < TH$ **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 $|R_{j,i}^{Current} - R_{k,i}^{Current}| \leq d$ **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}.

test as follows.

$$|R_{j,i}^{Current} - R_{k,i}^{Current}| \leq d. \quad (2)$$

If the above deviation test is positive, then the information is considered compatible with b_j's first-hand experience, and is accepted. b_j then updates $R_{j,i}$ in NRT_{b_j} as follows

$$R_{j,i}^{New} = \mu_2 \times R_{j,i}^{Current} + (1 - \mu_2) \times R_{k,i}^{Current}. \quad (3)$$

However, if the deviation test in equation 2 is negative, then the published information is considered to deviate too much from its own first-hand experience, and is disregarded as incompatible information. In order to discourage nodes from publishing false information, the *lying* node's reputation is decreased as follows

$$R_{j,k}^{New} = \mu_3 \times R_{j,k}^{Current}. \quad (4)$$

Note that this is equivalent to b_j overhearing b_k giving false location information. 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 since its findings will deviate 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.

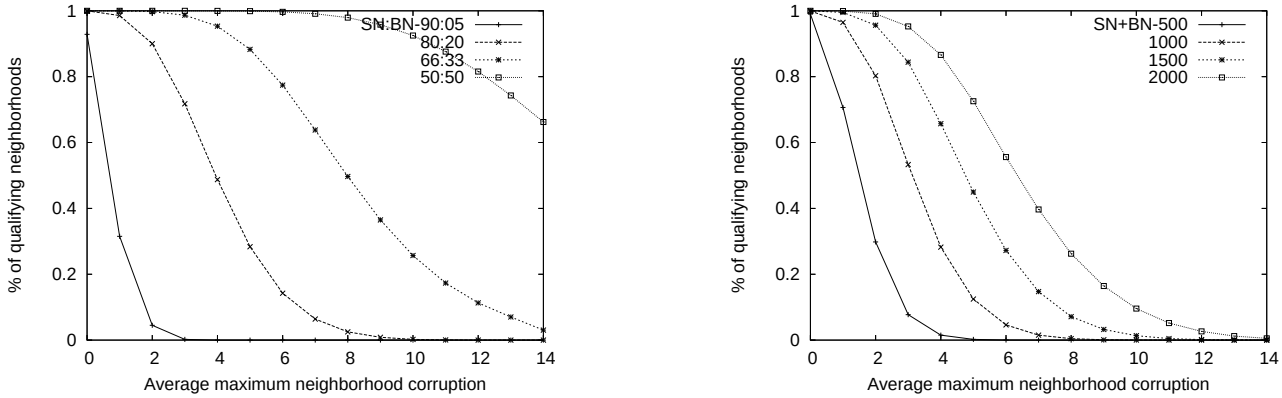
The SN s_i, on receiving the location information ($TLoc_{b_j}$) and NRT_{b_j} broadcast by its beacon neighbor b_j, after a certain time-out delay, assumes all BNs have answered and then tabulates results as follows. s_i first constructs $N(s_i)$ using Algorithm 1B. Then, for each b_j in $N(s_i)$, it counts the number of $+ve_j$ votes and the number of $-ve_j$ votes, storing them in a table called Trusted Beacon Neighbor (TBN) similar to NRT (see Table III). Then, finally location information from remaining beacon neighbors is used to calculate its location. Once the location is computed, the TBN is flushed to free up memory since the BNs are already keeping track of long term reputation. In equations 1, 3, and 4, μ_1, μ_2, and μ_3 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 most recent behavior.

V. Analysis

A. Common Neighbor(s) Set Requirement

The extent to which SNs can withstand collusion depends on the size of their common neighbor(s) set. For a SN to withstand a collusion of k malicious BNs in its neighborhood, its common neighbor(s) set with any beacon neighbor should have at least $2k$ BNs. Therefore, the robustness and performance of DRBTS depends on the size of the common neighbor(s) set. Any violation of the size requirement of the common neighbor(s) 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 III. 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 will be presented in section V-B. Let $N(s_i)$ and $N(b_j)$ represent the neighbor set of SN s_i 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



(a) Varying SN:BN

(b) Fixed SN:BN

Fig. 3. (a) SN:BN ratio varied with fixed n=1000. (b) n varied from 500-2000 in steps of 500 with SN:BN ratio fixed at 80:20.

will now look at how the size of $C(s_i, b_j)$ affects the performance of the system and what is the minimum size of $C(s_i, b_j)$ needed to defeat a collusion of k malicious nodes. We know that

$$C(s_i, b_j) = N(s_i) \bigcap N(b_j) \tag{5}$$

The value of $C(s_i, b_j)$ 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 are in $N(s_i)$. However the chances that this scenario occurs is very unlikely. In any event, the system must have, on average, a minimum of $2k$ beacon nodes in $\forall j, C(s_i, b_j)$ for a completely robust system. Therefore, the equation below gives the necessary and sufficient condition for a robust system.

$$\forall_{i,j}, C(s_i, b_j) = 2k \tag{6}$$

B. Simulation and Results

Our simulation has been done on a custom-built Java simulator. The simulation results show how different variables affect the common neighbor(s) 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 give an idea of the system's tolerance of failing nodes.

For each trial in the simulation, a field of 100mx100m 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(s) sets. 500 trials were performed and averaged together for statistical stability. We plot the misbehavior density against the percentage of SNs that have sufficient neighbors to withstand collusion in their neighborhood.

In Figure- 3(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, 66:33, and 50:50 were tested, and the system performed the best with 50:50 ratio. However, ratios of 80:20 and 66:33 also performed very well. With an 80:20 ratio, 50% of SNs can withstand a collusion of $k = 5$ malicious node in their neighborhood where as in 66:33 ratio they can withstand up to $k = 8$. It is evident from the results that higher the number of BNs the more robust DRBTS gets.

Similarly, we have studied the impact of the total number of nodes on the robustness of DRBTS. The results are presented in Figure- 3(b).

C. 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 publishing the NRT with location transmission, we mitigate the communication overhead to a large extent and almost reduce it to as much in the base model [6]. Also, using a network-wide group key over the pairwise keys, as used in [6], compensates for the memory overhead introduced by the storage of NRTs and TBNs. Additionally, because the reputation system is distributed, there is no information bottleneck at the base station and there is less network traffic.

VI. 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 little overhead, compared to similar schemes.

In our future work, we will examine more complex models for reputation, such as Beta distribution. We will also examine how DRBTS can be adapted to counteract a broader range of malicious behavior. We will also like to investigate the possibility of using BNs as cluster heads for routing purposes by extending the proposed DRBTS. This will enhance the life of SNs thereby enhancing the overall system lifetime. we would also conduct more exhaustive simulations to confirm the robustness of of our system.

VII. Acknowledgements

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

References

[1] 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.

[2] 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.

[3] 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.

[4] 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.

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

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

[7] 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.

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

[9] 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)*.

[10] 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.

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

[12] 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.

[13] Sonja Buchegger, Jean-Yves Le Boudec. Self-policing mobile ad-hoc networks by reputation systems. *IEEE Communications Magazine*, July 2005.

[14] M. Carbone, M. Nielsen, and V. Sassone. A formal model for trust in dynamic networks. *BRICS Report* RS-03-4, 2003.

[15] 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.

[16] 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

[17] 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