SECURITY AND COMMUNICATION NETWORKS
Security Comm. Networks (2010)
Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/sec.160

SPECIAL ISSUE PAPER

Secure reputation monitoring system—a novel connected dominating set-based framework for WSNs

A. Srinivasan[1]*, F. Li[2] and J. Wu[3]

[1] Pennsylvania Center for Digital Forensics, Bloomsburg University, Bloomsburg, PA 17815, U.S.A.
[2] Department of Computer, Information, and Technology, Indiana University-Purdue University Indianapolis, Indianapolis, IN 46202, U.S.A.
[3] Department of Computer and Information Sciences, Temple University, Philadelphia, PA 19122, U.S.A.

ABSTRACT

Reputation and Trust-based Monitoring Systems (RTMSs) have provided a ubiquitous framework for secure Wireless Sensor Network (WSN) computing. However, neighborhood monitoring, though useful, consumes substantial amounts of sensor resources. Therefore, employing sensors for neighborhood monitoring, which is secondary to their intended duties, depletes valuable and scarce resources. This eventually leads to shortened network lifetime, which is counter-productive in WSNs. In this paper, we propose a novel, Connected Dominating Set (CDS)-based reputation monitoring system. Our model is the first attempt to employ a CDS-based monitoring backbone to securely aggregate the reputation of sensors without subjecting them to energy depletion or reputation pollution attacks encountered in existing reputation monitoring systems. Secure and certificateless node mobility and robustness to node replication and ID spoofing attacks are two vital by-products of our model. We confirm the performance of our model *via* simulation studies. Copyright © 2010 John Wiley & Sons, Ltd.

KEYWORDS

adversary; attacks; connected dominating set (CDS); monitoring; mobility; reputation; security; wireless sensor networks (WSNs)

***Correspondence**

A. Srinivasan, Pennsylvania Center for Digital Forensics, 227 Ben Franklin Hall, Bloomsburg University, Bloomsburg, PA 17815, U.S.A.
E-mail: avinash@bloomu.edu

1. INTRODUCTION

In the past two decades, Reputation and Trust-based Monitoring Systems (RTMSs) have provided a ubiquitous framework for secure Wireless Sensor Network (WSN) computing by capitalizing on the openness of the wireless transmission medium. In the existing RTMSs, typically each node is equipped with a *watchdog* that operates in a promiscuous mode for gathering information about the behavior of neighboring nodes. Numerous RTMSs have been proposed for encouraging cooperation and mitigating both selfish and malicious misbehavior of nodes in Mobile *ad hoc* Networks (MANETs) [1] and Peer-to-Peer (P2P) Networks. However, the situation in WSNs is quite unique and one-of-a-kind. Sensors are highly energy-constrained and their autonomous operation in hostile, unattended territories renders them vulnerable to physical node capture attacks. Consequently, cryptography alone cannot ensure security in WSNs, since the adversary can extract all the information stored onboard the captured node, including the cryptographic seeds, keys, etc. This scenario is commonly referred to as an *insider attack*,* in which the adversary, represented by the captured node, becomes a legitimate member of the network. Currently, References [2] and [3] are two known works that counter the insider attacks in WSNs.

Under the operational conditions in which the nodes build their reputation from scratch after deployment, it can take a considerable amount of time before the system is bootstrapped to the operational threshold employing the existing RTMS framework. This can be detrimental in WSNs, since sensors are highly energy-constrained. In particular, until the RTMS can take over as the primary system, a back-up mechanism has to be in place for the system to function and to generate network activity that

* Insider Attacks is defined and discussed in detail in Section 3.1.

enables the monitoring system to build reputation for all the nodes. However, using the same sensors for monitoring as well as intended network services is counter-intuitive. We believe resource-constrained sensors should be used only for required services such that the network lifetime can be prolonged as much as possible. Therefore, using a set of nodes exclusively for neighborhood monitoring will be very productive and can greatly enhance the WSN lifetime.

Building reputation based solely on direct observation can be very time-consuming, though more accurate and can avoid reputation pollution and information asymmetry attacks. Therefore, information sharing is vital for faster convergence of the system. Information sharing is also very useful in having a more consistent local view. However, information sharing can be fatal to the system if nodes resort to a *tit-for-tat* attitude, which pollutes the reputation values. A tit-for-tat attitude is one in which a node u, on receiving a low rating from node v, decrements its rating of v as a retaliation.

In light of the above discussion, we propose a novel, CDS-based monitoring system to function as the monitoring backbone, thereby discharging sensor nodes from their neighborhood monitoring obligations. Our model also prevents nodes from polluting neighborhood reputation. The monitoring backbone is formed by a set of special nodes referred to as monitor nodes, which we will discuss further in Section 3.3. In our model, each sensor is under the jurisdiction of a single monitor node referred to as the *manager node*. Sensor nodes maintain reputation values for all the nodes in their neighborhood, which are provided to them by their manager node. If a node belongs to the jurisdiction of more than one monitor node, then the manager node with the highest priority, which is the node ID in our case, will be its manager. Alternately, other values like node degree, remaining battery power, etc., can also be used to serve as the node priority value when deciding on which manager node to join. The priority value used will always be global such that all nodes use the same priority value when deciding on the Manager node.

When a monitor node goes to sleep, coverage of the network is affected, which in turn jeopardizes the security of the entire system leaving holes in certain areas of the network making it convenient for the adversary to evade detection. This issue can be addressed by adding new monitor nodes to the CDS to collectively cover the jurisdiction of the sleeping CDS-Monitor node. The CDS-Monitor node intending to sleep sends out a copy of its gathered information to all the neighboring CDS-Monitor nodes. Now, when new monitor nodes are added to the CDS, they will have at least one CDS-Monitor neighbor of the sleeping CDS-Monitor node as their neighbor from whom they get the information it gathered prior to sleeping. This enables the newly added CDS-Monitor node(s) to continue monitoring the neighborhood from the point where the sleeping node left off. In effect, the new CDS-Monitor nodes are bootstrapped to the current state of the system saving them both time and energy. However, the challenge in our model lies in maintaining and reconstructing the CDS when one or more nodes go to sleep. Also, deciding on how much information to share with each neighboring monitor node prior to sleeping, to keep the redundancy to a bare minimum can be challenging.

Note that, it is possible that a CDS-Monitor node fails suddenly without sharing the information its has garnered with its neighboring CDS-Monitor nodes. While addressing this issue in detail is beyond the scope of this paper, we would like to briefly discuss a feasible solution. One way to address this problem is to have the CDS-Monitor nodes monitor each other along with monitoring sensor nodes through frequent *Hello* message exchanges. Consequently, in case a CDS-Monitor node fails, it can be detected quickly and replacement monitor nodes can be added without delay.

Additionally, the CDS-Monitor nodes can be configured to exchange the information that they have gathered at fixed intervals so that in case of sudden failure of a CDS-Monitor node, the newly added monitor node(s) can be given the information that was last exchanged by the failed CDS-Monitor node. While this may force the newly added monitor node(s) to start from a state that is not current, it certainly is better than starting from the scratch. Also, how close to the current state of monitoring the newly added monitor node(s) can start in case a CDS-Monitor node fails depends on the frequency of information exchanges between the CDS-Monitor nodes. While a very high frequency can increase network traffic and cause redundancy, a very low frequency means in case of a CDS-Monitor node failure the newly added monitor node(s) start(s) from a very old state. We shall propose a detailed solution to this problem with simulation studies and analysis in our future work.

Side stepping the above discussion, node mobility has proved to be very useful in increasing security [4], reducing uncertainty [5], improving coverage [6], etc. However, in RTMS, node mobility poses a new challenge which can be addressed by answering the following fundamental question:

- How does a node securely transfer its accumulated reputation to the new location?

If the answer to the first question is no, then there is no problem that needs to be addressed other than the down side that the node has to build its reputation from scratch in the new location. This can be very time consuming during which period the node cannot participate in regular network activities. In such scenarios, mobility will be counter-productive in RTMSs. However, if the answer to the first question is 'yes', then we need to answer the second question below.

- How does a node securely transfer its accumulated reputation to the new location?

The answer to second question has to be more comprehensive since during transferring the reputation, the node itself is vulnerable to physical capture whereas the reputation

values are vulnerable to numerous attacks including fabrication attacks, injection attacks, modification attacks, etc. In our model, this problem is addressed in an effective way. Whenever a sensor moves to a new location, the new manager of the node will ask the node's previous manager about its behavior since manager node is the node that is monitoring the behavior of nodes in its jurisdiction. The previous manager then provides the requested node's accumulated reputation to the new manager who will then disseminate it in its jurisdiction. This enables the node to move to a new location without having to transfer its existing reputation *via* cumbersome cryptographic certificates or build its reputation from scratch. We will discuss this in detail in Section 6.1.

Following are some salient features of our model proposed in this paper that summarizes our contributions:

- A CDS-based reputation monitoring system has been proposed for the first time. The proposed model significantly reduces system convergence time.
- Our model mitigates the burden of computation and communication overhead on energy-constrained sensors by discharging them from reputation monitoring and processing obligations.
- Our scheme can detect as well as thwart node replication and ID spoofing attacks very effectively.
- The proposed model ensures certificateless node mobility by securely and efficiently bootstrapping a mobile sensor node in its new location. This makes mobility a lightweight process.
- The proposed model is robust to reputation pollution caused by either information asymmetry attacks or a tit-for-tat attitude of nodes.
- We evaluate the performance of our model through simulation studies.

The remainder of this paper is organized as follows: in Section-2 we summarize related work. In Section-3.1 we present background information on reputation monitoring systems as well as connected dominating sets. We also present the motivation and underlying assumptions of our proposal in this section. In Section-4, we discuss the monitoring backbone in detail and in Section-5 we discuss the maintenance of the monitoring backbone. In Section-6, we discuss the benefits of our model and its robustness to attacks. In Section-7 we present the simulation environment and discuss the results in detail. Finally, we conclude the paper in Section-8 with directions for future research.

2. RELATED WORK

Numerous RTMSs, such as CORE [1], RFSN [2], and DRBTS [3] have been developed to stimulate node cooperation. In most of these models, nodes build their own view based on personal observations as well as recommendations from neighbors. Michiardi and Molva [1] 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 of a node.

The reputation bootstrap problem has been presented as a key challenge for the aforementioned systems in Reference [7], and extended bootstrap periods for newcomer nodes are considered to be burdensome. With mobility in Reference [4], these problems become amplified and grave due to the dynamics of network topology in each region. Consequently, a distributed framework is vital for nodes to securely transfer their accumulated reputation values to the target region and mitigate the newcomer problem. Furthermore, without reputation dissemination for bootstrapping an incoming node, nodes in the target region will have no knowledge regarding the incoming node and treat it as a newcomer.

In Reference [8], Brainard *et al*. introduce the concept of vouching as a tool for on-line authentication. The AVA authentication scheme proposed in [9] is an extension of Reference [8]. AVA has special nodes called Ambassadors, which are selected and dispatched according to several different criteria to represent their home region and perform node authentication. Any node that intends to move into a new region searches for its ambassador in the new region and takes the ambassador's authentication for transferring its reputation. But in AVA the adversary can compromise the ambassador nodes to launch its attacks and the authors have not addressed this problem. Also, AVA is vulnerable to node replication attacks. However, in our model, since tamper-proof monitors are used for authenticating, the adversary has no way of playing foul.

3. PRELIMINARIES

In this section, we provide a brief overview of both reputation monitoring systems and connected dominating sets. We also present the motivation of our work and list the underlying assumptions.

3.1. RTMS overview

In an RTMS, nodes monitor behavior of other nodes in the neighborhood using a *watchdog*. There are two kinds of information available to a node in an RTMS: *firsthand* and *secondhand*. The information gathered by a node by virtue of direct observations using the watchdog is referred to as *firsthand* information. Direct observations are recorded in the tuple (α, β) in which α and β are two positive shape parameters both greater than zero. α denotes a good encounter where as β denotes a bad encounter. The tuple (α, β) is then converted into a reputation value using the Beta distribution function $Beta(\alpha, \beta)$ [10].

Firsthand information is the most reliable piece of information available to a node. However, if nodes are allowed to build reputation based solely on firsthand information,

it can be very time consuming before the system can be bootstrapped to a stable state. Hence, nodes are encouraged to publish their findings in their neighborhood, which when used by other nodes is referred to as *secondhand information*. For instance, consider a neighborhood with four nodes $\{A, B, C, D\}$. Here, information published by node A, becomes secondhand information for nodes B, C, and D.

However, there is a downside to employing secondhand information for building reputation values. It is obvious that not all nodes in any given neighborhood have the same knowledge on the behavior of all nodes and this scenario is referred to as *information asymmetry* in which two nodes exchanging information about a third node may have varying opinions. A malicious node can take advantage of this information asymmetry and publish incorrect values with the intention of misleading the neighborhood and thereby polluting the reputation values. Therefore, nodes in a network usually perform a deviation test before accepting secondhand information published by other nodes to mitigate such information asymmetry attacks ([3]) causing reputation pollution. When a decision has to be made for choosing a partner for any network activity, nodes use the accumulated reputation values to choose the most trustworthy neighbor. Interested readers may refer to Reference [11] for further details on RTMSs.

3.2. CDS overview

In an unweighted, undirected graph $G = (V, E)$, where V is the set of vertices and E is the set of edges, a node u dominates another node v if and only if $u = v$ or u and v are adjacent. Let the CDS of G be a set of vertices $V_{\text{CDS}} \subset V$.

Definition 1. *A connected dominating set of a graph $G = (V, E)$ is a set of vertices $V_{\text{CDS}} \subset V$ such that, for every vertex $v \in V - V_{\text{CDS}}$, there is at least one vertex $u \in V_{\text{CDS}}$ that dominates v, such that V_{CDS} is connected.*

In Figure 1(b), the set $\{1, 2, 4, 7, 8, 9, 10, 11, 13, 14\}$ forms the CDS. Once the CDS is obtained, Dai and Wu's *Rule-k* algorithm [12] is applied to reduce the size of the CDS as in Figure 1(c) resulting in $\{4, 7, 8, 9, 13, 14\}$. A node u from the CDS can be unmarked (pruned) to reduce the size of the CDS if u is completely covered by a subset of its neighbors N' and the following conditions are satisfied:

1. Subgraph induced by N' is connected.
2. Each neighbor of u is adjacent to at least one node in N'.
3. All nodes in N' have a higher priority than u.

In our example, when *Rule-k* is applied to the CDS in Figure 1(b), a smaller CDS (40% smaller), as shown in Figure 1(c), is obtained. A CDS can be constructed using the vertex ID, vertex degree, remaining battery power, or a combination of any of these as the node priority value. The CDS constructed in Figures 1(b)–(c) is based on vertex ID as node priority value.

3.3. Motivation and assumptions

The CDS of a network is a set such that every node in the network is either in the set or a neighbor of one or more nodes in the set. This is a highly desirable property for monitoring systems because, by having the CDS nodes monitor the nodes in their jurisdiction, we can monitor the entire network with very little resource expenditure. Also, partial views of monitoring (CDS) nodes can be easily and quickly merged to form a global view since the number of nodes monitoring is small and well connected. Following are the underlying assumptions of our proposal:

- Node failure is assumed to occur when a node goes to sleep to ensure safe extraction of information.
- Sensor nodes are mobile and have limited resources.
- Monitor nodes are static and can perform all functions of a sensor node. They have large storage and processing capacities and are equipped with a watchdog for monitoring. Monitor nodes cannot be tampered with and are always trusted.
- No new nodes are added to the network after initial deployment.
- We assume that a CDS of the network exists at all times and under any given scenario.

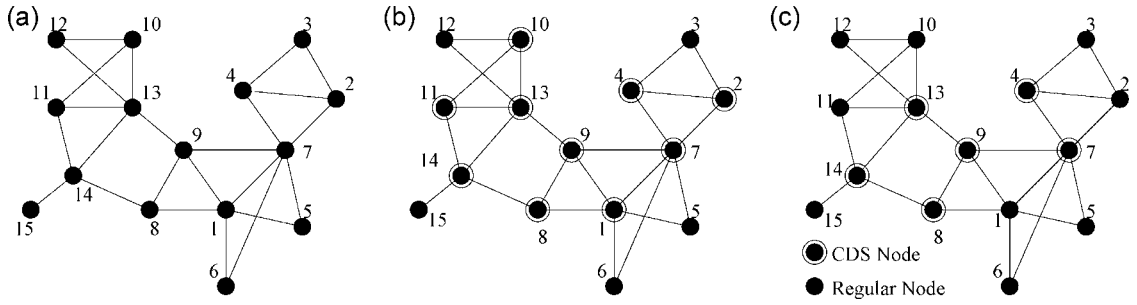

Figure 1. (a) A network of 15 nodes. (b) CDS of the network before applying *Rule-k*. (c) Reduced CDS of the network after applying *Rule-k*.

4. THE MONITORING BACKBONE

There are two types of nodes in our model: *sensors* and *monitors*. We consider a network with S sensors and M monitors. Each node in the network is randomly assigned a unique ID prior to deployment. The IDs assigned to the two types of nodes are drawn from different pools to ensure ID uniqueness. The monitoring backbone, which is a CDS, is constituted by monitor nodes hereinafter referred to as CDS-Monitors. The neighborhood of a node consists of three sets of nodes: sensors, CDS-Monitors, and Monitors. The sensor nodes are the nodes that perform regular network activities. CDS-Monitors are nodes that monitor the sensor nodes. Finally, Monitor nodes are backup nodes that will be added to the CDS when a CDS-Monitor node goes to sleep. The information gathered through watchdog monitoring is referred to as *firsthand* information and the information gathered by virtue of information sharing is referred to as *secondhand* information or *recommendation*. In our system, the information is primarily firsthand, and secondhand information is used for bootstrapping sensors when they move to a new region, which we will discuss in detail in Section 6.1.

If a node belongs to the jurisdiction of more than one CDS-Monitor, then the one with the highest priority will be its manager. For example, in Figure 4(b), node 2 belongs to the jurisdiction of CDS-Monitors A and B. Based on node priority, assuming priority $A >$ priority B, A will be 2's manager. In our model, we do not make any assumptions regarding the state of the monitor nodes, i.e., nodes are free to sleep and wake-up according to the underlying scheduling algorithm. A detailed discussion on node wake-up and sleep scheduling is beyond the scope of this paper. Interested readers may refer to References [13] and [14] for an in-depth discussion and for further references.

When nodes go to sleep, coverage of the network is affected, which in turn jeopardizes the security of the entire system. This issue can be addressed by adding new monitor nodes to the CDS to collectively cover the jurisdiction of the CDS-Monitor intending to sleep. However, if the newly added CDS-Monitors were to start building reputation of nodes in their jurisdiction from scratch, then it could take a significant amount of time for the neighborhood to be bootstrapped to the operational threshold. To overcome this problem, the CDS-Monitor intending to sleep sends out a copy of its gathered information to all the neighboring CDS-Monitor. This enables the newly added CDS-Monitor(s) to continue monitoring the neighborhood from where the sleeping node left.

4.1. Reputation computation

In RTMSs, where there is information sharing, nodes are vulnerable to reputation pollution, either due to information asymmetry attacks or due to nodes adopting a tit-for-tat attitude. This kind of attitude among nodes leads to pollution and inconsistency in reputation values and eventually to instability of the entire system. These problems can be overcome by using the monitoring backbone proposed in this paper for both garnering observations and reputation computation.

In our model, CDS-Monitors constantly update their observation parameters α and β for each sensor node in their jurisdiction, which is then converted into a reputation value. The CDS-Monitors then disseminate the reputation values in their jurisdiction using which the sensor nodes update their local view. Frequency of information dissemination by CDS-Monitor nodes can be tuned according to the requirements of the application domain. Since a malicious sensor node cannot harm the CDS-Monitor for publishing low reputation values, there is no room for it to adopt a tit-for-tat attitude. The main advantage of this model is that there is no computation overhead imposed on sensor nodes, which preserves scarce resources. If a sensor node has a neighbor which belongs to the jurisdiction of a different CDS-Monitor, then it requests its manager to provide it with the required information. The requesting node's manager then contacts the requested node's manager to obtain this information and provides it to the requesting node.

For instance, in Figure 2, when node 4 needs the reputation of node 3, it makes a request to its manager B. B, in turn, makes a request to A, who is the manager of node 3. A then provides the computed reputation value of node 3 to B, which then provides it to node 4. Though this process involves a couple of extra hops of information transmission, note that from the sensor nodes' perspective, it is still a two-step process: request and response, although it has a slightly higher delay.

Our reputation computation model is simple yet robust against information asymmetry attacks. Information asymmetry attacks specific to reputation monitoring systems were introduced by Srinivasan

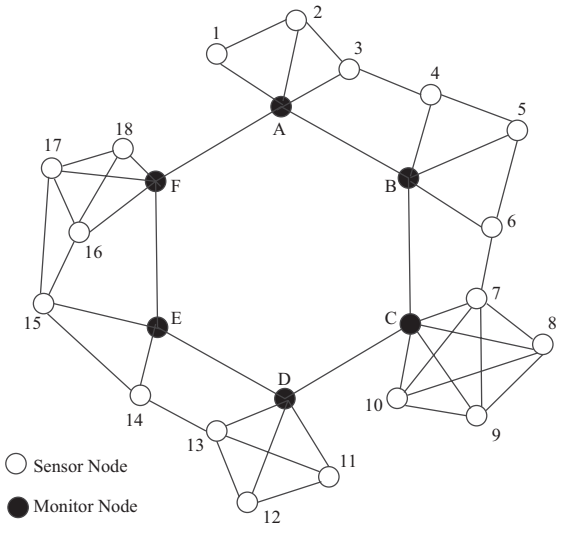

Figure 2. Schematic diagram representing our model on a network of 18 sensor nodes and 6 monitor nodes.

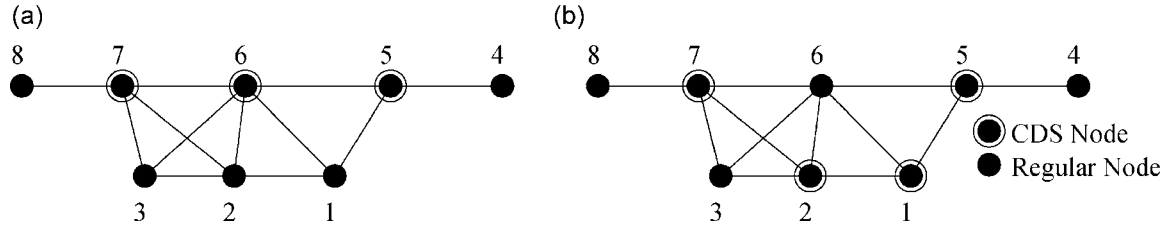

Figure 3. (a) CDS of network with seven nodes after applying *Rule-k*, (b) CDS of network after node 6 sleeps.

et al. in Reference [3], in the context of beacon-based sensor localization. In the model proposed in this paper, each monitor node uses its own observation for computing the reputation of every node in its jurisdiction. Since only firsthand information is used, it prevents reputation pollution arising due to information asymmetry attacks.

Algorithm 1 CDS Maintenance

for each CDS-Monitor node i intending to sleep **do**
  Send copy of accumulated reputation values to neighboring CDS-Monitor nodes;
  $V_{CDS} \leftarrow V_{CDS} - i$
end for
for each Monitor node j that is a neighbor of i or a neighbor of CDS-Monitor neighbors of i **do**
  $V_{CDS} \leftarrow j$
end for
Execute Localized *Rule-k* Algorithm on new V_{CDS} to prune redundant CDS-Monitor nodes;
Update new CDS-Monitors in V_{CDS} with reputation values of sensors in their jurisdiction;
Update sensor nodes of their new manager nodes;

5. MAINTENANCE OF MONITORING BACKBONE

Since sensor nodes are not static, the CDS has to be updated whenever there is node movement. The CDS has to be recomputed even when CDS-Monitors go to sleep. There are two important issues that need to be addressed for recomputing the CDS:

1. Are there sufficient monitor nodes to cover the entire network even if some of the current CDS-Monitor nodes go to sleep?
2. When a CDS-Monitor node goes to sleep, how does it distribute the information it stores among the neighboring CDS-Monitor nodes?

To address the issue raised by the first question, we assume that a sufficiently large number of monitor nodes are uniformly distributed during the initial deployment along with regular sensor nodes. This ratio of monitor to sensor modes is determined empirically such that the system can function smoothly at all time. The empirical results are presented in Figure 6(a)–(c). This guarantees that we will be able to find a monitor node to replace an existing CDS-Monitor when it goes to sleep or when a sensor node moves to a new location. Addressing the challenge raised by the second question is much trickier. When a node goes to sleep, the number of monitor nodes that are required to cover its jurisdiction is not fixed. The best scenario is when a single monitor node is needed to cover the jurisdiction of the CDS-Monitor going to sleep.

Therefore, when a CDS-Monitor is about to sleep, it provides a copy of its information to its neighboring CDS-Monitors. The reason it chooses to provide a complete copy of its information to all its CDS-Monitor neighbors is because more than one monitor node may be required to cover its jurisdiction. Secondly, the reason it chooses to provide a copy of the information to only its neighboring CDS-Monitors is because the new CDS-Monitor has to re-establish only the broken link(s) between the sleeping CDS-Monitor and its neighbors that are awake. For illustration, in Figure 4(b) according to Algorithm 1, when node A intends to sleep, it sends a copy of its information to CDS-Monitor neighbors B and G. Following this, monitor node H, which is a neighbor of A, is added to the set V_{CDS}. Also, monitor node F, which is a neighbor of both B and G, which are CDS-Monitor neighbors of node A, and monitor node E, which is a neighbor of G, are added to the set V_{CDS}. Following this, the Localized *Rule-k* Algorithm is executed to prune redundant nodes from the set V_{CDS}. Consequently, node E gets pruned from the V_{CDS} and the resulting V_{CDS} is shown in Figure 4(c).

Finally, the two newly added nodes H and F re-establish the broken links (B, A) and (A, G), connecting the two awake CDS-Monitor neighbors B and G of A with three new links (H, G), (G, F), and (F, B). From this example, it is clear as to why it gives a copy of the information to all its CDS-Monitor neighbors. For instance, in Figure 4(b) if node A intends to sleep and gives a copy of its information to only one of its CDS-Monitor neighbors, say B, then the newly added CDS-Monitor H would not have the information since A did not share its information with G. This would necessitate H to monitor the neighborhood building reputation of nodes in its jurisdiction from scratch, which is time consuming and counter-productive. A similar scenario is illustrated in Figure 3.

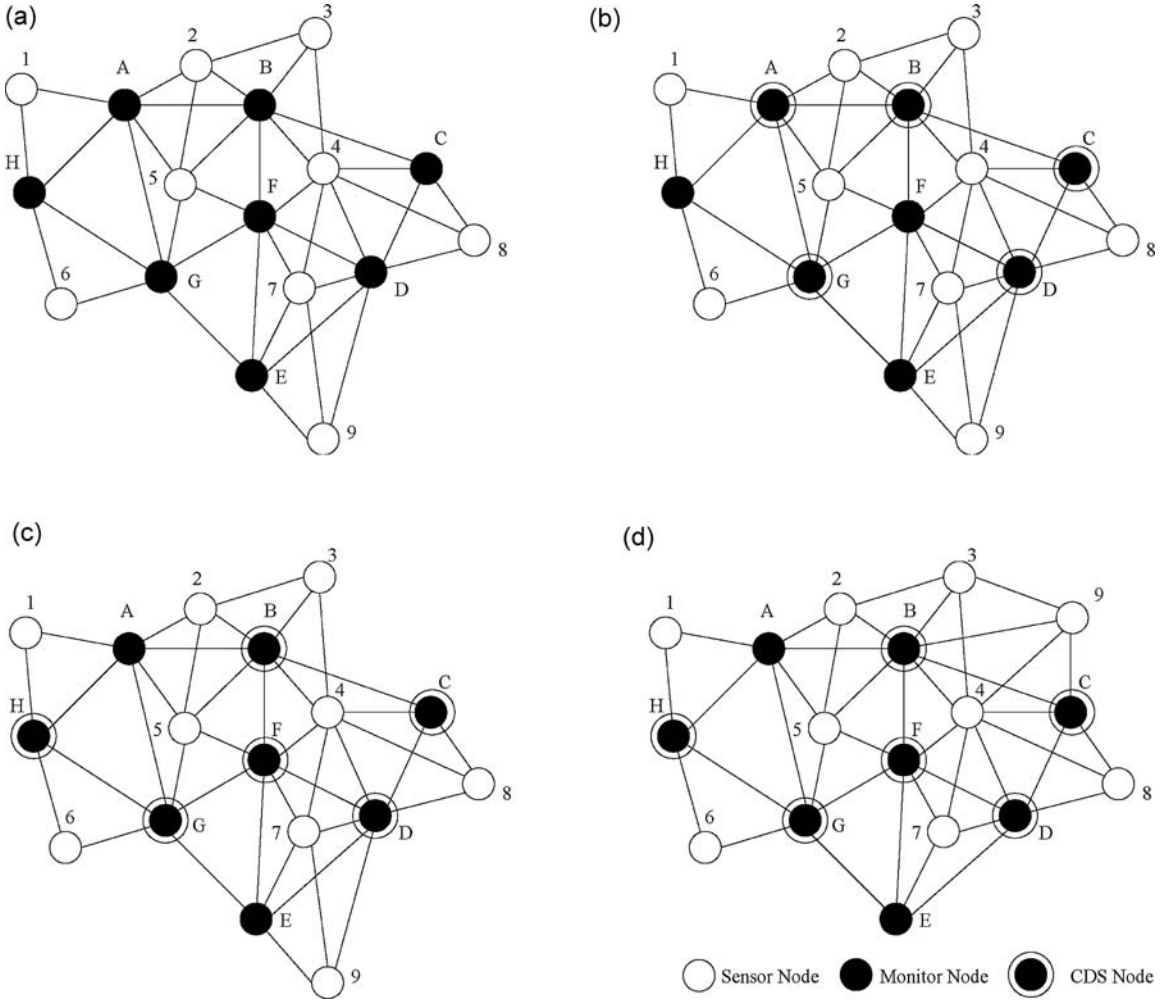

Figure 4. Hollow circles represent ordinary sensor nodes, shaded circles represent the more powerful monitor nodes, shaded circles with an outer band represent the CDS-Monitor, and the square enclosing all the nodes represents the network boundary. Mobility of nodes is confined to the square boundary. (a) Initial network, (b) CDS of the network, (c) updated CDS after monitor *A* sleeps, (d) updated CDS after node 9 moves to a new location and joins monitor *B*.

6. DISCUSSIONS

In this section, we discuss the ability of our model to ensure secure node mobility and the robustness it achieves against ID spoofing and node replication attacks.

6.1. Secure node mobility

Currently, node mobility models under the framework of RTMSs either require the node to collect digital certificates vouching for its reputation or to build its reputation afresh in the new location. The model proposed in this paper neither requires the node to carry digital certificates nor to build reputation from scratch in the new location. In our model, node mobility can be accomplished in one of the following two ways:

1. Reactive bootstrapping
2. Proactive bootstrapping

In the reactive bootstrapping mobility mode, a node simply moves to the new location and joins a new manager node and provides its previous manager's ID. The new manager then contacts the previous manager of the node to verify the node's claim of its previous location as well as collect its accumulated reputation. After authenticating the node and receiving its reputation information from the previous manager, the new manager bootstraps the node in its new location. This enables the node to start participating in

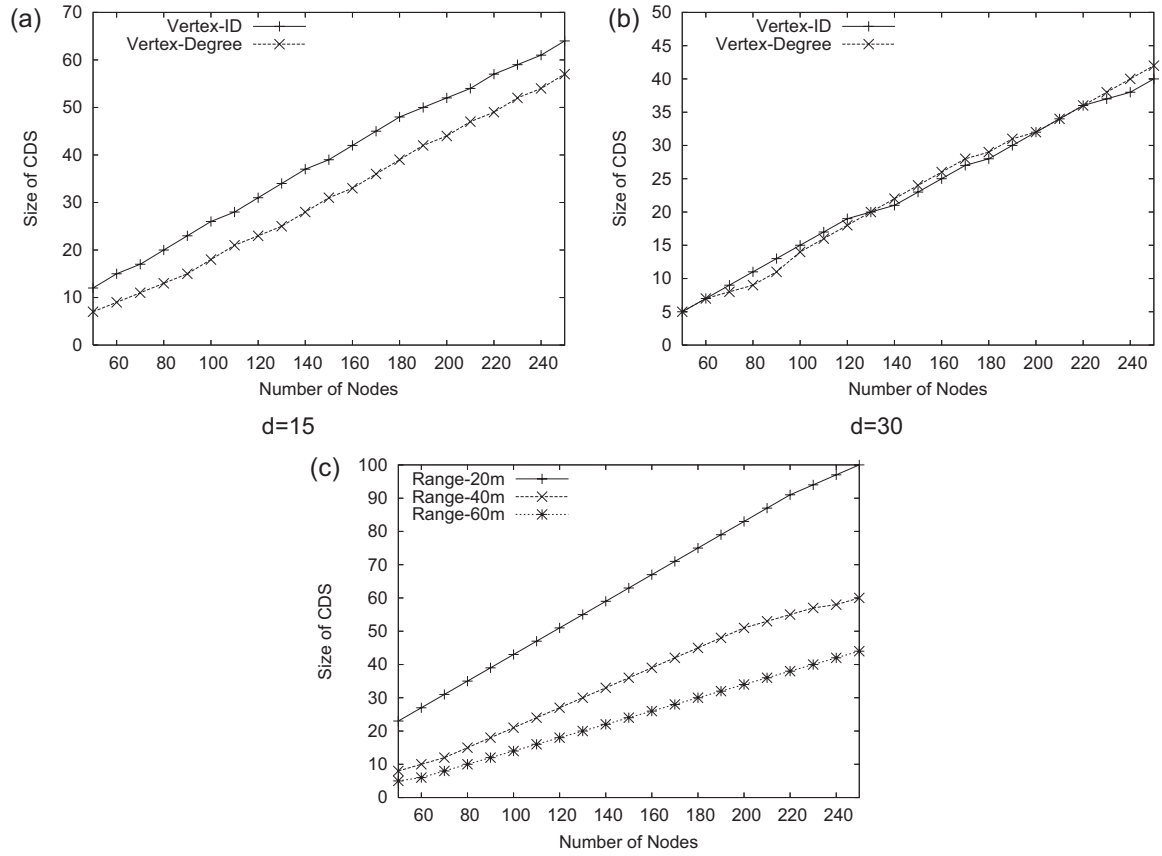

Figure 5. (a) Size of CDS with fixed $d = 15$. (b) Size of CDS with fixed $d = 30$. (c) Size of CDS with range varied from20 to 60 m.

network activities readily and receive cooperation from neighboring nodes. Without this novel framework, the node would have to build its reputation from scratch in the new location which is a time consuming process, during which period the node can neither receive cooperation nor render service to other nodes in the neighborhood since its reputation value is below the operational threshold. This is a tricky situation since reputation of the node cannot be built unless there is some activity but with a low reputation value the node cannot participate in network activities. However, our model overcomes this drawback enabling newcomer nodes in a region to be part of the network activities readily.

In the proactive bootstrapping mobility model, assuming predictive mobility, a node intending to move notifies its manager of its target location. With this information, the manager can communicate with the potential new managers in the target region and provide them with reputation information of the node. Subsequently, when the node moves into the new location, it has to merely provide its previous manager's ID to the new manager and it will be instantly bootstrapped to the reputation state that it had in the previous location. In this model we assume that there is an upper bound on time for the node to move to the new location and join the new manager. Therefore, after notifying the manager, a node has to immediately move out of its current location. Consequently, a node has no time to play foul after its request by causing damage in the current region and then moving to the new location before its updated reputation is reported. Even if such a situation were to arise, since the monitor nodes are closely connected, it would not be long before such malicious nodes are detected and isolated.

In Figure 4, hollow circles represent ordinary sensor nodes, shaded circles represent the more powerful monitor nodes, shaded circles with an outer band represent the active monitor nodes that constitute the CDS, and the square enclosing all the nodes represents the network boundary. Mobility of nodes is confined to within this square region. Figures 4(c) and (d) have captured the ideal of node mobility, in general, well. In Figure 4(c), node 9 belongs to the jurisdiction of monitors D and E, but its manager is D (priority $D >$ priority E). Now, after moving to a new location, node 9 is in the jurisdiction of monitors B and C, and based on priority resolution, B will be its new manager.

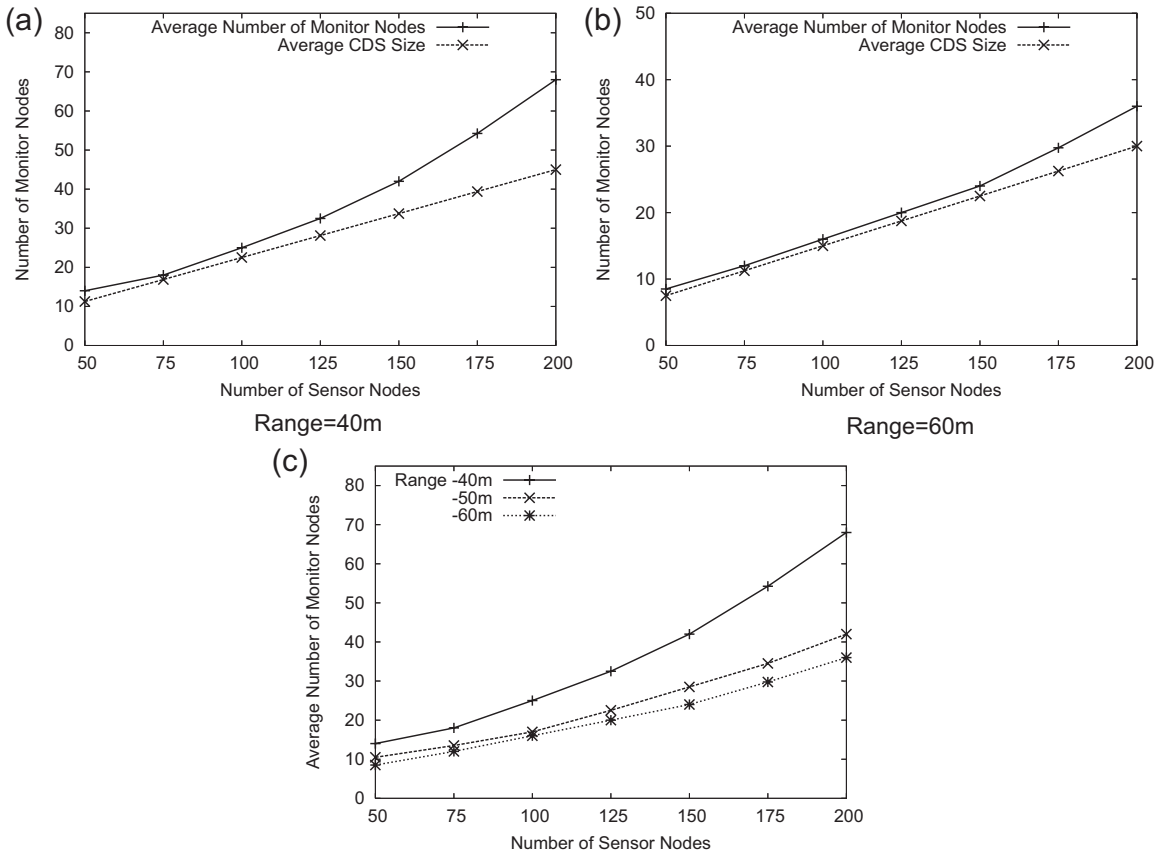

Figure 6. (a)–(b) Comparison of average number of monitor nodes with average CDS size for monitor transmission ranges 40 and 60 m, respectively. (c) Average number of monitor nodes for varying numbers of sensor nodes for varying monitor ranges.

6.2. Robustness to ID spoofing and node replication attacks

Our model curtails ID spoofing attacks very effectively and with little overhead. When a malicious node re-enters the network through ID spoofing, its new manager will ask it to provide its previous manager's ID. When the node fails to provide a valid manager ID, it will be immediately blacklisted and denied cooperation.

In our model, there is no way for the adversary to launch a node replication attack either, since whenever a node joins a network location, it has to furnish its previous manager's ID. With node replication attacks, replicated nodes can provide their previous manager's ID. However, when the previous manager is contacted, if indeed there has been a node replication attack, then the previous manager will still claim its dominance over the node under consideration. Immediately, an alarm will be raised and the corresponding node's ID will be broadcast along the CDS backbone to all monitor nodes and all copies of the node will be blacklisted and isolated.

6.3. CDS-based robust and secure localization

The proposed CDS backbone can be used to serve yet another important purpose in sensor networks- localization. Localization is the process by which a node computes its location. Localization, a very critical task, which if compromised can be detrimental to the entire system. When sensors report events of interest to the base station, if the location of the reporting sensor is reported incorrectly for any reason, then the response process will be futile. Srinivasan, Teitelbaum, and Wu have proposed a reputation and trust-based secure localization protocol in Reference [3]. In Reference [3], sensors are robust to beacon node based information asymmetry attacks during node localization. The same model can be extended to be further robust and secure using the CDS backbone constructed in our model. In this improved model, the CDS-nodes can replace the specialty nodes in Reference [3] that are more powerful compared to ordinary sensor nodes and serve as beacon nodes. With the CDS serving as the localization

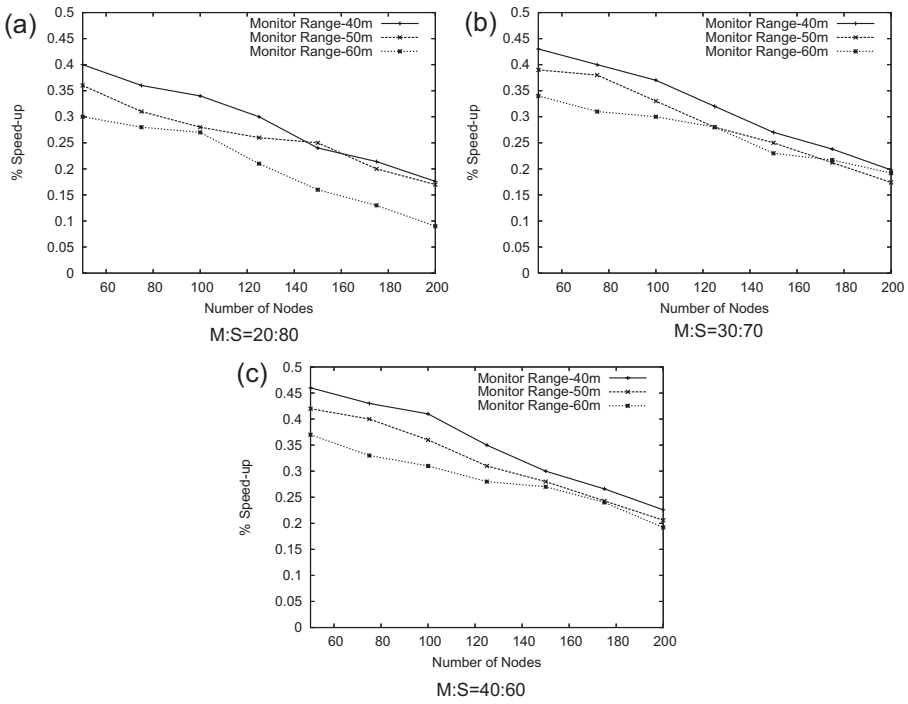

Figure 7. Comparison of speedup gain in performance between reactive and proactive bootstrapping mobility models for varying ratios of monitor and sensor nodes in the network. Mobility speed is 8 m/s.

backbone, the sensor nodes can further conserve energy that they would otherwise expend in helping other sensor nodes compute their location. We shall develop a complete sensor localization model exploiting the CDS property of a network graph, conduct simulation studies and mathematical analysis and present the results in our future work.

As discussed above in Section-6.1, in reactive bootstrapping mobility model, when a node moves to a new location, though it joins a single CDS-node as its manager, other CDS-nodes in its one-hop neighborhood including its manager can send a beacon signal that can help the node localize. Constraints on number of beacon signals required to localize with high accuracy can be addressed by enforcing k-CDS conditions, in which each node has k neighbors from the CDS. This can be achieved using Dai and Wu's k-connected k-dominating set (k-CDS) protocol proposed in [15].

7. SIMULATION

In this section we will first discuss our simulation environment and then present the results.

7.1. Environment

All simulations have been carried out on a custom built, stand-alone, C++ simulator. In our simulations, a sensor field of area $100 \times 100\,\mathrm{m}^2$ has been considered. The network has been modeled as an undirected graph in which the set of vertices and edges have been represented as V and E, respectively. All monitor nodes have a constant transmission range of T_{Range}. The set V consists of two sets of nodes—sensor nodes S and monitor nodes M. In our simulations, S, M, and T_{Range} have been considered as tunable parameters. An edge $(u, v) \in E$ exists between two nodes $u, v \in V$ if u and v lie in each other's transmission range. The results presented have been averaged over 1000 iterations for statistical stability.

7.2. Results

In Figures 5(a) and (b), we have presented results comparing the impact of network diameter on size of the CDS. We have compared the size of CDS obtained with vertex ID and vertex degree as the node priority values for diameter values $d = 15$ and $d = 30$, respectively. We can see that with increase in diameter, the size of the CDS shrinks. We have also studied the impact of density on CDS size by fixing the transmission range at 20, 40, and 60 m and the results are presented in Figure 5(c). It is evident that as the transmission range increases, the size of the CDS shrinks. With the transmission range fixed at 20 m, the size of the CDS is approximately 42% the size of the network. With transmission range fixed at 40 m, the size of the CDS is approximately 22% the size of the network. Finally, with

transmission range fixed at 60 m, the size of the CDS is approximately 15% the size of the network. It is evident that the size of the CDS decreases quickly with increase in transmission range.

In the rest of our simulations, unless otherwise specified, vertex ID is used as the node priority value in constructing the CDS. In Figures 6(a) and (b), we have presented results comparing the average CDS size for varying numbers of sensor nodes with the average number of monitor nodes required to ensure the existence of a CDS for monitor transmission range of 40 and 60 m, respectively. We see that the number of monitor nodes required is lower with a monitor range of 60 m compared to a range of 40 m. Therefore, the average difference between the average CDS size and the number of monitor nodes decreases with increase in transmission range of monitor nodes. However, the average number of monitor nodes grows much faster than the average CDS size with increase in network size.

In Figure 6(c), we have presented results comparing the average number of monitor nodes required for varying number of sensor nodes. We can see that as the number of sensor nodes increases, the number of monitor nodes required increases. However, this number decreases with increase in transmission range of monitor nodes.

In Figure 7(a), we have plotted the speedup gain in performance of the network for a monitor to sensor node ratio of 20:80 for monitor node transmission ranges of 40, 50, and 60 m. Speedup gain in our simulations is computed as the ratio of time taken for bootstrapping nodes in their new location using the reactive bootstrapping model to the time taken using proactive bootstrapping model. We can see that a maximum speed up gain is achieved for a network size of 50 nodes with a monitor range of 40 m. The average speedup gain with a monitor range of 40 m is 29%. Similarly for monitor ranges of 50 and 60 m, the average speed up gain is 26% and 20%, respectively.

Similarly, speedup gain results have been plotted for monitor-sensor ratios of 30:70 and 40:60 in Figures 7(b) and (c), respectively. From the results, we can see that the average speedup gain for monitor–sensor ratio of 30–70 is as follows: for monitor range of 40 m the speedup is 31.8%, for monitor range of 50 m, the speed up is 28.8%,

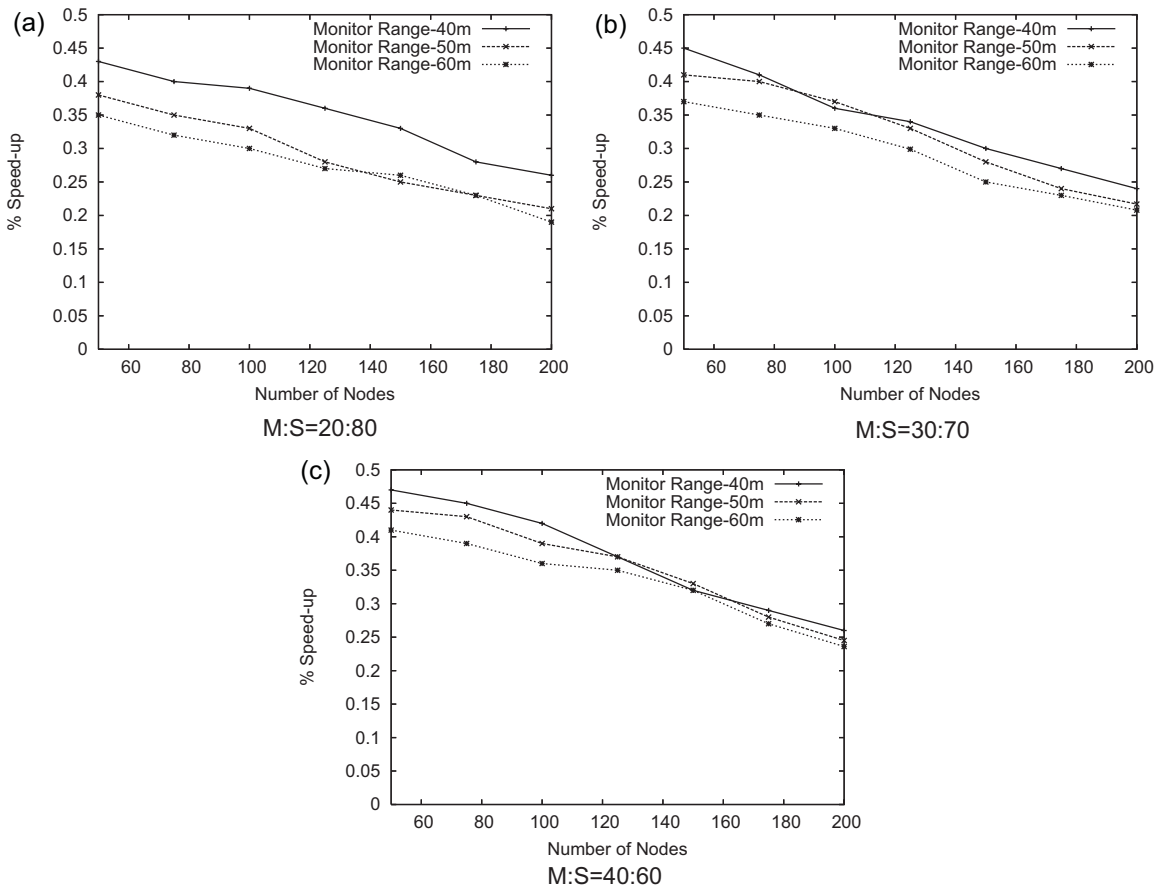

Figure 8. Comparison of speedup gain in performance between reactive and proactive bootstrapping mobility models for varying ratios of monitor and sensor nodes in the network. Mobility speed is 10 m/s.

and the speedup gain for monitor range of 60 m is 26.7%. On average, the speedup gain for monitor–sensor ratio of 30–70 is 29.1%. Similarly, the speedup gain for monitor–sensor ratio of 40–60 is as follows: for monitor range of 40 m, the speedup gain is 34.8%, for monitor range of 50 m the speedup gain is 31.7%, and for monitor range of 60 m the speedup gain is 28.4%. On average, the speedup gain for monitor–sensor ratio of 40–60 is 31.6%.

We can see from the graphs in Figures 7(a)–(c) that the speedup gain increases with increase in monitor ratio in the network. However, speedup gain decreases with increase in monitor range, which can be attributed to increase in neighborhood density consequently delaying reconstruction of CDS.

Similarly, we have plotted the results highlighting speedup in performance and the impact of various system parameters like monitor range, number of nodes in the network, monitor to sensor node ratio, and mobility speed in Figures 8 and 9.

In Figures 10(a)–(c), we have plotted performance speedup against monitor node transmission ranges of 40, 50, and 60 m. We see from the results that speedup in performance drops with increase in monitor transmission range. In Figure 10(a), we have plotted the results for node mobility speed of 8 m/s for monitor to sensor ratios of 20:80, 30:70, and 40:60. From the results its clear that with high monitor node ratios, there is a higher latency that is added to the mobility process leading to a drop in performance speedup when compared to lower Monitor node ratios.

Similarly, in Figures 10(b)–(c), we have we have plotted the results for node mobility speed of 10 and 12 m/s, respectively for monitor to sensor ratios of 20:80, 30:70, and 40:60. We have made similar observations in results as discussed above, and in addition, the speedup in performance increases with higher mobility speeds. This is because, with higher speeds, movement time is cut and thereby compensates for other latencies introduced.

Finally in Figure 10(d), we have presented results plotting Mobility Speed against average percentage speed-up gain for monitor to sensor ratios of 20:80, 30:70, and 40:60. From the results its evident that the speed-up gain increases with increase in monitor node ratio. With increase in mobiltiy speeds, the speed-up gain has no fixed behavior as can clearly be seen from the graph although it tends to

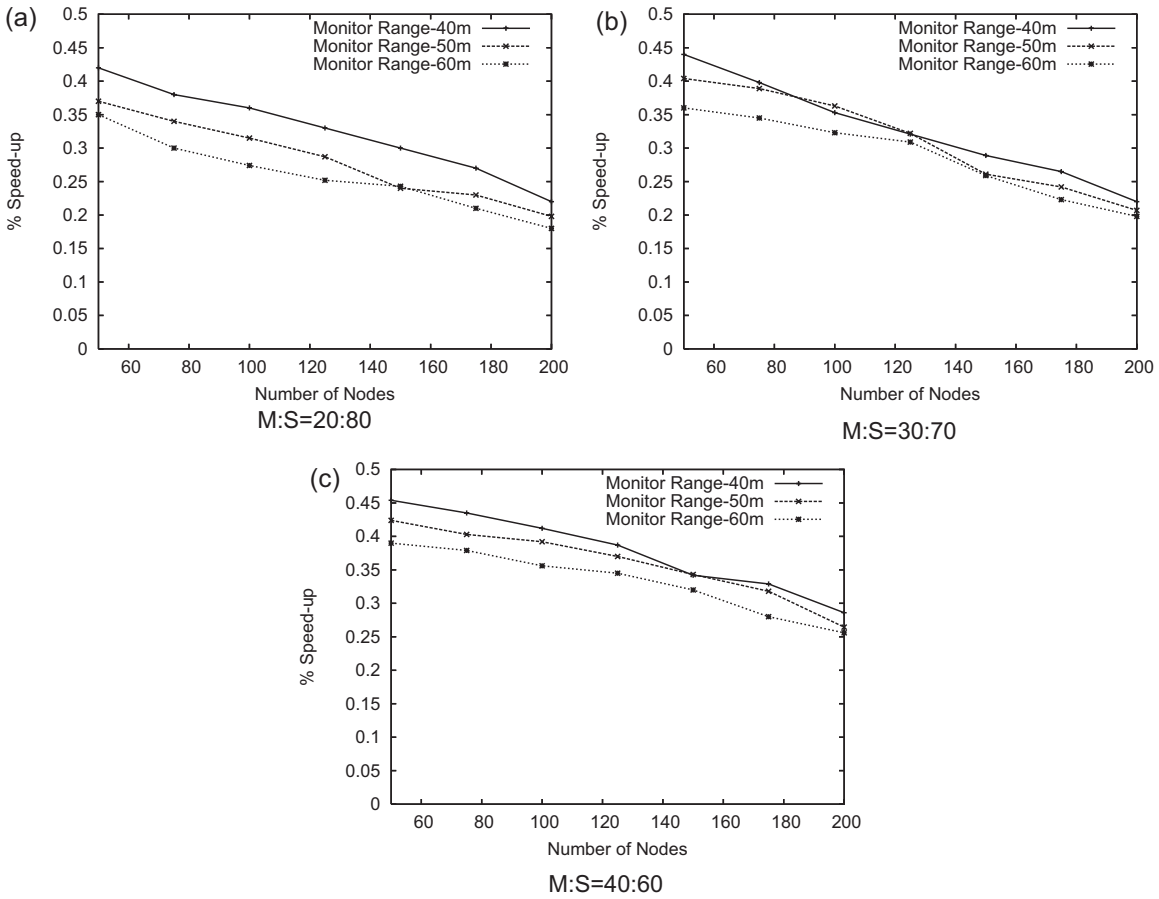

Figure 9. Comparison of speedup gain in performance between reactive and proactive bootstrapping mobility models for varying ratios of monitor and sensor nodes in the network. Mobility speed is 12 m/s.

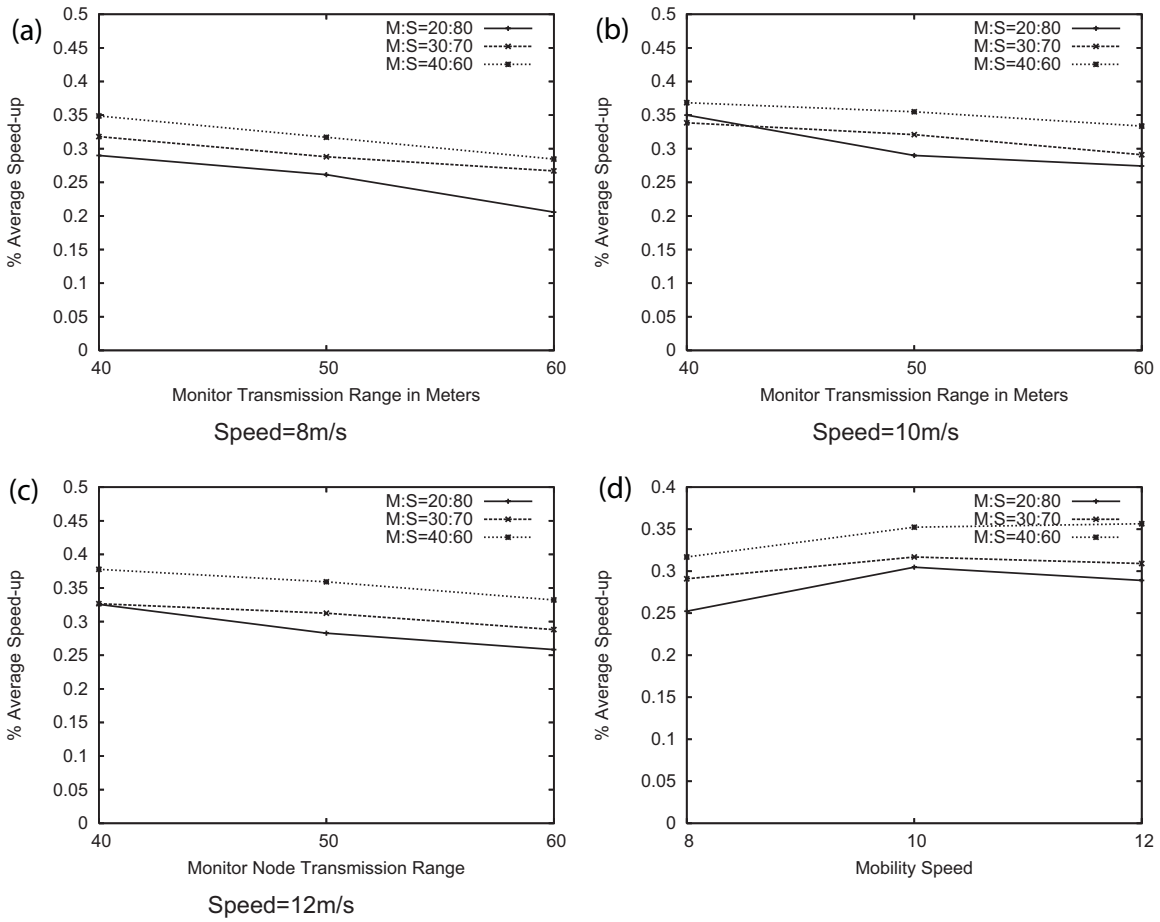

Figure 10. (a) Comparison of average speedup gain in performance between reactive and proactive bootstrapping mobility models for mobility speed 8 m/s averaged for varying Sensor to Monitor ratios. (b) Comparison of average speedup gain in performance between reactive and proactive bootstrapping mobility models for mobility speed 10 m/s averaged for varying Sensor to Monitor ratios. (c) Comparison of average speedup gain in performance between reactive and proactive bootstrapping mobility models for mobility speed 12 m/s averaged for varying Sensor to Monitor ratios. (d) Comparison of average speedup gain in performance averaged for varying Sensor to Monitor ratios, varying mobility speeds, and varying monitor ranges.

increase in the beginning. However, the gain in speed-up with increasing monitor node ratio is much higher when compared to speed-up gain with increase in mobility speeds in the beginning.

8. CONCLUSIONS

In this paper, we have proposed a novel Connected Dominating Set (CDS)-based reputation monitoring system. Our model is the first attempt to employ a CDS-based monitoring backbone to securely aggregate the reputation of sensors without subjecting them to energy depletion or reputation pollution attacks encountered in existing reputation monitoring systems. Secure and certificateless node mobility and robustness to node replication and ID spoofing attacks, arising due to information asymmetry and tit-for-tat attitude of nodes, are two vital by-products of our model. The same model was also extended to serve the purpose of secure and robust node localization. We have confirmed the validity and performance of our model *via* simulation studies.

In our future work, we plan to conduct a more in-depth simulation of our model along with simulating the scenario wherein a node fails abruptly instead of going to sleep and see its impact on coverage and information extraction from the failed node. We also wish to investigate the possibility of using the CDS property to address other security threats in wireless and *ad hoc* networks.

ACKNOWLEDGMENTS

Authors would like to thank the anonymous reviewers who helped in making this a quality paper. Part of this work has

been supported by the Bloomsburg University Research and Disciplinary Grant (Center No. 1012030405) 2009-2010.

REFERENCES

1. Michiardi P, Molva R. CORE: a COllaborative REputation mechanism to enforce node cooperation in Mobile Ad Hoc Networks. IFIP conference proceedings V 228, pp. 107–121. In *Proceedings of the IFIP Tc6/Tc11 Sixth Joint Working Conference on Communications and Multimedia Security. Advanced Communications and Multimedia Security*, 2002.
2. Ganeriwal S, Srivastava M. Reputation-based framework for high integrity sensor networks. In *Proceedings of ACM SASN*, 2004.
3. Srinivasan A, Wu J, Teitelbaum J. Distributed reputation-based secure localization in sensor networks. In *Special Issue of Journal of Autonomic and Trusted Computing*, 2008; **1**.
4. Capkun S, Hubaux J, Buttyan L. Mobility helps security in ad hoc networks. In *Proceedings of ACM MobiHoc*, 2003.
5. Li F, Wu J. Mobility Reduces uncertainty in MANETs. In *Proceedings of IEEE INFOCOM*, 2007.
6. Liu B, Brass P, Dousse O, Nain P, Towsley D. Mobility improve coverage of sensor networks. In *Proceedings of ACM MobiHoc*, 2005.
7. Buchegger S, Boudec J. A robust reputation system for P2P and mobile ad hoc networks. In *Proceedings of Economics of P2P Systems*, 2004.
8. Brainard J, Juels A, Rivest R, Szydlo M, Yung M. Fourth factor authentication: somebody you know. In *Proceedings of ACM CCS*, 2006.
9. Li F, Wu J. Authentication via ambassadors: a novel authentication mechanism in MANETs. In *Proceedings of MilCom*, 2007.
10. Josang A, Ismail R. The beta reputation system. In *Proceedings of 15th Bled Electronic Commerce Conference*, 2002.
11. Srinivasan A, Teitelbaum J, Liang H, Wu J, Cardei M. Reputation and trust-based system for ad hoc and sensor networks. In *Algorithms and Protocols for Wireless Ad Hoc and Sensor Networks*, Boukerche A (ed.). Wiley & Sons, 2008; 375–403.
12. Dai F, Wu J. An extended localized algorithm for connected dominating set formation in ad hoc wireless networks. *IEEE Transactions on Parallel and Distributed Systems* 2004; **15**(10): 908–920.
13. Keshavarzian A, Lee H, Venkatraman L, Lal D, Chintalapudi K, Srinivasan B. Wakeup scheduling in wireless sensor networks. In *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing, Florence (MobiHoc'06)*, May 2006.
14. Lee H, Keshavarzian A, Aghajan H. Multi-cluster multi-parent wake-up scheduling in delay-sensitive wireless sensor networks. In *Proceedings of IEEE International Conference on Global Communications (Globecom'08)*, November 2008.
15. Dai F, Wu J. On constructing k-connected k-dominating set in wireless networks. In *19th IEEE International Parallel and Distributed Processing Symposium (IPDPS'05)*, 2005, pp.81a.