SECURITY AND COMMUNICATION NETWORKS
Security Comm. Networks. (2008)
Published online in Wiley InterScience
(www.interscience.wiley.com) DOI: 10.1002/sec.54

Secure and reliable broadcasting in wireless sensor networks using multi-parent trees

Avinash Srinivasan*,† and Jie Wu
Department of Mathematics, Computer Science and Statistics, Bloomsburg University, PA 17815, USA.

Summary

Wireless sensor networks (WSNs) have been the focal point of research over the last several years. Broadcast communication is a key requirement for WSNs since many tasks in the network depend on broadcasting, including critical tasks like querying. Consequently, securing broadcast communication over sensor networks has become an important research challenge. Typically, broadcast communication involves two steps: broadcasting and acknowledging. In the broadcasting phase, the message is broadcast in the network. In the acknowledging phase, nodes that successfully received the broadcast message send an acknowledgment to the broadcast origination node, which in this paper is always the sink. The terms 'sink' and 'base station (BS)' are used interchangeably throughout this paper. Intuitively, broadcast communication has two important metrics: reliability and security. Though the reliability metric has drawn sufficient attention in the research community, the security metric has not. In this paper, we address both metrics with an emphasis on the former and address the Denial-of-Broadcast Message attacks (DoBM) in sensor networks. We propose a novel multi-parent tree-based model called the *k*-Parent Flooding Tree Model (*k*-FTM). We also present distributed algorithms for the construction of *k*-FTM and prove via simulation and analysis that the proposed *k*-FTM is robust against DoBM. Our Multi-Parent tree model enables the BS to detect DoBM very efficiently, even in the presence of a prudent adversary who focuses on remaining undetected by causing damage below the detection threshold. *k*-FTM is, to our best knowledge, the first fault-tolerant tree model that is both reliable and secure. Through simulations we confirm that our model achieves detection rates close to that of a static tree and a broadcast reliability close to that of blind flooding. Copyright © 2008 John Wiley & Sons, Ltd.

KEY WORDS: adversary; broadcast communication; flood tree; malicious; reputation; security; wireless sensor networks

1. Introduction

Broadcasting has been one of the most important and widely used communication techniques for information dissemination, from heralding in ancient times to the current state-of-the-art communication networks. Broadcast communication, which forms the basis of all communication in wireless networks, operates in two phases: the broadcast phase and the acknowledgment phase, and is vulnerable to attacks during either or both of these phases. Intuitively, broadcast communication has two important metrics:

*Correspondence to: Avinash Srinivasan, Department of Mathematics, Computer Science and Statistics, 227 Ben Franklin Hall, Bloomsburg University, PA 17815, USA.

†E-mail: asriniva@fau.edu

security and reliability. Unlike reliability, which has drawn much attention from researchers, very few researchers have addressed the security metric [1]. In a one-to-all communication paradigm like broadcasting, where every broadcast originates from a single source, the only way for the source to ensure that all members have received the message is to have them acknowledge the message.

From the system perspective, having all of the nodes in a network acknowledge every single broadcast message is not a scalable solution. Particularly, in large scale networks like wireless sensor networks (WSNs), this increases the network load, causing a higher amount of collision and contention [2,3]. Apparently, it becomes imperative to fine tune the detection threshold to accommodate appropriate levels of natural loss in the network. Otherwise, the system will be coaxed to raise false alarms. False alarms result in unnecessary system downtime as well as wasted resources by executing countermeasures on a network free of attacks. This is particulary detrimental in WSNs since the resources in WSNs are very scarce. The aforementioned problem has been addressed in Reference [1] by using a technique called Secure Implicit Sampling (SIS) in which only a subset of nodes are randomly sampled to acknowledge each message broadcast. This alleviates the burden on the network to a large extent. We will discuss this method in further detail in Section 2. Traditionally, tree-based broadcasting has been considered to be highly unreliable over other techniques because a single malicious[‡] node in the tree can block the message to the entire subtree rooted at it. The extent of damage caused in this scenario is a function of the size of the subtree rooted at the malicious node. Therefore, the attacker can cripple a substantial portion of the network by compromising a single node with a large subtree. For this reason, fault tolerance has been a major concern in flooding tree-based broadcasting. This idea has been captured in Figure 2(a), where the sensor network is represented as a graph and BS stands for BS. In the remainder of this paper, we will refer to static tree-based broadcasting as the Flooding Tree Model (FTM).

The aforementioned drawback of the FTM can be overcome by using blind flooding, in which each node rebroadcasts the message upon receiving it for the first time. This ensures very high reliability; that is, even in the presence of malicious nodes, every node receives the message unless there is a partitioning of the network. However, the fault tolerance of blind flooding comes at the cost of redundant transmissions that may cause a serious problem, referred to as the broadcast storm problem [4]. Retransmission increases communication congestion and contention in the network and wastes critical resources in WSNs. Hence, in light of the above discussions, an inevitable tradeoff between reliability and redundancy always exists.

However, we believe that using an FTM for broadcasting also has some hidden advantages from the system's perspective. The advantage is that when a large number of nodes are deprived of the broadcast message simultaneously, the attacker can be detected immediately. This is because the acknowledgment ratio computed by the BS, which is the ratio of the number of acknowledgments received to the number of acknowledgments expected, will be substantially small. The more nodes the adversary attempts to deny the broadcast message to, the greater the chances are that he will be detected. Even if the system attributes some fixed percentage of lost acknowledgments, say L per cent, to collision and contention in the network, it can still detect an attack with very high probability.

We draw our motivation from the above situation and propose a distributed k-parent Flooding Tree Model (k-FTM), which is robust against Denial-of-Broadcast-Message attacks (DoBM)[§] in WSNs. Our main motivation in proposing the k-FTM is to retain the high detection rate of FTM but at the same time achieve a reliability close to blind flooding with a reduced number of rebroadcasts. When $k = 1$, k-FTM represents the basic FTM. FTM, k-FTM for $k = 2$, and blind flooding have been depicted in Figure 1(a)–(c), respectively. In Figure 2(g), which is a 2-FTM, the solid edges represent the first parent and the dashed edges represent the second parent of a node. The dotted lines indicate the acknowledgment path.

In k-FTM, blind flooding is carried out once at the beginning, after node deployment, to construct the k-parent tree. Once constructed, all the subsequent message broadcasts and acknowledgments flow along the k-FTM. k-FTM is an excellent fault-tolerant model, and to our best knowledge, is the first fault-tolerant tree model to be applied for securing broadcast communication in WSNs. We will confirm

[‡] The terms malicious and compromised are used interchangeably throughout this paper.

[§] A class of attack in which the adversary's primary motive is to deny the broadcast message to as many nodes as possible. In Reference [1], it is referred to as Denial-of-Message (DoM) attack.

DOI: 10.1002/sec

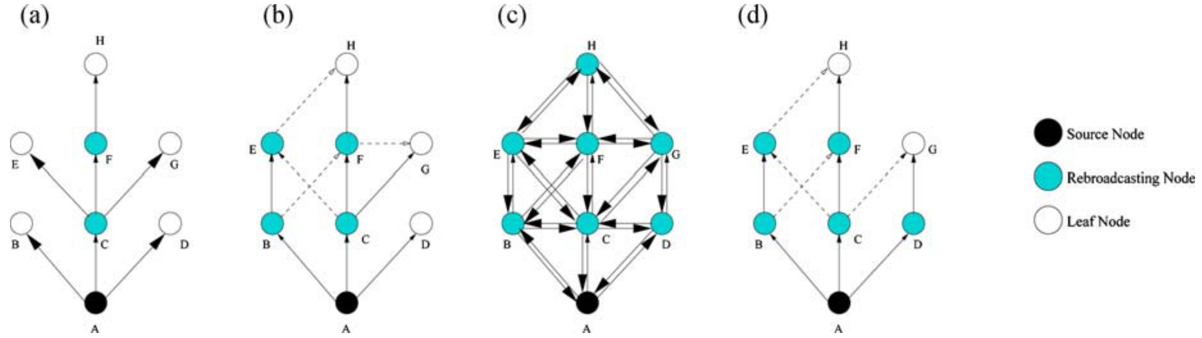


Fig. 1. (a) FTM for a network of eight nodes. (b) *k*-FTM for a network of eight nodes where $k = 2$. (c) Blind flooding in a network of eight nodes. (d) A 2-FTM where every node has two parents with disjoint path to base station.

via simulations that *k*-FTM strikes an optimal balance between the FTM and blind flooding. Our contributions in this paper can be summarized as follows:

1. We propose *k*-FTM, a novel distributed *k*-parent flooding tree model, that has a detection rate close to that of static tree-based broadcasting. *k*-FTM is the first tree model that achieves broadcast reliability close to that of blind flooding with reduced redundant rebroadcasts.
2. *k*-FTM is the first model to employ a reputation and trust-based framework for securing broadcast communication in WSNs. *k*-FTM is also the first fault-tolerant tree model for securing broadcast communication in WSNs.
3. We extend the model to include 'Proactive acknowledgment' in order to enhance the model's detection rate.
4. We present algorithms for constructing the *k*-FTM. We also analyze our model and evaluate it through simulations.

The remainder of this paper is organized as follows. In Section 2, we review relevant literatures. In Section 3, we discuss the proposed model in detail including a brief comparison with blind flooding, and discussions on the attacker model and underlying assumption. In Section 4, we discuss the two acknowledgment techniques, reactive and proactive, in detail. In Section 5, we present four different techniques that a node may use to select its *k*-parents, along with formal algorithms. In Section 6, we provide a detailed discussion on our model and shed light on its strengths and novelty. In Section 7, we describe our simulation environment and discuss the simulation results in detail. Finally, in Section 8 we conclude the paper with directions for future research.

2. Related Work

Though researchers have addressed the problem of energy-efficient and reliable broadcasting in wireless and sensor networks [4–8], security issues have not been addressed adequately. In Reference [4], Ni *et al*. have discussed several drawbacks of the classical flooding algorithm, including energy consumption and reliability. They have also proposed several schemes to reduce redundant rebroadcasts. In References [6,9], Lim and Kim show that finding an optimal flooding tree in an *ad hoc* wireless network is similar to the Minimum Connected Dominating Set (MCDS) problem and show the NP-completeness of the same.

In Reference [10], the internal node-based broadcasting algorithm is presented where it is assumed that each node has knowledge of the geographical coordinates as well as the degree of all its neighbors. With this knowledge, it decides if a node is internal or not, and only internal nodes relay the broadcast message. Perrig *et al*. [11] present two building block security protocols optimized for use in sensor networks, SNEP and μTESLA. SNEP provides confidentiality, authentication, and freshness between nodes and the sink, while μTESLA provides authenticated broadcasts for severely resource-constrained environments.

McCune *et al*. [1] have proposed SIS for the detection of DoM attacks on sensor network broadcasts. In SIS, using appropriate cryptographic functions and pseudo-random keys, the BS encrypts the message, in which it is encoded which nodes are required to acknowledge, and broadcasts it. The adversary has no way of knowing *a priori* the subset of nodes that will be sampled during each round. On receiving the message, each node authenticates the message and, if required, sends back an acknowledgment. In each round, on receiving the

DOI: 10.1002/sec

acknowledgments, the BS authenticates them and computes the acknowledgment ratio, which is the ratio of the received acknowledgments R_i, to the number of expected acknowledgments S_i. If this ratio is below a certain threshold h, that is, $\frac{R_i}{S_i} < h$, then the system raises an alarm.

Our k-FTM uses the aforementioned SIS technique as the underlying message encoding scheme to embed in the broadcast message as to which nodes are expected to acknowledge. k-FTM achieves better performance by integrating SIS with a reputation monitoring system. Note that, in Reference [1], the probability of the attacker remaining undetected varies with the location of the attacked node. It generally increases with the distance between the attacked node(s) and the BS. Therefore, when the acknowledgment from a geographically distant node is lost, there is a very high probability that the BS will attribute it to natural loss. Additionally, when fewer nodes are attacked, the probability of sampled nodes being blocked decreases and so does the probability of the attacker's detection. We thus advocate the use of k-FTM, which is more sensitive to attacks due to the topology in such scenarios. In our model, the probability of detection is not as sensitive as SIS to the physical distance of the node from the BS. Therefore, our model achieves a higher probability of detection compared to the model proposed in Reference [1].

For the sake of completeness, we shall discuss some of the state-of-the-art reputation monitoring systems. Numerous RTMSs such as CORE [12], RFSN [13] and DRBTS [14] have been developed to stimulate node cooperation in MANETs and WSNs. In most of these models, nodes build their own view based on personal observations as well as the recommendations from neighbors. Michiardi and Molva proposed CORE [12], 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. Here, nodes exchange only positive reputation information. The authors argue that this prevents bad-mouthing attacks. However, they 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 continual basis to remain trusted. Otherwise, their reputation will deteriorate until they are excluded.

Buchegger and Boudec [15] have presented CONFIDANT with predetermined trust, and later improved it with the Bayesian trust system and a passive acknowledge mechanism (PACK), respectively. This model makes misbehavior unattractive in MANETs based on selective altruism and utilitarianism. CONFIDANT is a distributed, symmetric reputation model which uses both first-hand and second-hand information for updating reputation values. Mundinger and Boudec [16] have presented a two-dimensional reputation system for protecting the system from liars to ensure cooperation and fairness in mobile *ad hoc* networks.

Ganeriwal and Srivastava [13] 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.

3. *k*-FTM

In this section, we will first briefly discuss the differences between our model and blind flooding. Then, we delineate the attacker model and enlist some underlying assumptions of our model. We then discuss how k-FTM works in detail.

3.1. *k*-FTM *versus* Blind Flooding

Though blind flooding is one of the most preferred and reliable methods for achieving network-wide broadcast, it has a few drawbacks that can pose serious problems, particularly if a network uses blind flooding frequently. In this subsection, we compare and contrast k-FTM with blind flooding.

In blind flooding, each node receives n copies of the same message, where n is the number of neighbors. In a dense network, this will result in very high redundancy. In k-FTM each node receives up to a maximum of k copies, one from each of its k-parents, and k can be as small as 2. We do not address the situation where $k = 1$, since this is same as FTM.

In blind flooding, each node rebroadcasts the message upon receiving it for the first time. So, in a network of N nodes, there will be N rebroadcasts. In k-FTM, only internal nodes rebroadcast the message on receiving it for the first time. So, in a network with N nodes of which M are leaf nodes, there will be only $(N - M)$ rebroadcasts. Therefore, in k-FTM, the number of rebroadcasts is reduced to the number of internal nodes. This results in substantial savings of communication bandwidth as well as energy for

DOI: 10.1002/sec

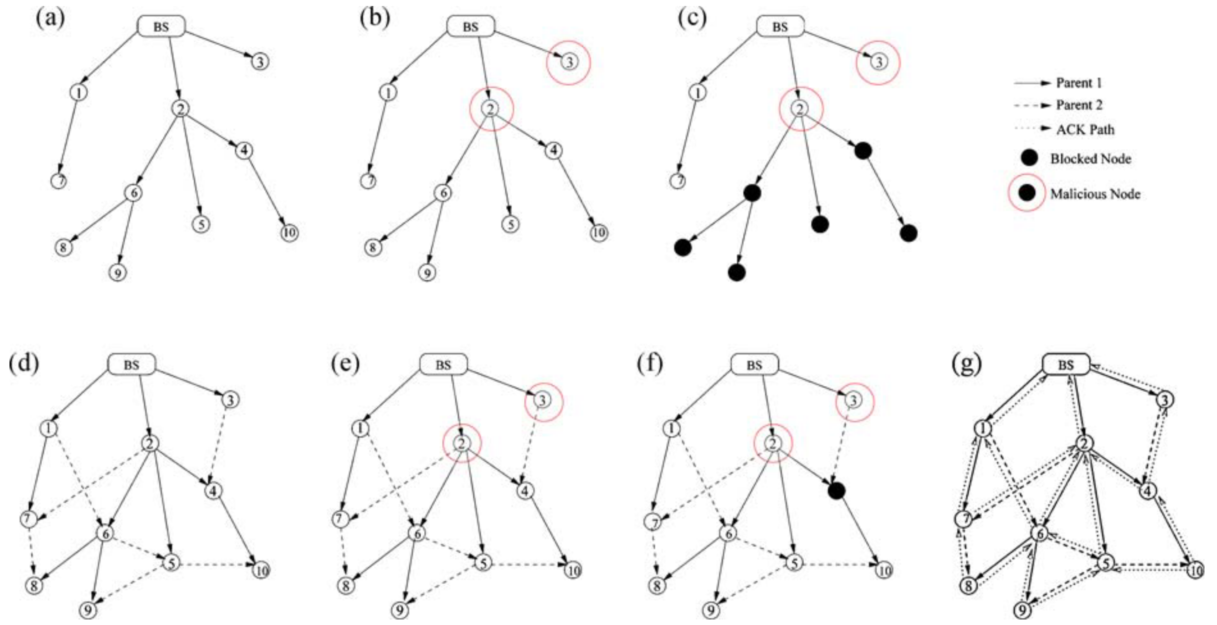


Fig. 2. (a)–(c) Impact of compromised nodes on system throughput in FTM; (d)–(f) impact of compromised nodes on system throughput in k-FTM ($k = 2$); (g) ACK path in a k-FTM ($k = 2$).

non-broadcasting nodes. For example, in Figure 2(d), the network has 10 nodes, of which 7 are broadcasting and 3 are non-broadcasting with $k = 2$. This reduces the number of rebroadcasts by 30 per cent compared to blind flooding. Note that k is a tunable parameter. Therefore, the redundancy can be varied depending on the application domain as well as the associated security and reliability requirements.

3.2. Attacker Model

The attacker's sole motive is to block each broadcast message to as many nodes in the network as possible. The attacker can potentially attack the broadcast communication either during the broadcast phase or the acknowledgment phase. However, the impact of the attack on the network and the adversary differs dramatically for these two phases. By attacking during the broadcast phase, the attacker is blocking the message from reaching the nodes, which is a serious threat since the BS cannot communicate the message to all the nodes. The number of nodes deprived of the message depends on the degree of the attacked node in the k-parent tree. The degree of a node i in the k-parent tree is a measure of the number of children the node has, which is denoted as $|C_{\text{List}}|$.

On the otherhand, by attacking during the acknowledgment phase, the attacker is merely increasing the false alarm rate since the message would have already been delivered to the nodes. This is not as serious as the previous situation and could be referred to as 'False negative.' This does not benefit the attacker since it only increases the probability of his detection. Hence, the attacker only attacks during the broadcast phase. There may be other attacker models in which the adversary may attack during the acknowledgment phase and cause the system to panic or to rebroadcast the message multiple times, consuming precious network resources, but we do not address such attacks in this paper.

In addition, we assume that in k-FTM the adversary will not attack during the initial tree setup phase. This results from a straightforward logic as follows. By attacking during the initial tree setup phase and blocking the message to as many nodes as possible, the adversary will end up being a non-broadcasting node. Consequently, during the subsequent actual message broadcast rounds, the adversary will not be able to induce any damage. This is counterproductive for the adversary and in contradiction with his motive.

We assume that the attacker is external to the network and randomly chooses nodes and compromises them. He cannot compromise the BS, and can never restore a compromised node to its original configuration. Compromised nodes attack by not forwarding the message to their children and selective forwarding is not permitted in k-FTM. The attacker is not aware of the network topology which makes it more difficult for him to choose nodes for launching his attack. We consider only non-forwarding attacks in a dense static network. When sampled, a compromised node promptly acknowledges and never drops the acknowledgments of other nodes. We assume that there is a secure key-management protocol to establish pairwise keys between each node and the BS. Each node encrypts its acknowledgment using the pairwise key it shares with the BS. Hence, compromised nodes cannot inject fabricated acknowledgments. We also assume that compromised nodes do not collaborate. Nodes have uniform transmission ranges and the rate of message propagation is uniform. Finally, we assume that there is an appropriate back-off timer to resolve contention/congestion-related issues.

3.3. *k*-FTM–Model Description

In *k*-FTM, after node deployment, the BS carries out an initial round of flooding with a *Hello* message after network initialization as presented in Algorithm 1. Upon receiving the *Hello* message for the first time, every node rebroadcasts it. After rebroadcasting, nodes wait until Time$^{\text{ACK}}_{\text{out}}$ occurs. If a node i receives acknowledgment(s) from neighboring nodes before the timeout, then it stores the *ID*s of the acknowledging nodes in C^i_{List}. C_{List} of a node keeps track of its children. The node also sets its state to *broadcasting* by setting Flag$^i_{\text{state}}$ = true. Otherwise, it sets its state as *non-broadcasting* by setting Flag$^i_{\text{state}}$ = false. A node with Flag$_{\text{state}}$ = false does not rebroadcast during subsequent broadcast rounds. Each node, excluding those that are 1-hop from the BS, acknowledges as child to exactly k nodes from which it receives the *Hello* message. Each node i maintains a second list P^i_{List} to keep track of its parents, that is, the nodes to whom it has acknowledged as child and stores their *ID*s. Note that a node can choose its *k*-parents using several different methods, which we will discuss in detail in Section 5. We have relaxed the *k*-parent constraint on nodes that are within the communication range of the BS since they are assured of message delivery. Algorithm 2 formally presents the process of updating the P_{List} and the C_{List}.

Algorithm 1

```
Initialize
for each node $i$ in the network do
    if $i$ in range of base station then
        $P^i_{\text{List}} \leftarrow P^i_{\text{List}} \bigcup base\_station$;
        $\text{Flag}^i_{\text{State}} \leftarrow$ broadcasting;
    else
        $P^i_{\text{List}} \leftarrow \emptyset$;
        $\text{Flag}^i_{\text{State}} \leftarrow$ non-broadcasting;
    end if
    $C^i_{\text{List}} \leftarrow \emptyset$;
end for
```

Algorithm 2

```
Update_$C_{\text{List}}$ $(j, i)$
$C^j_{\text{List}} \leftarrow C^j_{\text{List}} \bigcup i$;
if $\text{Flag}^j_{\text{State}}$ is non-broadcasting and $C^j_{\text{List}} \neq \emptyset$ then
    $\text{Flag}^j_{\text{State}} \leftarrow$ broadcasting;
end if

Update_$P_{\text{List}}$ $(i, j)$
$P^i_{\text{List}} \leftarrow P^i_{\text{List}} \bigcup j$;
```

Once a node establishes all *k*-parents and Time$^{\text{ACK}}_{\text{out}}$ occurs, it sends a copy of its P_{List} and C_{List} to the BS. At the end of the tree construction phase, the BS will have a copy of P_{List} and C_{List} of every node in the network. The BS then carries out a simple sanity check to ensure nodes are not faking their lists. The sanity check is performed as follows: if node A claims to be the parent of node B, then node B should have claimed itself as the child of node A. With a single such inconsistency, it may be difficult for the BS to determine which of the two nodes is lying. However, since we are assuming that there is no collaboration among malicious nodes, two such inconsistencies for a single node will flag it as malicious. If a node has faked its list, it is immediately blacklisted. This completes the construction of the *k*-FTM. During subsequent broadcast rounds, nodes accept messages only from nodes in their P_{List} and C_{List}.

After the initial *k*-FTM construction, the reputation and trust-based system comes into action. In our model, each node is equipped with a watchdog that monitors its neighborhood. Neighborhood of a node i is the set of all nodes that are in i's communication range. In our model we shall restrict the neighborhood of a node i to its parents (P^i_{List}) and children (C^i_{List}), which we will discuss later. Consequently, using the feedback from watchdog, each node assigns a reputation value to nodes in its P_{List} and C_{List}. The reputation value that a node i has assigned to its neighbor node j at time t is represented as $R^t_{i,j}$. We will use this reputation monitoring system as the underlying framework in our model. When a node is sampled, it includes the reputation value of nodes in its neighborhood in the acknowledgment. With the watchdog-driven reputation monitoring system, a node can report on the forwarding behavior of its children as well.

Using k-FTM has a twofold benefit from the system's perspective. First, the attacker's malicious intentions are thwarted, since the message is delivered to the otherwise blocked nodes in FTM by at least one of the k-parents, assuming each node has at least one benign parent, that is, a node can have at most $(k-1)$ malicious parents. Suppose a node has n nodes in its neighborhood, then it can have no more than $n-1$ malicious nodes in its neighborhood to ensure that it has at least one benign parent. Second, the attacker is very likely to be detected immediately since a compromised node's child nodes, if sampled, report to the BS on failing to receive k copies of the message. Note that a malicious node's parents can also report on its non-forwarding behavior monitored by the watchdog.

When the BS has to broadcast a message, it encodes in the message which nodes are expected to acknowledge using the SIS technique. The subset of nodes selected to acknowledge each broadcast message is probabilistically chosen, which makes it completely unpredictable and extremely hard for the adversary to determine which nodes are being sampled for any given round. On receiving the message, nodes decrypt the message to determine if they are expected to acknowledge. If so, then they send back an acknowledgment. Two methods exist for a node i to choose one or more nodes among its k-parents to send the acknowledgment to. In the first method, i randomly chooses one or more parents to send the acknowledgment to. The number of parents it chooses to send the acknowledgment to is a function of permissible redundancy. In the second method, i uses the accumulated reputation value and chooses one or more parent nodes to send the acknowledgment, in decreasing order of reputation value. Note that, for any given message, if node i does not receive a copy of the message from its parent j, then i automatically excludes j when an acknowledgment has to be sent. The above strategy of choosing parents to send acknowledgments to also applies to intermediate nodes that are not sampled to acknowledge but have to forward the acknowledgment of a descendant node.

The acknowledging process itself can be divided into two groups: *reactive* and *proactive* acknowledgments. We shall first discuss the process of acknowledgment in general before comparing and contrasting the two. The acknowledgment of a node is a k-field list with one field assigned to each parent which has two pieces of information. The first piece of information indicates whether the corresponding parent forwarded the message during the current round and the second piece of information is the accumulated reputation value of the corresponding parent node. A binary 1 is used to indicate if the corresponding parent forwarded the message and 0 is used otherwise. Similar lists can be used by a node to report on the forwarding/non-forwarding behavior of its children. To control redundancy, the BS can additionally indicate in the message whether the sampled node has to report on its parents only or on both parents and children.

For the sake of completeness, we shall now briefly discuss the reputation monitoring system, reputation update, and distribution. Every node maintains a reputation value for each of its parent and child nodes using the reputation monitoring system. The reputation value is a continuous value and is range bound between 0 and 1. The reputation value of a parent node is computed as follows, with j as the parent node and i as the monitoring child.

$$R_{j,i}^{\text{new}} = \mu_1 \times R_{j,i}^{\text{cur}} + (1-\mu_1) \times \tau \qquad (1)$$

Similarly, the reputation of a child node is computed as follows, with l as the child and i as the monitoring parent:

$$R_{l,i}^{\text{new}} = \mu_2 \times R_{l,i}^{\text{cur}} + (1-\mu_2) \times \tau \qquad (2)$$

In Equation (1), $\tau = 1$ if the corresponding parent node forwarded the message, else $\tau = 0$. Similarly in Equation (2), $\tau = 1$ if the corresponding child node forwarded the message further, otherwise $\tau = 0$. The constants μ_1 and μ_2 are system-dependant parameters, each range bound between 0 and 1. They decide the extent to which past history can be discounted and substituted with the most recent behavior. For illustration, consider five nodes i, j, k, l, and m. Let $j, k \in P_{\text{List}}^{i}$ and $l, m \in C_{\text{List}}^{i}$. Let us assume that i received a copy of the message for j but not k. Then the reputation values of j and k are updated using Equation (1) as follows: $R_{j,i}^{\text{new}} = \mu_1 \times R_{j,i}^{\text{cur}} + (1-\mu_1)$ and $R_{k,i}^{\text{new}} = \mu_1 \times R_{k,i}^{\text{cur}}$. Similarly, if i noticed that l forwarded the message further and m did not, then their reputation values are updated using Equation (2) as follows: $R_{l,i}^{\text{new}} = \mu_2 \times R_{l,i}^{\text{cur}} + (1-\mu_2)$, and $R_{m,i}^{\text{new}} = \mu_2 \times R_{m,i}^{\text{cur}}$.

4. Acknowledgment Techniques

In this section, we will discuss the two acknowledgment techniques, *reactive* and *proactive*, presented in

DOI: 10.1002/sec

Algorithm 3

Reactive Acknowledgment

```
for each node $i$ in the network do
    if $i$ is sampled then
        Acknowledge;
    end if
end for
```

Proactive Acknowledgment

```
for each node $i$ in the network do
    if Message Copies $\leq \mathrm{TH}_{\mathrm{rep}}^{\mathrm{ack}_{\mathrm{pro}}}$ then
        Report to base station;
    end if
end for
```

Algorithm 3, in detail, delineating their merits and differences. The acknowledgment of a node indicates how many copies of the message it received and which parents failed to forward. It can also indicate how many of the child nodes further forwarded the message successfully.

4.1. Reactive Acknowledgment

In the reactive acknowledgment scheme, only nodes that are sampled by the system send back an acknowledgment. The set of nodes sampled to acknowledge each broadcast message is unique and determined probabilistically using the SIS technique proposed in Reference [1]. The reactive acknowledgment scheme is very effective in curtailing network traffic since only a subset of nodes acknowledge. Note that in the reactive acknowledgment scheme, the same node may not be sampled to acknowledge for a long time. Hence, it records both the malicious and benign behavior of nodes in the reputation metric, and reports it to the BS when sampled.

When the BS fails to receive acknowledgment from node i that is sampled for a particular round, it first determines node i's parents using P_{List}^{i}. Then, it checks the received acknowledgments to see if any of the sampled nodes have common parent(s) with i, against whom they have reported, that is, reported against node(s) in P_{List}^{i}. If there are no reports, then it is very likely that the node has either failed or the acknowledgment has been lost enroute due to contention/collision. Note that nodes cannot inject spoofed ACKs in reactive acknowledgment because only sampled nodes send ACK and the adversary has no information ahead of time as to which nodes will be sampled for a particular round.

4.2. Proactive Acknowledgment

In the proactive acknowledgment scheme, every node that receives fewer than k copies of the message, one from each of its k-parents, sends back an acknowledgment to the BS informing it about the incident.

The proactive acknowledgment scheme overcomes the drawbacks of the reactive acknowledgment scheme effectively in two scenarios: (1) in case a sampled node is blocked by all its parents and no other sample node happens to share a common parent with the attacked node, the BS is going to ignore the attack (this situation will not arise in our simulations since we are assuming that each node has at least one benign parent), and (2) in case a non-sampled node is attacked, the BS will never be informed about it and therefore the attack goes unnoticed. Consequently, the attacker continues to survive in the network with the potential to damage the system further.

Proactive acknowledgment provides the BS with ample evidence, helping it in detecting the attacks almost every single time with greater confidence and certainty. It also enables the BS to ball park the malicious node more accurately. However, the traffic overhead and collision incurred in proactive acknowledgment approaches that of blind flooding. Nonetheless, by adjusting the threshold on the number of copies needed to report *via* proactive acknowledgment, $TH_{\mathrm{rep}}^{\mathrm{ack}_{\mathrm{pro}}}$, the traffic overhead and collision can be controlled. For illustration, consider a k-FTM with $k = 5$. In this tree, if $TH_{\mathrm{rep}}^{\mathrm{ack}_{\mathrm{pro}}} = 3$, then a node will send a report in the proactive acknowledgment scheme only if the number of message copies received is less than 3. Note that in proactive acknowledgment, the adversary can spoof acknowledgments only from captured nodes over which it has full control and thus can access all the information stored in that node. It cannot spoof the ACK of a benign node for two reasons: (1) it does not have access to the key shared by a benign node with the BS and (2) it does not know if the benign node has received less than $TH_{\mathrm{rep}}^{\mathrm{ack}_{\mathrm{pro}}}$ copies of the message.

5. Parent Selection Methods

Below are three different methods for choosing the k-parents during k-FTM construction such that the system throughput is maximized while keeping the detection rate as high as possible. We present formal algorithms

Algorithm 4

Fastest First *k*-Parents

```
Initialize;
for each node $i$ not in range of base station do
    $parent\_count_i \leftarrow 0$;
    while $parent\_count_i < k$ do
        for ($message$) received from each neighbor $j$ do
            if $message$ is fresh then
                rebroadcast ($message$);
            end if
            $Update\_P_{\mathrm{List}}(i, j)$;
            $parent\_count_i + +$;
            $Update\_C_{\mathrm{List}}(j, i)$;
        end for
    end while
end for
```

for these methods and discuss them in the remainder of this section. The first method is called the *Fastest First k-parents*. In this method, a node acknowledges as child to the first k nodes from which it receives the *Hello* message. This method has been formally presented in Algorithm 4. This is the simplest of the three methods proposed and has the least overhead.

The second method is called the *Disjoint Path k-parents*. In this method, a node receives a *Hello* message from all its neighbors along with *path_label* until $\mathrm{Time}_{\mathrm{out}}^{\mathrm{msg}}$ occurs. *path_label* is a list of *ID*s of nodes through which the message has passed starting from the BS. Nodes that broadcast the message after $\mathrm{Time}_{\mathrm{out}}^{\mathrm{msg}}$ are ignored during the parent selection process. A node, while choosing its k-parents, chooses nodes with disjoint *path_label*. The advantage of this method is that it augments system throughput significantly. However, a node may encounter a situation wherein it may not find all k-parents with a disjoint *path_label*. Under this circumstance, assuming that a node finds only m out of k parents with a disjoint path, it chooses the remaining $(k - m)$ parents according to a first-come-first-served principle. Algorithm 5 formally presents this method. After choosing the k-parents, a node appends its ID to k unique *path_label* received from the selected k-parents and then rebroadcasts it along with the message. Figure 1(d) is an example of a 2-FTM where each node has two parents with disjoint paths to the BS. This method has a larger overhead in terms of both storage and communication. To reduce this overhead, we can have a node rebroadcast the message by appending its ID to only one unique *path_label* from one of its parent's. This improved method is called the *Improved Disjoint Path k-parents*. In this paper, due to space limitations, we shall not discuss it further but we will look into it in our future work.

Algorithm 5

Disjoint Path *k*-Parents

```
Initialize;
for each node $i$ not in $TR_{BS}$ do
    while $\mathrm{Time}_{\mathrm{out}}^{\mathrm{msg}}$ has not occured do
        for ($message$, $path\_label$) received from each neighbor $j$ do
            if $message$ is fresh then
                Put ($message$, $path\_label$) in buffer;
            else
                Put $path\_labels$ in buffer;
            end if
        end for
    end while
    Choose max available parents with disjoint $path\_label$;
    Choose remaining parents on FCFS basis;
    add all $k$-parents to $P_{\mathrm{List}}^{i}$;
    for each parent $j$ selected do
        $Update\_C_{\mathrm{List}}(j, i)$;
        $\mathrm{Flag}_{\mathrm{State}}^{j} \leftarrow broadcasting$;
    end for
    Append ID to $k$ unique $path\_labels$ received from the $k$-parents and rebroadcast along with $message$;
end for
```

The third method is called the *Unique Level-1 Ancestors*. This method is quite similar to the *Disjoint Path k-parents* method with minor changes to mitigate the communication overhead. In *Unique Level-1 Ancestors*, only level-1 nodes append their ID to the message and rebroadcast it. When choosing the k-parents, a node chooses its parents with unique level-1 ancestors. If a node cannot find all k-parents with unique level-1 ancestors, then it reverts to the first-come-first-served principle as explained before. Every node rebroadcasts the message along with the ID of the level-1 ancestor.

6. Discussion

The adversary's strategy changes with his motive and the degree of his conservativeness. He can be either extremely aggressive, inducing maximum damage over a short period of time, and get caught, or, he can be

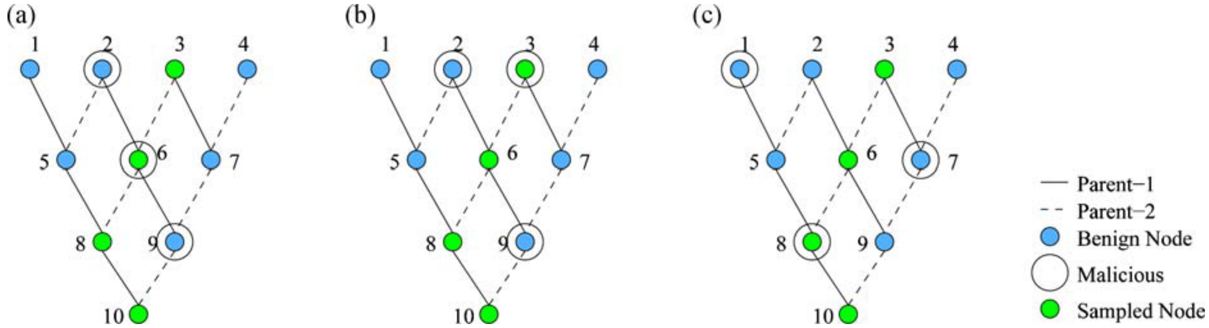


Fig. 3. Impact of malicious nodes in k-FTM ($k = 2$) and their detection.

very prudent, inducing minimal damage during each broadcast round, and remain undetected for extended periods of time. An adversary can choose to vary his strategy by varying his aggressiveness between the above two extremes. At all times, the adversary solely strives to maximize his reward, measured as the extent of damage caused to the system, by balancing the tradeoff between his aggressiveness and his expected survival time. The system, on the other hand, always aims at maximizing its performance with high detection rate by thwarting the adversary's attempt before he induces irreparable damage to the network. However, there is an inevitable tradeoff between good performance and a high detection rate.

Good performance often comes at the cost of high communication overhead, high memory usage, and low detection rate. By opting to maximize the reliability, the system allows malicious nodes to persist in the network. This is because when reliability is the requirement, blind flooding suits best since the chances of every node receiving the message, even in the presence of malicious nodes, is very high due to redundant rebroadcasts. On the other hand, when a high detection rate is the requirement, FTM suits best. By using FTM to achieve a high detection rate, the system sacrifices its performance, since a single malicious node with a large subtree can cripple a substantial portion of the network. However, the proposed k-FTM, $\forall k > 1$, strikes an optimal balance between blind flooding and FTM, by retaining a reliability close to blind flooding and a detection rate close to FTM.

In a network with n nodes, each node has a uniform probability of $\frac{1}{n}$ for being sampled. If sampled nodes are blocked by the compromised nodes, then they fail to receive the broadcast message. Consequently, the BS will fail to receive their ACK. In an ideal-world situation where there is no natural loss, the BS will attribute this loss to DoBM. But in a real-world scenario, where there is natural loss, the BS may attribute this loss to either natural loss or DoBM depending on various factors like network load, permissible natural loss in the network, etc. As k increases, our model approaches blind flooding and ensures a definite coverage. But at the same time the number of redundant rebroadcasts increase since, with an increase in k, the number of non-broadcasting nodes decreases. We confirm this through simulation and the results are presented in Figure 6(a).

The number of children a node i can have is upper bound by $m - k$, where m is the number of nodes in the neighborhood of i and k the number of parents i has to have. Also, for a node with m neighbors, it is stable up to $(m - 1)$ malicious nodes provided only $(k - 1)$ parents are chosen from the $(m - 1)$ malicious nodes and one parent is the benign node in the neighborhood. For illustration consider Figure 3, which is a 2-FTM. Let nodes 3, 6, 8, and 10 be the sampled nodes (30 per cent sampling) in all three scenarios. Now, following are the different scenarios from the point of view of a malicious node. A malicious node itself can be denied the message from one of its parents which is also a malicious node. In this scenario, as shown in Figure 3(a), though node 6 is malicious, it only gets one copy of the message from node 3. Node 2, which is also malicious, blocks the message to both 5 and 6. Node 6 further blocks the message to nodes 8 and 9. However, nodes 8 and 9 get a copy of the message from nodes 5 and 7, respectively. Finally, node 9 blocks the message to node 10, which gets a copy of the message from node 8. Now, during the acknowledgment phase, the sampled nodes 3, 6, 8, and 10 report the following: 3 has no reports (assuming it received both copies of the message), 6 reports on 2, 8 reports on 6, and 10 reports on 9. In this scenario, even if node 9 were to be sampled and even if it did report on 6, it would be caught with node 10's report since, there was no reason for 9 to block the message to 10 when it got the message from node 7. Assuming 9 did not get a copy of

DOI: 10.1002/sec

the message from either parents, then there was no way that 9 could have guessed the existence of the message and acknowledged it.

There is another possibility here. If nodes 6 and 9 are collaborating, then node 9 can report against node 7 instead of node 6. Now, since node 9 is the only child of node 7, the BS does not have enough evidence to punish node 9. Also, now the number of reports against node 6 will be one short, making it further ambiguous for the BS to take a decision. Hence, the simple majority decision rule (SMDR) is useful in such a situation. If at least half the number of children of a node have reported against it, then the node will be treated as misbehaving and accordingly punished. This again opens up the possibility of bad-mouthing attacks. In a 2-FTM, a node can simply report against one of its parents randomly, and since it satisfies the requirement to apply the SMDR, a benign parent node can get punished. Such attacks can be overcome with the BS querying the parents of a reported node to check if the node forwarded the message. Since selective forwarding is not permitted, if the parent nodes confirm that the node indeed forwarded the message, then the BS can take decisions with higher confidence. Note that in a collaboration free environment, that is, when no malicious nodes collaborate, then the number of reports against a malicious node i in the acknowledgment phase will be equal to $|C^{i}_{\text{List}}|$.

In the scenario presented in Figure 3(b), the message is blocked to node 6 by both its parents. Consequently, during acknowledgment phase, node 3 reports normal, node 6 does not acknowledge, node 8 reports against node 6, and node 10 reports against node 9. Similarly, for the scenario presented in Figure 3(c), during acknowledgment phase, node 3 reports normal, node 6 reports normal, node 8 reports normal, and node 10 reports against node 8.

The BS reconstructs the tree $\text{Time}^{\text{Tree}}_{\text{out}}$ occurs. Prior to the reconstruction, the BS broadcasts a message informing nodes about the malicious node(s) so that they do not choose the malicious node(s) as one of their parents when k-FTM is reconstructed. There is a possibility that the warning message itself may be denied to nodes. However, this is not a major threat since nodes in the current tree have kept track of the forwarding/non-forwarding behavior of their neighborhood using the reputation system. The main purpose of broadcasting this message is to warn those nodes that are in the vicinity of a malicious node but are not its parent or child in the current k-parent tree. These nodes are highly vulnerable to becoming the child node of such malicious nodes in the reconstructed tree by virtue of their locality. By broadcasting a warning message, the malicious node is forced to be either a non-broadcasting node or at best a node with a very small subtree rooted at it.

Rebuilding the tree based on $\text{Time}^{\text{Tree}}_{\text{out}}$ has a twofold advantage. First, it prevents a dormant adversary from getting familiar with the locality, thereby curtailing the damage it can cause in the future. Second, it ensures that nodes are moved around providing all nodes a fair chance of being both a broadcasting and a non-broadcasting node. On the other hand, rebuilding the tree when the malicious behavior exceeds a predetermined threshold renders a compromised node as a non-broadcasting node or at least tapers its node degree such that only a small subtree is rooted at it in the new k-FTM. This curtails the damage the adversary can cause in the subsequent message broadcast rounds.

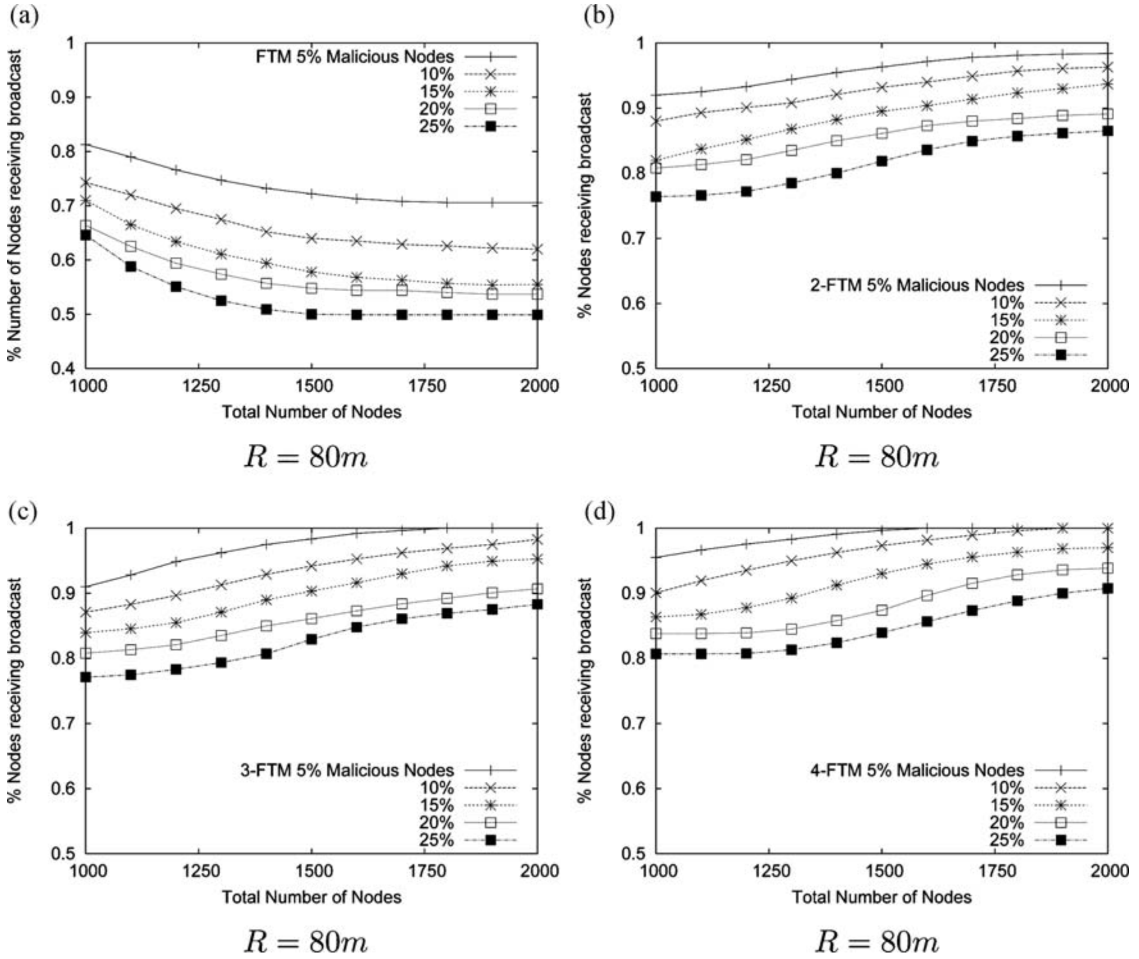
7. Simulation and Results

In this section, we discuss the simulation environment used in our simulations followed by a detailed discussion on the result obtained.

7.1. Environment and Setup

Our simulations have been carried out on a custom JAVA simulator for proof-of-concept evaluation of our protocol without concerning lower-layer details such as packet contention and collision. In all our simulations, the number of nodes N and the transmission range R of nodes are considered as tunable parameters. The number of parents k and the choice between reactive and proactive acknowledgments are also treated as tunable parameters. We have varied N from 1000 to 2000 in increments of 100 and R from 40 to 100 in increments of 20. With the above variations in the network, we could generate 44 different network settings. The results have been averaged over 1000 iterations for statistics stability. For each trial, a $500\,\text{m} \times 500\,\text{m}$ field was randomly seeded with arbitrarily deployed sensors. In our simulations, we consider N homogeneous sensors and model the network as an undirected graph $G = (V, E)$. Here, V is the set of sensor nodes and E the set of links between the sensor nodes. A link exists between two nodes if they lie in each other's communication range R. Each link is treated as a bidirectional link, that is, if node i can communicate with node j *via* link (i, j), then node j can communicate with node i *via* link (j, i).

DOI: 10.1002/sec



$R = 80m$

$R = 80m$

$R = 80m$

$R = 80m$

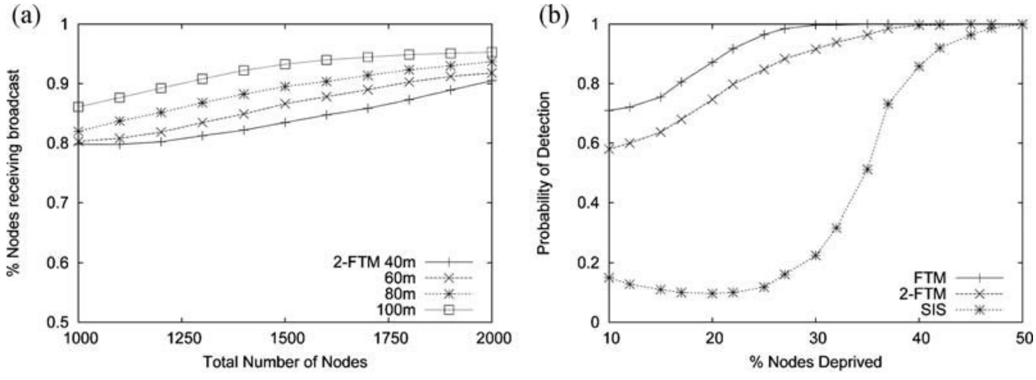
Fig. 4. (a) Impact of different percentage of malicious nodes on throughput in a static tree. (b)–(d) Impact of different percentage of malicious nodes on throughput in a k-FTM for $k = 2$, 3, and 4.

7.2. Results

In this section we shall use S, M, N, R, and ACK to denote the percentage of sampled nodes, the percentage of malicious nodes, the total number of nodes in the network, the node transmission range, and the acknowledgment type.

In Figure 4(a), we have presented the results for total number of nodes actually receiving the broadcast message, that is, throughput, in a FTM with varying percentages of malicious nodes. We can see that the throughput decreases steadily as the percentage of malicious nodes in the network increases. Throughput also decreases with increasing density, since with higher density, more nodes tend to get blocked by a single malicious node. However, throughput is not as sensitive to increasing density as it is to increasing percentages of malicious nodes. Similarly, in Figure 4(b), we have presented the results for throughput in k-FTM where $k = 2$. We shall call this model 2-FTM. It is very clear from the graph that the throughput in 2-FTM increases with N, although it tends to decrease with increasing M. Throughput reaches a maximum of about 97 per cent with $N = 2000$ and $M = 5$ per cent, and reaches a minimum of about 77 per cent with $N = 1000$ and $M = 25$ per cent. On the other hand, in FTM (Figure 4(a)), throughput reaches a maximum of 81 per cent with $N = 1000$ and $M = 5$ per cent, and a minimum of 50 per cent with $N = 2000$ and $M = 25$ per cent. Therefore, 2-FTM successfully achieves a significantly higher throughput, on average about 25 per cent when compared to FTM. This throughput is nearly as good as in blind flooding but with a reduced number of rebroadcasts. We have observed results for k-FTM with $k = 3$ and $k = 4$, and the results are presented in Figure 4(c) and (d), respectively. All results presented are for $R = 80$ m.

DOI: 10.1002/sec



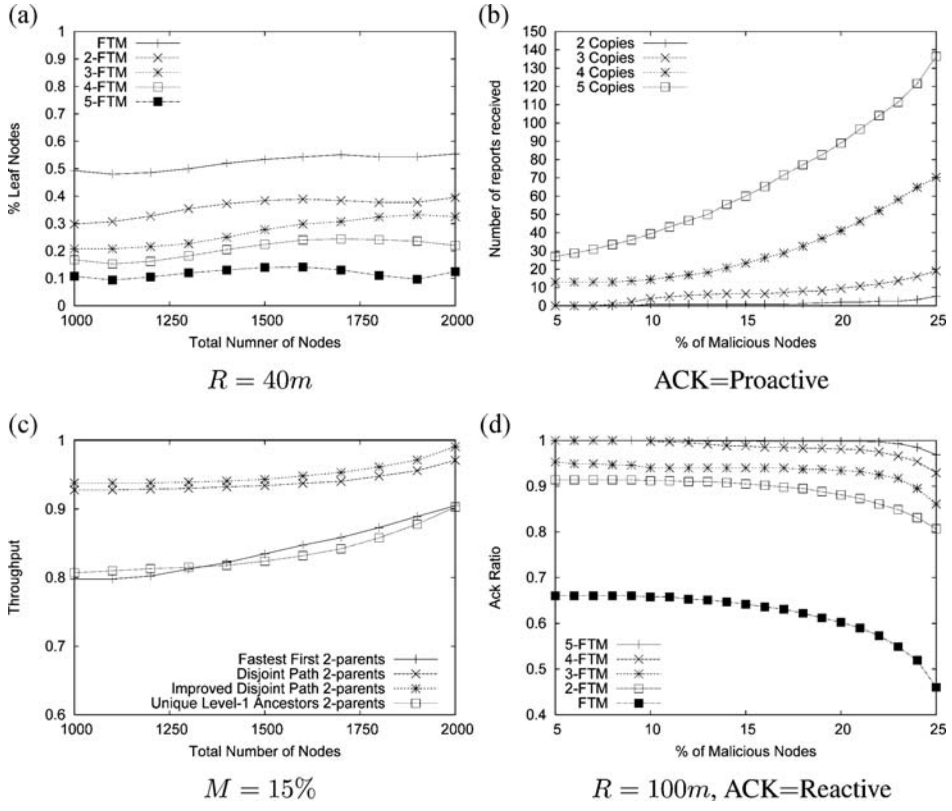
$M = 15\%$

$R = 40m, M = 15\%$

Fig. 5. (a) Impact of different transmission range on coverage. (b) Impact of different percentage of malicious nodes on probability of detection.

In Figure 5(a), we have presented results comparing the system throughput in a k-FTM, with $k = 2$, by varying R from 40 to 100 m in steps of 20 m. The system performs best with a 100 m range. However, at 100 m the energy consumption will also be higher. We can see that it achieves good throughput even at lower values of R. In this graph we can see that k-FTM achieves an average throughput of about 87 per cent for $k = 2$.

In Figure 5(b), we have plotted the results comparing probability of detection of FTM, k-FTM, and SIS [1]. Probability of detection has been plotted against percentage of deprived nodes. From the graph, it is



$R = 40m$

ACK=Proactive

$M = 15\%$

$R = 100m$, ACK=Reactive

Fig. 6. (a) Percentage of leaf nodes in k-FTM for different values of k. (b) Number of reports received by BS for different threshold values. (c) Comparison of three different methods of k-FTM construction. (d) Comparison of acknowledgment ratio at BS for different values of k.

DOI: 10.1002/sec

very clear that both FTM and *k*-FTM outperform SIS. Though the performance of these models converge, the difference in their performance with fewer percentage of nodes deprived is significant. The results are in confirmation with our discussion at the end of Section 2.

In Figure 6(a), we have plotted the percentage of non-broadcasting nodes against total number of nodes, for our *k*-FTM varying *k* from 1 to 5 in increments of 1. We can see that as *k* increases, the percentage of non-broadcasting nodes decreases. This clearly indicates that with increasing *k*, *k*-FTM tends to approach the performance of blind flooding in terms of traffic overhead and redundancy. Note that, for a given *k*, the percentage of non-broadcasting nodes marginally increases with increasing density.

In Figure 6(b), we have compared the number of reports received at the BS under different scenarios using proactive acknowledgment. We have considered a 5-FTM of 1000 nodes with 60 m transmission range and 10 per cent malicious nodes. We have studied the extent to which redundancy can be controlled by varying $TH_{\text{rep}}^{\text{ack}_{\text{pro}}}$ from 2 to 5. When sampled nodes are asked to report on receiving fewer than two copies of the broadcast message, the BS receives a maximum of five reports. This number increases with the number of copies expected. It also increases with increasing density. We can see that when sampled nodes are asked to report if they fail to receive fewer than five copies, about 135 reports are received.

In Figure 6(c), we have compared the performance of different methods for constructing the *k*-FTM with $k = 3$, with 80 m transmission range and 10 per cent malicious nodes. We see that *Disjoint Path k-parents* and *Improved Disjoint Path k-parents* have almost the same performance and achieve the best coverage. In Figure 6(d), we have plotted the acknowledgment ratio for FTM and *k*-FTM with *k* varied from 1 to 5 in steps of 1. We see that FTM has the lowest acknowledgment ratio followed by 2-FTM, 3-FTM, 4-FTM, and 5-FTM. The acknowledgment ratio tends to increase with increase in *k* but decreases with increasing *M*.

8. Conclusion

We have proposed *k*-FTM, a novel distributed *k*-parent Flooding Tree Model that efficiently addresses both reliability and security metrics of broadcasting in WSNs. It can be easily adapted to other networks. *k*-FTM is very robust and efficient in detecting DoBM and to our best knowledge, it is the first fault tolerant tree model. *k*-FTM is also the first model to employ a reputation and trust-based framework for secure and reliable broadcasting in WSNs. Our model has reliability close to blind flooding and a detection rate close to a static tree. We have presented various methods, with algorithms, for constructing *k*-FTM. We have also proposed two different acknowledgment techniques with different redundancy and detection rates. Simulation studies have been conducted to evaluate the applicability of the model and results have been promising. In our future work, we would like to investigate the following: (1) use of directional antennas instead of omni-directional antennas to mitigate energy consumption, (2) correlation between network topology and various attacks on WSN communication, (3) relax the constraint permitting collaboration among malicious nodes and study its impact on network throughput, and (4) conduct a in-depth simulation of the protocol on NS-2.

Acknowledgment

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

References

1. McCune JM, Shi E, Perrig A, Reiter MK. Detection of denial-of-message attacks on sensor network broadcasts. In *Proceedings of IEEE Symposium on Security and Privacy*, 2005.
2. Byers J, Luby M, Mitzenmacher M, Rege A. A digital fountain approach to reliable distribution of bulk data. In *Proceedings of ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, 1998; pp. 56–67.
3. IETF RMT Working Group. Reliable multicast transport (RMT) charter. http://www.ietf.org/html. charters/rmt-charter.html
4. Ni S-Y, Tseng Y-C, Chen Y-S, Sheu J-P. The broadcast storm problem in a mobile ad hoc network. In *Proceedings of ACM Conference on Mobile Computing and Networks (MobiCom)*, 1999.
5. Pagani E, Rossi G. Reliable broadcast in mobile multihop packet networks. In *Proceedings of ACM Conference on Mobile Computing and Networking (MobiCom)*, 1997.
6. Lim H, Kim C. Flooding in wireless ad hoc networks. *Computer Communications* 2001; **24**(3): 353–363(11).
7. Cagalj M, Hubaux J-P, Enz C. Minimum-energy broadcast in all-wireless networks: NP-completeness and distribution issues. In *Proceedings of ACM Conference on Mobile Computing and Networking (MobiCom)*, 2002.
8. Yi Y, Gerla M, Kwon TJ. Efficient flooding in ad hoc networks using on-demand (passive) cluster formation. In *Proceedings of ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, 2002.

DOI: 10.1002/sec

9. Lim H, Kim C. Multicast tree construction and flooding in wireless ad hoc networks. In *Proceedings of the 3rd ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWIM)*, 2000; pp. 61–68.
10. Stojmenovic I, Seddigh M, Zunic J. Internal node based broadcasting algorithms in wireless networks. In *Proceedings of the Hawaii International Conference on System Sciences*, 2001.
11. Perrig A, Szewczyk R, Wen V, Culler D, Tygar J. SPINS: security protocols for sensor networks. In *Proceedings of ACM Conference on Mobile Computing and Networks (MobiCom)*, 2001.
12. Michiardi P, Molva R. CORE: A COllaborative REputation mechanism to enforce node cooperation in Mobile Ad Hoc Networks. *Communication and Multimedia Security*, September 2002.
13. Ganeriwal S, Srivastava M. Reputation-based framework for high integrity sensor networks. In *Proceedings of 2nd ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN'04)*, 2004.
14. Buchegger S, Le Boudec J-Y. Performance analysis of the CONFIDANT Protocol (Cooperation Of Nodes—Fairness In Dynamic Ad-hoc NeTworks). In *Proceedings of ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, 2002.
15. Mundinger J, Le Boudec J-Y. Analysis of a reputation system for mobile ad-hoc networks with liars. In *Proceedings of The 3rd International Symposium on Modeling and Optimization*, 2005.
16. Abdalla M, Shavitt Y, Wool A. Key management for restricted multicast using broadcast encryption. *IEEE/ACM Transactions on Networking (TON)* 2000; **8**(4): 443–454.
17. Perkins CE, Belding-Royer EM, Das SR. IP flooding in ad hoc mobile networks. In *IETF Internet-Draft*, 2001.
18. Lou W, Wu J. On reducing broadcast redundancy in ad hoc wireless networks. In *IEEE Transactions on Mobile Computing* 2002; **1**(2): 111–122.
19. Buchegger S, Le Boudec J-Y. A robust reputation system for peer-to-peer and mobile ad-hoc networks. In *Proceedings of Third Workshop on Economics of Peer-to-Peer Systems (P2PECON)*, 2004.
20. Srinivasan A, Wu J. A novel k-parent flooding tree for secure and reliable broadcasting in sensor networks. In *Proceedings of IEEE International Conference on Communications—Computer and Communications Network Security (ICC CCN)*, 2007.

DOI: 10.1002/sec