A Novel k-parent Flooding Tree for Secure and Reliable Broadcasting in Sensor Networks

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

***Abstract*— Securing broadcast communication over sensor networks is an important research challenge. In general, broadcast communication has two important metrics: security and reliability. Though the reliability metric has drawn sufficient attention in the research community, the security metric has failed to do so. In this paper we address both these metrics with more emphasis on the former and address the Denial-of-Broadcast Message attacks (DoBM) in sensor networks. We propose a tree based model called the k-parent Flooding Tree Model (k-FTM) and present algorithms for the construction of k-FTM. The proposed k-FTM is robust against DoBM and enables the base station 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 secure and reliable. We confirm through simulations that our model achieves detection rates close to a static tree and a broadcast reliability close to blind flooding.**

***Index Terms*— Adversary, broadcast communication, flood tree, malicious, reputation, sampling, security, simple majority, trust.**

The following pairs of words are used interchangeably in this paper: (internal, broadcasting), (adversary, attacker) and (compromised, malicious).

I. 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. In particular, broadcasting forms the basis of all communications in wireless networks. However, broadcast communication, unfortunately, has not received much attention in sensor networks. Broadcast communication has two very important metrics: security and reliability. Though the reliability metric has drawn the attention of researchers, very few researchers have addressed the security metric of broadcast communication in sensor networks [12]. Intuitively, broadcasting has two phases: the broadcast phase and the acknowledgement phase and is vulnerable to attacks during either or both these phases. The attacker's motive, however, is to disrupt the communication by blocking the message to as many nodes as possible but by attacking during the acknowledgement phase, the attacker is merely increasing the false alarm rate. This does not benefit the attacker since it only increases the probability of his detection and hence we assume attacks are only during the broadcast phase.

In a one-to-all (OTA) communication like broadcasting, in which every message 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. This, however, is not a scalable solution, in particular for large networks, as it increases the network load leading to higher natural loss [2], [14]. Consequently, if the detection threshold is not tuned to accommodate the right amount of natural loss, then 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. The aforementioned problem can be addressed by using a technique like Secure Implicit Sampling (SIS) [12] in which only a subset of nodes are randomly sampled to acknowledge during each broadcast.

A Flooding Tree Model (FTM) can be regarded as a OTA communication model in which message flows from the root towards the leaves which ideally suits the requirements of broadcast communication. When a FTM is used for broadcasting, an attacker can be detected immediately since each compromised node will be, in effect, blocking the message to its whole subtree. Therefore, high detection rate comes at the cost of increased damage to the system since the attacker can cripple a substantial portion of the network by compromising a single node with a large subtree. So, every node in the tree is a potential point of failure, in particular nodes with large subtrees. For this reason fault tolerance has been an issue of concern in flooding tree based broadcasting.

In an FTM, two aspects of a malicious node impact the system throughput: height and node degree. For instance, in Figure 1, circled node 3 has height 2 and blocks two nodes where as circled node 14 has heights 1 but blocks three nodes. This clearly implies that the impact of height of the malicious node on throughput is less compared to the malicious node's degree. The aforementioned problem of coverage in FTM can be overcome by using blind flooding, in which each node rebroadcasts the message on receiving it for the first time. This ensures that all nodes receive the message which makes blind flooding highly fault tolerant, 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

This work was supported in part by NSF grants ANI 0073736, EIA 0130806, CCR 0329741, CNS 0422762, CNS 0434533, CNS 0531410 and CNS 0626240. Corresponding author: jie@cse.fau.edu.

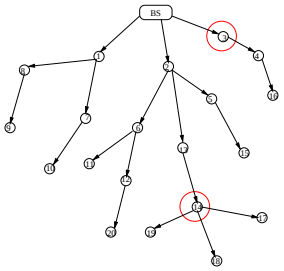


Fig. 1. Impact of height of the malicious node and node degree on throughput in FTM.

a serious problem, referred to as the broadcast storm problem [3], which causes communication congestion and contention. Hence, there is an inevitable tradeoff between reliability and redundancy.

In this paper, we propose the k-parent Flooding Tree Model (k-FTM) which is robust against Denial-of-Broadcast Message attacks (DoBM) in sensor networks. k-FTM is a variation of FTM in which all nodes, except those that are within the transmission range of the base station, have exactly k parents. When $k = 1$, k-FTM represents the basic FTM. The main motivation behind proposing k-FTM is to retain the high detection rate of FTM but at the same time achieve a reliability close to blind flooding. We have relaxed the k-parent constraint on nodes that are in the range of the base station since they are assured of message delivery. Also, these nodes will frequently encounter situations in which they will not find k distinct parents and this situation worsens as k increases. This results in the loop-back problem, a situation in which two nodes end up as both parent and child of each other.

In k-FTM, blind flooding is carried out once to construct the initial k-parent flooding tree and all subsequent messages are broadcasted along this tree. We chose to use a static k-parent tree rather than a dynamic tree to take advantage of internal node based broadcasting. This reduces the redundancy in communication, consequently reducing collision and contention. k-FTM is an excellent fault tolerant model and to our best knowledge the first fault tolerant tree model to be applied for securing broadcast communication. In k-FTM, every node is robust against $k-1$ malicious parent nodes. Therefore, k-FTM overcomes the above discussed drawback of FTM $\forall k > 1$ and achieves a reliability close to blind flooding. Using k-FTM is a two fold benefit to the system: First, the attacker's malicious intentions are thwarted since the message is delivered to the blocked nodes by their remaining $k - 1$ parents. Second, the attacker is immediately detected since a compromised node's child nodes report to the base station on failing to receive the k^{th} copy of the message. Subsequently we will show that our model is an optimum balance between FTM and blind flooding. Our contributions in this paper are as follows

- We propose a novel k-parent Flooding Tree Model, k-FTM, that has a very high detection rate.
- We present algorithms for constructing the k-FTM.
- The proposed k-FTM is the first fault tolerant tree model for securing broadcast communication in sensor networks.
- The proposed k-FTM is the first tree model that achieves broadcast reliability close to blind flooding with reduced redundant rebroadcasts.
- We analyze our model and evaluate it through simulations.

The rest of this paper is organized as follows. In section II we discuss the related work. In section III, we present the underlying basic model followed by a detailed discussion of our k-FTM. Section IV provides analysis of our model. In section V, we present simulation results and finally in section VI we conclude our work with directions for future research.

II. Related Work

Though researchers have addressed the problem of energy efficient and reliable broadcasting in wireless and sensor networks [1], [3], [6], [7], [9], [10], [11], security issues have not been addressed adequately. In [3], 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 [4], [6], 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 [5] 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. [8] 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 broadcast for severely resource-constrained environments.

McCune et al have proposed Secure Implicit Sampling (SIS) for detection of denial of message attack on sensor network broadcasts [12]. In SIS, using appropriate cryptographic functions and pseudo-random keys, the base station encrypts the message and broadcasts it, in which it is encoded which nodes are required to acknowledge. The adversary has no way of knowing a priori the subset of nodes that will be sampled for each round. On receiving the message, each node authenticates the message and if required sends back an acknowledgement (ACK). On receiving the ACK, the base station authenticates it and for each round computes the ratio of the number of expected ACKs, R_i, to the number of received ACKs, S_i. If this ratio is below a certain threshold h, i.e., $\frac{R_i}{S_i} < h$, then the system will raise an alarm.

In [12], the probability of the attacker remaining undetected varies with the position of the attacked node. It generally increases with the distance between the attacked node(s) and the base station. Therefore, when acknowledgement (ACK) from a geographically far node is lost, there is a very high probability for the base station to attribute it to natural loss. Additionally, when fewer nodes are attacked the probability of sampled nodes being blocked decreases and consequently, the probability of the attacker's detection also decreases. We thus advocate the use of k-FTM in such scenarios.

III. k-FTM

In this section we discuss k-FTM in depth. We begin by comparing and contrasting k-FTM and blind flooding. Then we present the attacker model followed by the list of underlying assumption of our model. We then provide an overview of the basic model followed by a detailed discussion on the working of k-FTM.

A. k-FTM vs 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 if a network uses blind flooding for each message broadcast. In this subsection we compare and contrast k-FTM with blind flooding.

In blind flooding, each node receives multiple copies of the same message and the number of copies is equal to the number of nodes in its neighborhood. In a dense network this will result in very high redundancy. Where as in k-FTM, a node receives only k copies of the message, one from each of its k-parents, where k can be as small as 1. This results in reduced redundant rebroadcasts. For example, in Figure 3 (a), there are 15 broadcasting nodes and 5 non-broadcasting nodes which reduces the number of rebroadcasts by 25%. Also, since k is a controlled parameter, the redundancy can be varied depending on the application domain as well as security and reliability requirements.

In blind flooding, every node rebroadcasts the message on receiving it for the first time where as in k-FTM, only internal nodes rebroadcast the message. 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 non-broadcasting nodes.

B. Attacker Model and Assumptions

Attacker Model: In k-FTM, the attacker is external to the network who randomly chooses nodes in the network and compromises them. Compromised nodes attack during the broadcast phase by not forwarding the broadcast message to their children. Note that compromised nodes do not attack during the acknowledgement phase as this only increases the probability of the attacker's detection. The attacker is not aware of the network topology which makes it more difficult for him to choose nodes for launching his attack. The attacker can only compromise a node. He can never restore a compromised node to its original configuration nor can he compromise the base station. He may choose to compromise fresh benign nodes for each broadcast round or he may compromise a set of benign nodes initially and not bother thereafter. There may be other attacker models in which the adversary's strategy could be to attack during acknowledgement phase, raise false alarms, and engage the system in counter measure. We will not focus on such attacker models in this paper. In our k-FTM, the adversary will not compromise nodes during the initial tree setup phase because blocking the message during this phase will render the malicious nodes as non-broadcasting nodes and consequently they will not be able to induce any damage during the actual message broadcast.

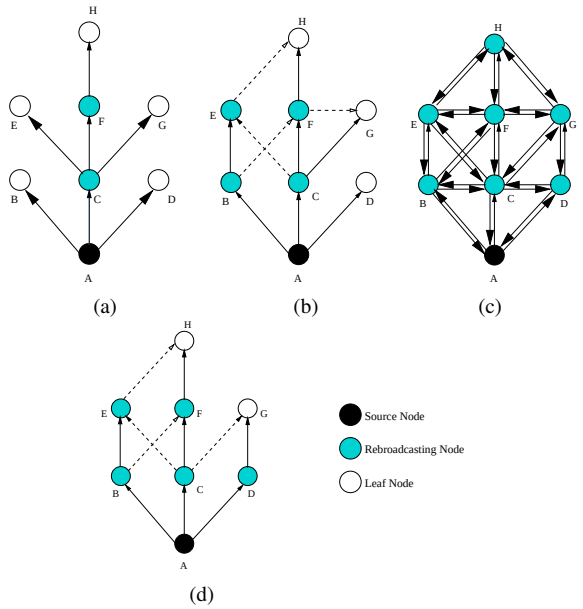


Fig. 2. (a) FTM for a network with 8 nodes. (b) k-FTM for a network with 8 nodes where $k = 2$.(c) Blind Flooding in a network with 8 nodes. (d) A 2-FTM where every node has 2 parents with disjoint path to base station.

Assumptions: We do not address node failure and consider only non-forwarding attacks in a static network. The attacker's sole intention is to prevent broadcast messages from reaching as many nodes as possible. When sampled, malicious nodes promptly acknowledge and never drop acknowledgements of other nodes. Malicious nodes do not collaborate and the attacker cannot inject fabricated ACKs. Nodes have uniform transmission range and the rate of message propagation is uniform. There is a secure key-management protocol to establish pair-wise keys between each node and the base station. Each node, excluding 1-hop neighbors of the base station, has exactly k parents. Nodes can have any number of children. Finally, in k-FTM, omni directional transmission is used and selective forwarding is not permitted.

C. Basic Model Overview

The basic underlying model is a flooding tree in which only internal nodes rebroadcast. Constructing an FTM is similar to the CDS problem. Given a set of vertices V, CDS is a set $S \subset V$ such that $\forall n \in V - S$, n has a neighbor in S. Extending this to k-FTM, we have the k-CDS problem which is, given a set of vertices V, k-CDS is a set $S \subset V$ such that $\forall n \in V - S$, n has at least k neighbor in S. Figure 2 (a) is an example of FTM and Figure 2 (b) is an example of k-FTM where $k = 2$. Figure 2 (c) represents blind flooding.

Figure 3 (a) is an example of a k-FTM where $k = 2$. The solid edges and the dashed edges indicate the 2 parents of a node and in Figure 3 (b), the dotted lines indicate the ACK path. In k-FTM, the base station carries out an initial round of flooding with a *Hello* message, after node deployment, during which the k-parent tree is constructed. This tree will be used for broadcasting as well as receiving ACKs in the

Algorithm 1

Initialize

for each node i in the network **do**
  if $i \in Transmission\_Range$ of base station **then**
    $parent\_list_i \leftarrow parent\_list_i \bigcup base\_station$;
    $State_i \leftarrow broadcasting$;
  else
    $parent\_list_i \leftarrow \emptyset$;
    $State_i \leftarrow non\text{-}broadcasting$;
  end if
  $child\_list_i \leftarrow \emptyset$;
end for

future. It is assumed that there are no topological changes and the network remains stable thereafter. However, the base station may choose to rebuild the tree after a predetermined number of reports of misbehavior have been received or after a timeout $Timeout_{tree}$ occurs. Rebuilding the tree has three main advantages. First, it prevents a dormant malicious node from getting familiar with the locality thereby curtailing the damage it can cause in the future. Second, it renders a malicious 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-parent tree. We will discuss in detail on how this is accomplished in later sections when we discuss the working of our model. Third, it ensures that nodes are moved around so that all nodes get a fairly equal chance of being both broadcasting and non-broadcasting nodes.

Every node on receiving the *Hello* message the first time rebroadcasts it in its neighborhood. Each node, except 1-hop neighbors of the base station, also acknowledges as child to exactly k nodes from which it receives the *Hello* message and enters the IDs of these nodes in its $parent\_list$. During subsequent broadcast rounds, nodes receive message only from the nodes in the $parent\_list$. However, the way in which k-parents are chosen can differ, which we will discuss later in this subsection. After rebroadcasting the message, nodes wait till $Timeout_{ACK}$ occurs. If a node receives acknowledgements within $Timeout_{ACK}$, then it stores the IDs of the acknowledging nodes in its $child\_list$ and sets its state as broadcasting. Else, it sets its state as non-broadcasting and does not rebroadcast during subsequent broadcast rounds. Once a node establishes all k-parents and $Timeout_{ACK}$ occurs, it sends a copy of its $parent\_list$ and $child\_list$ to the base station.

Construction of k-FTM: Below are three different methods for choosing the k parents such that the system throughput is maximized while keeping the detection rate as high as possible. We present algorithms for these methods and discuss them briefly.

The first method is called the *Fastest First k-parents* in which a node acknowledges as child to the first k rebroadcasting nodes from which it receives the *Hello* message. This method has been presented in Algorithm 2. In this method, the node appends its ID to the $path\_label$[1] received from the first parent and rebroadcasts it.

[1]List containing IDs of nodes the message has passed through hence far starting from the base station. It enables every node to trace its route back to the base station.

Algorithm 2

Fastest First k-parents

Initialize;
for each node i not in $Transmission\_Range$ of base station **do**
  $parent\_count_i \leftarrow 0$;
  while $parent\_count_i < k$ and $!Timeout_{msg}$ **do**
    for ($message$, $path\_label$) received from each neighbor j **do**
      if $message$ is fresh **then**
        append ID to $path\_label$
        rebroadcast ($message$, $path\_label$);
      end if
      $parent\_list_i \leftarrow parent\_list_i \bigcup j$;
      $child\_list_j \leftarrow child\_list_j \bigcup i$;
      $parent\_count_i + +$;
      if $child\_list_j \neq \emptyset$ **then**
        $State_j \leftarrow broadcasting$;
      end if
    end for
  end while
end for

The second method is called the *Disjoint Path k-parents* in which a node receives *Hello* message from all its neighbors till $Timeout_{msg}$ occurs. Nodes that broadcast the message after $Timeout_{msg}$ are ignored during parent selection. A node, while choosing its k-parents, chooses nodes with disjoint $path\_labels$. The advantage of this method is that it augments system throughput significantly. A node may encounter a situation wherein it may not find all k-parents with disjoint $path\_labels$. Under this circumstance, assuming a node finds only m parents with disjoint $path\_labels$ such that $m < k$, it will choose the remaining $(k - m)$ parents according to first-come-first-served principle. Algorithm 3 illustrates this method in detail. After choosing the k-parents, a node appends its ID to all the $path\_labels$ received from the k-parents and then rebroadcasts it along with the message. This facilitates subsequent nodes in choosing their parents with disjoint paths to the base station. Figure 2 (d) is an example of a 2-FTM where each node has 2 parents with disjoint paths to the base station. For instance, in Figure 2 (d), node H receives the following $path\_labels$: $(A : B : E)$, $(A : C : E)$, $(A : C : F)$, $(A : C : G)$, and $(A : D : G)$ and node H chooses $(A : B : E)$ and $(A : C : F)$. To reduce the $path\_label$ overhead introduced by this method, we can have a node rebroadcast the message by appending its ID to the $path\_label$ from only one of its parent's with disjoint path to the base station. This improved method is called the *Improved Disjoint Path k-parents*.

The third method is called the *Unique Level*-1 *Ancestors* which is very similar to the *Disjoint Path k-parents* method. However, in *Unique Level*-1 *Ancestors* only level-1 nodes attach their ID and rebroadcast the message. When choosing the k-parents, a node chooses its parents with unique level-1 ancestors and if the node cannot find all k-parents with unique level-1 ancestors, this method reverts to the first-come-first-served principle to choose the remaining parents.

In all these methods, there is one problem that needs a

Algorithm 3

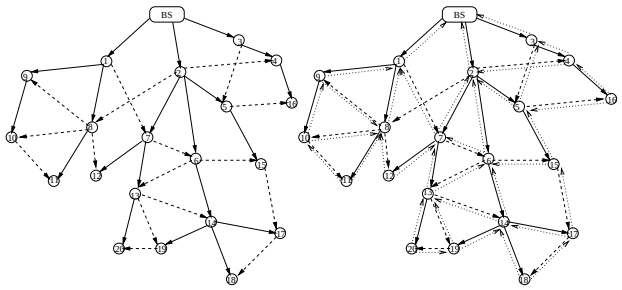
Disjoint Path k-parents

```
Initialize;
for each node i not in Transmission_Range of base station do
    while !Timeout_msg do
        for (message, path_labels) received from each neighbor j do
            if message is fresh then
                Put (message, path_labels) in buffer;
            else
                Put path_labels in buffer;
            end if
        end for
    end while
    Choose max available parents with disjoint path_labels;
    Choose remaining parents on FCFS basis;
    add all k-parents to parent_list_i;
    for each parent j selected do
        add i to child_list_j;
        State_j ← broadcasting;
    end for
    Append ID to all path_labels received from the k-parents and rebroadcast along with message;
end for
```

special mention. This is the loop-back problem in which two nodes end up being both child and parent of each other. To eliminate this situation, we can introduce a constraint in which a node will acknowledge as child only to node(s) with higher node ID ($A > B$). For instance, in Figure 2 (b), node G can choose node F as its parent but node E cannot choose node F as its parent due to the aforementioned constraint. However, in this paper, we use a different method to deal with this loop-back problem. In k-FTM, each node compares its $parent\_list$ and $child\_list$ to make sure no node appears in both these lists. For clarity, consider two nodes i and j. Let node j appear in both $parent\_list_i$ and $child\_list_i$. Then, node i has to appear in $parent\_list_j$ and $child\_list_j$. Now, both i and j compare their $parent\_list$ and $child\_list$ and realize that there is a loop-back. They immediately exchange messages with each other and come to an agreement where the node with a higher ID, i, is declared as parent and the node with a lower ID, j, is declared as child. Following this, node j is erased from $parent\_list_i$ and node i is erased from $child\_list_j$. Node i then sends a message to all its children so that they can update their $path\_labels$. Nodes i and j also send a copy of their updated $parent\_list$ and $child\_list$ to the base station.

D. Working of k-FTM

In k-FTM, using a technique like SIS, the base station encodes in each broadcast message which nodes are expected to acknowledge. On receiving the message, nodes determine if they are expected to ACK. If yes, then they send back an ACK to the base station. This is known as reactive acknowledgement ACK_{Rea}. However, nodes can send ACKs even if they are not asked to and this is known as proactive acknowledgement ACK_{Pro}. We shall discuss these two ACK schemes in detail later in this section. In either case, the ACK is sent along



(a) (b)

Fig. 3. (a) k-FTM for a network of 20 nodes ($k = 2$). Origin of solid line indicates the first parent and origin of dashed line indicates the second parent. (b) k-FTM for a network of 20 nodes ($k = 2$). The dotted line indicates the path of the ACKs.

only one of the k-parents and this is the parent the node trusts the most based on the reputation value. If there is a tie in the reputation values of two or more parents, then the node chooses any one parent randomly. How nodes monitor the trust level of their children and parent nodes using reputation metric and how the reputation metric itself is initialized, maintained and utilized in the decision making process will be discussed later.

Reactive ACK: When a node is implicitly directed in a broadcast message to ACK, it is known as reactive acknowledgement ACK_{Rea}. ACK_{Rea} contains a field that is k-bits long. The value of this field indicates if the parents of a sampled node successfully forwarded the message. A binary 1 in these k-bits indicates that the corresponding parent forwarded the message and a 0 otherwise. If none of the k-parents of the sampled node forward the message, then the base station will not receive ACK_{Rea} from that node.

Proactive ACK: In the proactive acknowledgement scheme, any node that does not receive k copies of the message sends an ACK_{Pro} immediately to inform the base station. In this scheme, it is important to note that due to the promiscuity of wireless transmission, the parents of the node that does not forward the message also can send an ACK_{Pro} to inform the base station. This helps in preventing bad mouthing attacks to a large extent. ACK_{Pro} also has a k-bit field to indicate which parent(s) failed to forward the message. In addition, it has an n-bit field (n is the number of children reporting node has) to indicate which child nodes failed to forward the message.

Every ACK has a separate 1-bit, which the base station checks first on receiving an ACK, that indicates whether it is an ACK_{Rea} or ACK_{Pro}. This field is set to 1 if it is ACK_{Rea} and 0 otherwise. In addition, the base station on receiving an ACK first confirms if the reporting node is actually a parent and/or child of the reported node using the $parent\_list$ and/or $child\_list$ respectively, and if otherwise punishes the reporting node for badmouthing. This is considered the first level of sanity check against bad mouthing attacks. It also checks if the reported node is a non-broadcasting node and if true then any message against that node is discarded and the reporting parent node is not punished for bad mouthing because nodes in k-FTM do not have node degree information of their 1-hop neighbors. The base station, as a preventive

mechanism to eliminate bad mouthing uses a simple majority scheme as follows. Since omnidirectional transmission is used, if a compromised node blocks the message, then none of its n children should receive the message and all its k-parents should observe this. So, the total number of reports at the base station will be $k+n$. Considering natural loss, the base station sets a benchmark and whenever node i is reported as not forwarding the message by its child and parent nodes, the base station increments $count_i$, a counter that keeps track of node i's misbehavior, only if more than $\lfloor \frac{n}{2} \rfloor + 1$ child nodes and $\lfloor \frac{k}{2} \rfloor + 1$ parent nodes report the misbehavior. Else, the reporting node is punished for badmouthing and its misbehavior counter is incremented. When $count_i$ exceeds $Threshold_{mal}$, node i is declared malicious. In addition, the base station maintains a global counter, $count_{Glo}$, which is incremented every time $count_i$ is incremented for some i in the network. $count_{Glo}$ keeps track of overall level of misbehavior in the network.

Finally, when a sampled node i does not send back ACK_{Rea}, the base station first checks $parent\_list_i$. Then it checks to see if there are any ACK_{Rea} or ACK_{Pro} from other nodes against the nodes in $parent\_list_i$. If there are no reports, then it is very likely that the node has either failed or the ACK_{Rea} has been lost enroute due to contention/collision. But if there are reports, then the simple majority scheme as explained above is employed to punish node i.

Reputation Metric: In k-FTM, each node maintains a metric called reputation to monitor the trustworthiness of its parents and children. Each time a node fails to forward the broadcast message to its children, both child and parent nodes of the misbehaving node update their reputation metric for that node. Later, this information will be used by the node for two purposes: First, in choosing the parent along which the ACK will be sent to the base station. Second, in choosing the new k-parents when the tree is reconstructed. The reputation metric is represented as $R_{i,j}$, which represents the accumulated reputation of node j from node i's perspective. At the beginning, when time $t = 0$, every node has a neutral impression about every other node in its neighborhood. This assumption is necessary to begin the k-parent tree setup and holds true only during this phase. Once the k-parent tree is setup every node sets the reputation value of nodes in its $parent\_list$ and $child\_list$ to 0 and starts monitoring them and accordingly updates their reputation value.

$$R_{i,j}^{New} = \alpha_1 \times R_{i,j}^{Current} + \alpha_2 \times \lambda \quad (1)$$

$$R_{i,j}^{New} = \alpha_1 \times R_{i,j}^{Current} - \alpha_2 \times \lambda \quad (2)$$

where α_1 and α_2 are weights assigned to the past behavior and the behavior during current broadcast round respectively. The values of α_1, α_2, and $R_{i,j}$ are range bound between $0-1$ and $\alpha_2 = (1 - \alpha_1)$.

Now, consider three nodes i, j, and k such that nodes j and k are parents of node i. If i receives a copy of the message from j but does not receive a copy from k during the current broadcast round, then i uses Equation-1 to increment its trust in j and Equation-2 to decrement its trust in k. In both the cases the value of $\lambda = 1$. In each message, the base station also encodes a time interval δ which is an upper bound on time within which the node should receive the next message. If i received the previous message at time t' and does not receive the next message from either j or k till time $t' + \delta$, then it uses Equation-1 or 2 to decrement its trust in both j and k with $\lambda = 0$. This is referred to as aging. Therefore, in k-FTM, the reputation metric is both event and time driven. For interested readers, Srinivasan et al have provided a detailed discussion on reputation and trust based systems for ad hoc and sensor networks in [13].

The base station reconstructs the tree under three circumstances: $count_{Glo}$ exceeds $Threshold_{Glo}$, $count_i$ exceeds $Threshold_{mal}$ where $Threshold_{mal}$ is the threshold for individual node misbehavior or $Timeout_{tree}$ occurs. Prior to the reconstruction, the base station broadcasts a message informing nodes about the malicious node(s) so that they don't choose the malicious node(s) as one of their parents in the reconstructed k-FTM. There is a possibility that the warning message itself may be denied to nodes which is not a major threat since child and parent nodes of the malicious node in the current tree are keeping track of its misbehavior. The main purpose of broadcasting this message is to warn those nodes that are in the neighborhood 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. Now, after reconstruction, as a result of the warning message, the malicious node is forced to be either a non-broadcasting node or at best a node with low node degree.

IV. Analysis

In a k-FTM with n nodes, each node has a uniform probability of $\frac{1}{n}$ of being sampled by the base station for ACK. k-FTM has a very big advantage of near perfect detection rate when $k = 1$. This is because, the larger the subtree rooted at the malicious node, the higher are the chances of more sampled nodes being deprived. Therefore the adversary will be detected immediately. The survival time of the adversary is inversely proportional to the number of sampled nodes he deprives which is directly coupled to the total number of nodes attacked. However, there is a downside to FTM. A compromised node with a large subtree can potentially cripple a substantial portion of the network. This drawback of FTM is overcome in k-FTM $\forall k > 1$.

If sampled nodes are blocked by malicious nodes, then they fail to receive the broadcast message. Consequently, the base station will fail to receive their ACK. In an *ideal-world* situation where there is no natural loss, the base station will attribute this loss to DoBM. But in a *real-world* scenario where there is natural loss in the network, the base station 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 increases since, with an increase in k, the number of non-broadcasting nodes decrease. We confirm this through simulation.

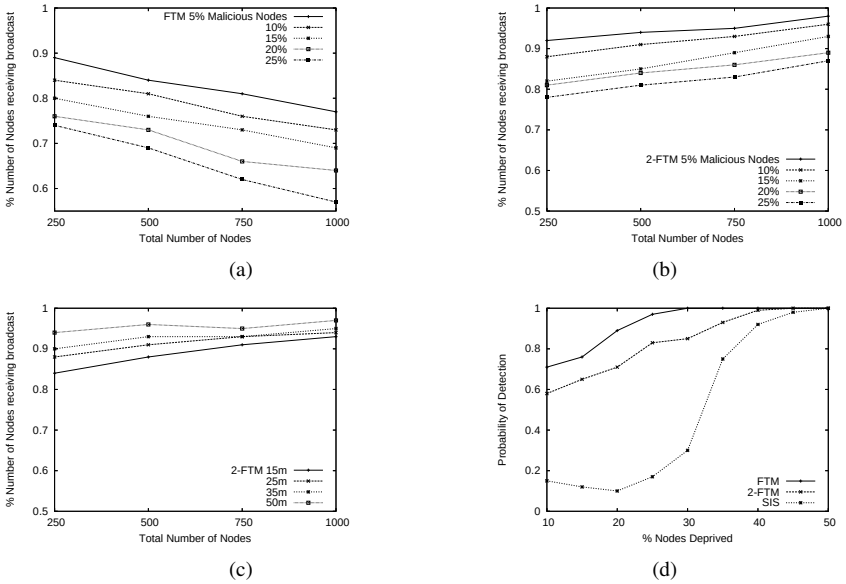


Fig. 4. (a) Impact of % of Malicious nodes on throughput in static tree. (b) Impact of % of Malicious nodes on throughput in k-FTM. (c) Impact of different transmission range on coverage.(d) Impact of % of Malicious nodes on probability of detection.

Since omni directional transmission is used in k-FTM, the energy consumption is high. This constraint is introduced to enable the nodes to promiscuously observe if their child nodes successfully forward the message. However, the energy consumed can be mitigated by allowing the nodes to choose a transmission range of their choice before the initial tree setup. Once a node choose its transmission range, it remains fixed for the lifetime of that node. Note that the nodes can choose their transmission range only from a fixed range, the minimum and maximum of which is set appropriately with security and reliability requirements in mind.

A. Adversary Tactics

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

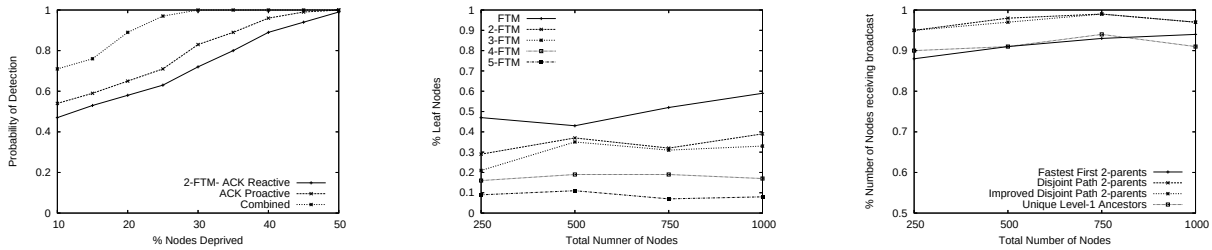
B. System Counter-Tactics

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 an irreparable damage to the network. However, there is an inevitable tradeoff between good performance and a high detection rate. A 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 very high detection rate is the requirement, tree broadcasting suits best. By using FTM to achieve a high detection rate, the system sacrifices its performance, since a single malicious node that roots a large subtree can cripple a substantial portion of the network. Therefore, in this case, a high level of security comes at the cost of system performance. However, the proposed k-FTM $\forall k > 1$ strikes a near optimal balance between blind flooding and FTM by retaining a reliability close to blind flooding and a detection rate close to FTM.

V. Simulation and Results

We consider a wireless sensor network consisting of n homogeneous sensors $s_1, s_2, ..., s_n$ and model the network as a graph $G = (V, E)$. The set of vertices represents the set of sensors. An edge exists between two nodes if they are in each other's communication range and share parent-child relationship. The root represents the base station and all other nodes represent sensors. The simulation has been carried out on a custom simulator written in JAVA. All the simulations have been performed for varying numbers of sensor nodes, 250, 500, 750 and 1000, deployed over an area of $500m \times 500m$. Statistics have been collected for 200 trials of 1000 iterations each and averaged.

In Figure 4 (a), the results for the number of nodes actually receiving the broadcast message with varying percentages of malicious nodes in a FTM has been presented. Similarly, in Figure 4 (b), we have presented the results for throughput in k-FTM where $k = 2$. We shall call this model as 2-FTM. It is very clear from the graph that the throughput in 2-FTM



(a) (b) (c)

Fig. 5. (a) Performance of k-FTM with different types of ACKs. (b) % of Leaf nodes with varying number of parents. (c) Comparison of different methods of k-FTM construction.

reaches a maximum of about 98% with 5% malicious nodes and reaches a minimum of about 78% with 25% malicious nodes where as in FTM it reaches a minimum value of 57% with 25% malicious nodes. Therefore, 2-FTM successfully achieves a significantly higher throughput when compared to FTM. This throughput is nearly as good as in blind flooding but with reduced number of rebroadcasts.

Figure 4 (c) is a comparison of system throughput with different transmission ranges. The system performs best with 50m range. However, at 50m the energy consumption will also be higher. We can see that it performs reasonably well even at 25m range. In Figure 4 (d), we have plotted the results comparing percentage of nodes deprived against probability of detection. We have compared FTM and 2-FTM with the SIS model [12]. Both these models out-perform SIS in terms of probability of detection.

Figure 5 (a) compares the performance of 2-FTM under three different scenarios. In the first scenario only ACK_{Rea} is considered and has the poorest detection performance. In the second scenario only ACK_{Pro} is considered and has better performance compared to ACK_{Rea}. The best performance, however, is achieved when both ACK_{Rea} and ACK_{Pro} are combined. In Figure 5 (b), we have plotted the percentage of non-broadcasting nodes against total number of nodes varying k from 1 to 5. We can see that as k increases, the percentage of non-broadcasting node decreases. Finally in Figure 5 (c), we have compared the performance of different methods of construction of k-FTM for $k = 2$. We see that *Disjoint Path k-parents* and *Improved Disjoint Path k-parents* have almost the same performance and achieve the best coverage.

VI. Conclusion

We have proposed a novel k-parent Flooding Tree Model (k-FTM) to efficiently address both security and reliability metrics of broadcast communication in sensor networks. k-FTM is very robust and efficient in detecting DoBM and to our best knowledge the first fault tolerant tree model which has reliability close to blind flooding and a detection rate close to a static tree. We have also presented various methods with algorithms for construction of k-FTM. In our future work, we would like to investigate the possibility of using directional/sectored antennas instead of omni directional antennas to mitigate energy consumption. We will also investigate the correlation between the network topology and other attacks on sensor network communication and do a more in-depth analysis and simulation of k-FTM. Wireless bandwidth is consumed whenever a node broadcasts a packet. Hence, using k-FTM can save bandwidth since in k-FTM only internal nodes rebroadcast. In our future work, we wish to simulate and see how much bandwidth can be saved and how much energy does it save for non-broadcasting nodes.

References

[1] E. Pagani and G. Rossi. Reliable broadcast in mobile multihop packet networks. *In Proceedings of the ACM Conference on Mobile Computing and Networking (MobiCom)*, pages 34-42, 1997.

[2] J. Byers, M. Luby, M. Mitzenmacher, and A. Rege. A Digital Fountain approach to reliable distribution of bulk data. *In Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols forComputer Communication*, pages 56-67, 1998.

[3] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and J.-P. Sheu. The broadcast storm problem in a mobile ad hoc network. *In Proceedings of the ACM Conference on Mobile Computing and Networks (MobiCom)*, Aug. 1999.

[4] H. Lim and C. Kim 2000. 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 (Boston, Massachusetts, United States, August 20 - 20, 2000). MSWIM '00. ACM Press, New York, NY, 61-68.*

[5] I. Stojmenovic, M. Seddigh and J. Zunic. Internal node based broadcasting algorithms in wireless networks. *Proceedings of the Hawaii Int. Conf. on System Sciences* (Jan. 2001).

[6] H. Lim and C. Kim. Flooding in wireless ad hoc networks. *Computer Communications*, 24, Feb. 2001.

[7] I. Stojmenovic, M. Seddigh, and J. Zunic. Dominating sets and neighbor elimination-based broadcasting algorithms in wireless networks. *IEEE Transactions on Parallel and Distributed Systems*, 12(12):1-12, Dec. 2001.

[8] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J. Tygar. SPINS: Security protocols for sensor networks. *In Proceedings of the ACM Conference on Mobile Computing and Networks (MobiCom)*, July 2001.

[9] W. Lou and J. Wu. On reducing broadcast redundancy in ad hoc wireless networks. *IEEE Transactions on Mobile Computing*, 1(2):111-122, Apr. 2002.

[10] M. Cagalj, J.-P. Hubaux, and C. Enz. Minimum-energy broadcast in all-wireless networks: NP-completeness and distribution issues. *In Proceedings of the ACM Conference on Mobile Computing and Networking (MobiCom)*, Sept.2002.

[11] Y. Yi, M. Gerla, and T. J. Kwon. Efficient flooding in ad hoc networks using on-demand (passive) cluster formation. *In Proceedings of the ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, 2002.

[12] J. M. McCune, E. Shi, A. Perrig, and M. K. Reiter. Detection of Denial-of-Message Attacks on Sensor Network Broadcasts. *In the Proceedings of the 2005 IEEE Symposium on Security and Privacy*.

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

[14] IETF RMT Working Group. Reliable multicast transport (RMT) charter. http://www.ietf.org/html. charters/rmt-charter.html.