Performance Analysis of Group Based Detection for Sparse Sensor Networks

Jingbin Zhang, Gang Zhou†, Sang H. Son, John A. Stankovic, Kamin Whitehouse
Computer Science Department, University of Virginia, Charlottesville, VA 22903
{jz7q, son, stankovic, whitehouse}@cs.virginia.edu
†Computer Science Department, College of William and Mary, Williamsburg, VA 23187
†{gzhou}@cs.wm.edu

Abstract

In this paper, we analyze the performance of group based detection in sparse sensor networks, when the system level detection decision is made based on the detection reports generated from multiple sensing periods. Sparse deployment is essential for reducing cost of large scale sensor networks, which cover thousands of square miles. In a sparse deployment, the sensor field is only partially covered by sensors' sensing ranges, resulting in void sensing areas in the region, but all nodes are connected through multi-hop networking. Further, due to the unavoidable false alarms generated by a single sensor in a network, many deployed systems use group based detection to reduce system level false alarms. Despite the popularity of group based detection, few analysis works in the literature deal with group based detection. In this paper, we propose a novel approach called Markov chain based Spatial approach (M-S-approach) to model group based detection in sensor networks. The M-S-approach successfully overcomes the complicated conditional detection probability of a target in each sensing period, and reduces the execution time of the analysis from many days to 1 minute. The analytical model is validated through extensive simulations. This analytical work is important because it provides an easy way to understand the performance of a system that uses group based detection without running countless simulations or deploying real systems.

1 Introduction

Wireless sensor network researchers have been using dense sensor networks to track moving targets, like battlefield monitoring [1] [2]. When an enemy tank is simultaneously detected by multiple local sensors within a sensing period, a local leader sensor reports that an intruding target is recognized, and sends this intrusion information back to a base station through multiple hops. This design principle works well when the deployed sensor network has high node density so that enough sensing redundancy is available to cover the moving target at any location and at any time.

In this paper, we consider sparse sensor networks, where sensing coverage is very limited but communication coverage is available through multi-hop networking. A sparse sensor network is possible when sensor nodes' communication ranges are greater than twice their sensing ranges, which is not a problem with existing mainstream sensor devices [3] [4]. Sparse sensor networks are especially useful for emerging applications like border control and undersea surveillance. For example, thousands of cameras can be deployed at the border to detect illegal border crossers. For this application, it is too expensive to deploy a dense sensor network that covers every square inch of the border with multiple camera sensors, since cameras may only be able to see a short distance due to obstacles, nighttime, etc. An alternative and feasible solution is to deploy a sparse sensor network with much fewer cameras, which partially covers the border with void sensing areas allowed. Since the cameras can be equipped with tall antennae, the communication can go pretty far and all cameras in the network can still report detection information back to base stations through multiple hops. Also, considering the high cost of an undersea sensor [5] that is in the order of thousands of dollars, a sparse deployment achieves the tradeoff between the size of the surveillance area and the detection performance when only limited resources are available.

In real deployments, an individual sensor is inclined to generate false positive alarms due to limited sensing capabilities and environmental noise. In this paper, we use false alarms to denote node level false positive alarms. Because of false alarms, instantaneous detection that is based on a single detection report is problematic and group based detection becomes a better choice. In group based detection, the system level detection decision is made based on the detection reports generated by various sensors within multiple sensing periods. Only the detection reports generated in a sequence, which can be mapped to a possible target

track, are recognized as true target detections. In this case, most false alarms are filtered out, since false alarms are not likely to be generated in a sequence that can be mapped to a target track. Group based detection is widely used in real applications [1] [2] [6] due to its effectiveness in reducing system level false alarms. Thus, performance analysis based on group based detection is extremely important for understanding the performance of a real sensor network application.

This is the first paper to present an accurate theoretical model for analyzing the performance of group based detection in sparse sensor networks, where the system level detection decision is made based on the detection reports generated from multiple sensing periods. In this paper, we first reveal the significant complexities involved in the conditional detection probability of a target in each sensing period. Then we propose the Spatial approach (S-approach) to overcome the complicated conditional detection probability of the target in each sensing period. However, the S-approach incurs high computation overhead, which restricts its wide usage. To address the computation overhead problem in the S-approach, a novel approach called Markov chain [7] based Spatial approach (M-S-approach) is proposed. The model is validated and shown to be extremely accurate through extensive simulations. Our model quantifies the relationship between surveillance attributes and system parameters, which provides an easy way to understand the impact of various system parameters on surveillance performance.

The rest of this paper is organized as follows: we describe the terminology and assumptions used in this paper in Section 2. Then we present the analytical model of group based detection in Section 3. We validate the model in Section 4 and review related work in Section 5. Finally, we conclude our work and discuss future work in Section 6.

2 Terminology and Assumptions

We assume that sensor deployment conforms to a uniform random distribution, primarily for ease of analysis. We also believe that this is a reasonable assumption especially for sparse sensor networks. For example, in undersea sensor networks, both the randomness brought by deployment and sensor drift due to ocean flows [8] are common.

We assume that the sensing algorithm in each node is executed periodically and that at the end of each period the local sensing algorithm decides whether the target is detected or not in that period. We use sensing period to denote the period during which the sensing algorithm is executed. Parameter t is used to denote the length of a sensing period, which mainly depends on how long a specific sensing algorithm needs to sample the environment before a detection decision is made. We assume that in a sensing period, if a target is within a sensor's sensing range, the probability of the target being detected by the sensor in that period is P_d[1]. We use R_s to denote the sensor's sensing range and we assume that the sensing ranges of all the sensors are the same. During each sensing period, if a target is detected by a sensor, a detection report is generated by that sensor. For simplicity of presentation, we assume that the communication range is larger than twice the sensing range to support sparse deployment, that is, sensing coverage is not available but communication coverage is available through multi-hop networking.

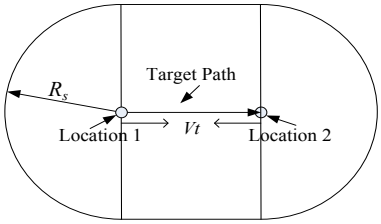


Figure 1. The Detectable Region (DR) of a target in one sensing period.

We use the Detectable Region (DR) of a target in a sensing period to denote the area, in which, if there is a sensor, it has probability P_d to detect the target in that period. So if the target is static, the DR of the target in a sensing period is a circular area, whose center is the target's location and radius is R_s. The size of the DR of a static target is πR_s^2. In this paper, we assume that the target travels in a straight line with a constant speed V and the size of the target can be neglected [2]. We also do not consider the situation when multiple targets cross paths, since we are currently considering very large areas and very rare events, like undersea submarine detection, in which one target at a time is very reasonable. If more than one target exist but are far from each other, our analysis still holds per target. In the future, we plan to deal with multiple targets that might be near each other and/or crossing. As shown in Figure 1, a target moves from location 1 to location 2 during a sensing period, so the size of the DR of the moving target in this sensing period is $2R_sVt + \pi R_s^2$.

In this paper, we assume that group based detection is used to reduce the system level false alarms. Without loss of generality, we abstract the group based detection algorithm as follows: the system level detection decision is made when the sensor network generates a sequence of at least k detection reports within M sensing periods that can be mapped to a possible target track; otherwise, detection re-

[1] We assume that P_d is independent of the length the target overlaps with the sensing range in a sensing period primarily for ease of analysis. This assumption will be revisited and revised in future work.

[2] In section 4, our performance evaluation shows that even when the target changes its moving directions frequently, our group based detection model still gives very good performance

ports are perceived as false alarms. The value of k is chosen based on the system's false alarm rate. If the false alarm rate is high, a large k is configured, so that a possible sequence of false alarms does not result in a system level false alarm. We assume that k is given based on empirically obtained false alarm patterns.

We analyze the detection probability of a target, given that the group based detection algorithm is known and the M and k values are specified. During the analysis, we do not consider false alarms. If we have false alarms mixed with real target detections, it only increases the probability of the real target being detected, since it results in more detection reports to be generated along the target track. Therefore, when our analysis result is applied to a real system that has false alarms, we may expect a little higher detection probability from the real system due to the impact of false alarms.

3 Analytical Model

In this section, we present the theoretical model for analyzing the performance of group based detection in wireless sensor networks. We first analyze the preliminary case in section 3.1 when the system level detection decision is made based on the detection reports from a single sensing period, i.e., $M = 1$. Then in Section 3.2, we reveal the difficulties involved in computing the conditional detection probability of a target in each sensing period when $M > 1$. After that, we present the Spatial approach (S-approach) to resolve that problem. However, the S-approach is restricted due to its high computation overhead. So we finally propose the Markov chain based Spatial approach (M-S-approach) that uses a Markov chain to overcome the high computation complexity in the S-approach.

3.1 Preliminaries

In this section, we consider the preliminary case when $M = 1$. When $M = 1$, the performance analysis of group based detection becomes straightforward and is addressed in [9].

We use p_{indi} to denote the probability of the target being detected by an individual sensor sampled from the sensor field in a sensing period. So, p_{indi} is the probability of an individual sensor sampled from the sensor field being in the DR of the target in that sensing period multiplied by P_d. We use S to denote the size of the sensor field. For a uniform random distribution, $p_{indi} = P_d \frac{2R_s Vt + \pi R_s^2}{S}$.

Suppose we have N sensors deployed in the sensor field. Since the nodes are uniformly and randomly distributed, the number of reports generated from the DR of the target in a sensing period conforms to a binomial distribution $B(N, p_{indi})$. Accordingly, $P_1[X = k]$, the probability of having exactly k detection reports in one sensing period when there is a target in the network, is computed as follows:

$$P_1[X = k] = \left(\begin{array}{c} N \\ k \end{array} \right) p_{indi}^k (1 - p_{indi})^{N-k} \quad (1)$$

So, $P_1[X \geq k]$, the probability of having at least k detections in one sensing period when there is a target in the sensor field, is computed as follows:

$$P_1[X \geq k] = 1 - \sum_{i=0}^{k-1} P_1[X = i] \quad (2)$$

Although [9] provides a neat solution to analyze the performance of a group based detection algorithm when $M = 1$, its usage is limited, especially when the deployment is sparse. In sparse deployments, M should be much greater than 1, since in sparse deployments, the probability of having more than one reports in one sensing period is very low even in the existence of a real target. In this case, if $M = 1$, k should also be set to 1, because if k is greater than 1, the probability of the target being detected will be very low. When both M and k are set to 1, group based detection becomes instantaneous detection, which is unable to filter any false alarms. This is true even if the deployment is not sparse. In real applications [1] [2], to effectively eliminate the system level false alarms, the system level detection decision should be made based on the detection reports from multiple sensing periods, rather than a single sensing period. This motivates the need to analyze the performance of group based detection when $M > 1$, which is almost totally different from the case when $M = 1$ and is much more challenging.

3.2 Detection Dependencies

The main difficulty of analyzing the performance of group based detection when $M > 1$ is that target detections in a sensing period are dependent on the target detections in the previous periods. We use a simple example to illustrate the detection dependencies in different sensing periods. In this example, $M = 2$.

Figure 2 shows the DRs of a target in sensing periods 1 and 2. The sensing period number is set to 1 when the target first appears in the sensor field. In this example, we assume that the target moves through location 1, 2, and 3, which lie on a straight line. To better show the DR of the target in each period, we move the DRs of period 2 vertically to avoid visual overlapping of these DRs. So location 2 shown in the DRs of period 1 and period 2 are actually the same location. The Newly Explored Detectable Region (NEDR) in a sensing period is defined as the area that belongs to the DR of that period but does not belong to the DRs of

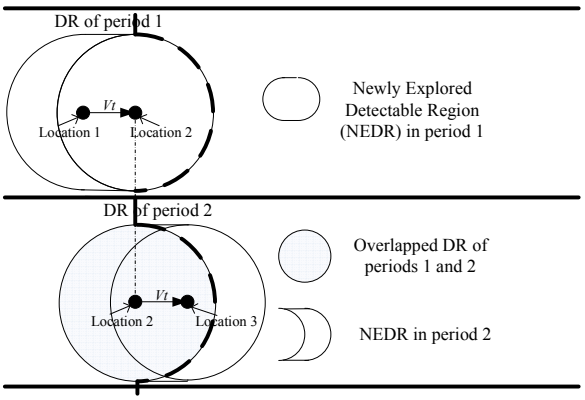


Figure 2. DRs of a target in sensing periods 1 and 2.

the previous periods. Because period 1 denotes the period when the target first appears in the sensor field, the NEDR of period 1 equals the DR of period 1. In other periods, the NEDR is only part of the DR in that period as shown in the figure, which is like a crescent. The overlapped DR of periods i, $i+1$, ..., and j, in which i and j are positive integers and $i < j$, is defined as the area that belongs to the DRs of periods i, $i+1$, ..., and j.

Let us first consider the detection probability of the target in period 2. In Figure 2, we can see that the number of reports generated in period 2 not only depends on the number of sensors in the NEDR of period 2, but also depends on the number of sensors in the overlapped DR of periods 1 and 2. Although the probability of having different number of sensors in the NEDR of period 2 is independent of the target detections in period 1, the probability of having different number of sensors in the overlapped DR of periods 1 and 2 is decided by the target detections in period 1. If the target meets a sensor in the overlapped DR of periods 1 and 2 in period 1, the probability of having a sensor in the overlapped DR of periods 1 and 2 in period 2 is 100%. The conditional detection probabilities involved in these two periods affect the probability distribution of having different numbers of detection reports generated within these two periods.

The conditional detection probabilities are much more complicated than the simple example shown in the previous paragraph, due to the following two aspects. First, let m_s denote the number of sensing periods a target takes to traverse a distance of $2R_s$, so $m_s = \lceil \frac{2R_s}{Vt} \rceil$. If we consider the detection probability in period i, where $i > m_s$, we can see that it depends on target detections in the previous m_s sensing periods. Note that m_s reflects the fact that the DR of the target in period i overlaps with the DRs of the target in periods $i-1$, $i-2$, ..., and $i-m_s$. Second, we may have multiple sensors in the DR of the target in a single period, especially when the deployment is not very sparse. In this case, we need to consider the probabilities of having different numbers of nodes in the overlapped DRs.

An intuitive approach, which we call the Temporal approach (T-approach), is that we compute the probabilities of having different numbers of detection reports based on the temporal sequencing: period by period. In each period from period 1 to period M, we compute the probabilities of having different numbers of detection reports generated in that period based on previous target detections. Then we use a Markov chain [7] to compute the probabilities of having different numbers of detection reports generated from period 1 to period M. However, simple analysis, which is not shown here due to the page limit, indicates that this method can easily get into the state explosion problem, in which the Markov chain needs to use millions or more states to deal with these conditional probabilities. The reason is that whether a target is detected or not by a sensor in a sensing period is temporally correlated: once the target is within a sensor's sensing range, the target may remain in that sensor's sensing range for up to $m_s + 1$ periods. When we compute the probabilities of having different numbers of detection reports in a sensing period, we need to keep track of the number of nodes in the overlapped DR of that period and the period before that period to resolve the temporally correlated detection dependency problem. This process seems simple, but it requires a huge number of states to achieve it, especially when m_s is large.

3.3 Spatial Approach

The failure of the T-approach indicates that the temporal correlation of detection dependency problem is hard to be resolved by considering the detection reports according to the temporal sequencing. In this paper, we conquer that problem by considering the detection reports spatially. We propose an approach called the Spatial approach (S-approach). In the S-approach, instead of considering the detection reports period by period, we compute the detection reports area by area. Note that in a random uniform distribution, the probability of having a node in any location is the same. The temporal correlation of detection dependency only affects the number of periods a sensor covers a target. Here when we say a target is covered by a sensor, it means that the target is within the sensor's sensing range.

In the S-approach, we use the Aggregate Region (ARegion) to denote the DRs from period 1 to period M and we want to compute the probabilities of having different numbers of reports generated from the ARegion from period 1 to period M. We divide the ARegion into multiple subareas based on the number of periods a sensor covers the target if the sensor is in a subarea. This is to account for the fact that sensors in different subareas may cover the target for different number of periods. We use Region(i) to denote

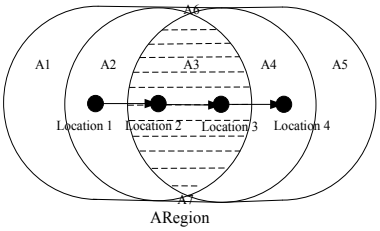


Figure 3. ARegion of a target in sensing period 1, 2 and 3 ($M = 3$).

all the subareas within the ARegion, in which if there is a sensor, the sensor covers the target for i sensing periods. In this section, we do not describe the way to compute the size of Region(i), since it involves details that will be explained in the following section. Rather, we use the example in Figure 3 to illustrate what Region(i) is comprised of. The ARegion in Figure 3 is divided into 7 subareas by the DRs of periods 1, 2 and 3. It is intuitive to see that sensors in different subareas may cover the target for different numbers of periods. For example, a sensor in the subarea A3 covers the target for 3 periods, because it is within all the three DRs. So, Region(1)=A1+A5+A6+A7, Region(2)=A2+A4, Region(3)=A3.

Note that if $M > m_s$, we have $m_s + 1$ regions. In the remainder of the paper, we only consider the general case in which $M > m_s$. Suppose we have computed the size of Region(i). Then we can compute the probabilities of having different numbers of detection reports generated from the ARegion from period 1 to period M. Note that, in the remainder of the paper, when we mention the detection reports, we mean the detection reports generated from period 1 to M.

The number of reports generated from the ARegion not only depends on the number of sensors in the ARegion, but also the specific sensor locations. Let $p_{S\{(m)(i_1,i_2,...,i_m)\}}$ denote the probability of having m sensors in the ARegion, which are distributed in Region(i_1), Region(i_2), ..., Region(i_m). $p_{S\{(m)(i_1,i_2,...,i_m)\}}$ is computed as follows:

$$p_{S\{(m)(i_1,i_2,\ldots,i_m)\}} = \left(\begin{array}{c} N \\ m \end{array} \right) (1 - \frac{ARegion}{S})^{N-m} \prod_{i=i_1}^{i_m} \frac{Region(i)}{S}$$

Note that the size of ARegion is $2MR_sVt + \pi R_s^2$. Since a sensor in a different region covers the target for a different number of periods, the probability of having a certain number of reports generated by a sensor in a different location is different. Let $p_{(m,i)}$ denote the probability for a sensor in Region(i) to generate m detection reports. It is computed as follows:

$$p_{(m,i)} = \left(\begin{array}{c} i \\ m \end{array} \right) P_d^m (1 - P_d)^{i-m} \quad (3)$$

Let $p_{s:m}$ denote the probability for sensors in the ARegion to generate m detection reports. To compute $p_{s:m}$, we first compute $p_{s:m:n}$, the probability of having m detection reports generated from the ARegion when there are n sensors in the ARegion. $p_{s:0:0}$ means the probability of having no sensor in the ARegion, and is computed as follows:

$$\begin{aligned} p_{s:0:0} &= \left(\begin{array}{c} N \\ 0 \end{array} \right) (\frac{ARegion}{S})^0 (1 - \frac{ARegion}{S})^N \\ &= (1 - \frac{ARegion}{S})^N \end{aligned} \quad (4)$$

In this paper, we use the pseudocode in Algorithm 1 to illustrate how to compute $p_{s:m:n}$. Due to the page limits, we only present the code for computing $p_{s:m:1}$ and $p_{s:m:2}$. A similar algorithm can be used to compute $p_{s:m:n}$ when $n > 2$.

Algorithm 1 Pseudocode for computing $p_{s:m:n}$

```
initialize p_{s:m:1} to zero {Compute the value of p_{s:m:1}.}
for R_1 = 1 to m_s + 1 do
    p_loc = p_{S{(1)(R_1)}} {Probability of having one node in Region(R_1)}
    for N_1 = 0 to R_1 do
        p_{s:N_1:1} += p_loc p_{(N_1,R_1)}
    end for
end for
initialize p_{s:m:2} to zero {Compute the value of p_{s:m:2}.}
for R_1 = 1 to m_s + 1 do
    for R_2 = 1 to m_s + 1 do
        p_loc = p_{S{(2)(R_1,R_2)}} {Probability of having two nodes: one in
        Region(R_1) and the other in Region(R_2)}
        for N_1 = 0 to R_1 do
            for N_2 = 0 to R_2 do
                p_{s:N_1+N_2:2} += p_loc p_{(N_1,R_1)} p_{(N_2,R_2)}
            end for
        end for
    end for
end for
```

After we obtain $p_{s:m:n}$, we can compute $p_{s:m}$ as follows: $p_{s:m} = \sum_{n=0}^{N} p_{s:m:n}$, in which N denotes the total number of nodes deployed in the sensor field. Since the time complexity for computing $p_{s:m}$ is $O(m_s^{2N})$, we need to limit the value of n to reduce the computation time and sacrifice analysis accuracy. Suppose the maximum value of n taken into the computation is G. Let η_S denotes the analysis accuracy when the maximum value of n is G, which is as follows:

$$\begin{aligned} \eta_S &= \frac{\sum_{n=0}^{G} \sum_{m=0}^{\lambda} p_{s:m:n}}{\sum_{n=0}^{N} \sum_{m=0}^{\lambda} p_{s:m:n}} \\ &= \sum_{i=0}^{G} \left(\begin{array}{c} N \\ i \end{array} \right) (\frac{ARegion}{S})^i (1 - \frac{ARegion}{S})^{N-i} \end{aligned} \quad (5)$$

In the above equation, λ denotes $n(m_s + 1)$, which is the maximum number of reports n sensors can generate, since a sensor can cover the target for at most $m_s + 1$ periods. The equation quantifies the relationship between G and η_S. If users define the required analysis accuracy, we can compute the smallest value of G based on that equation. However, to achieve enough accuracy like 95%, the value of G

is always equal to or greater than 5, especially when the size of the ARegion is large. This may result in high computation overhead. For example, if m_s is 10 and G is 6 based on the user requirements for analysis accuracy, which are reasonably small values, the time complexity would be in the order of 10^{12}. In this case, we need to wait at least many days to get the results. This is not acceptable.

3.4 Markov Chain based Spatial Approach

Although the S-approach correctly resolves the problem, it is restricted because of its computation complexity. In this section, we propose a Markov Chain based spatial approach to significantly reduce the computation complexity.

3.4.1 Main Idea

In the Markov Chain based S-approach (M-S-approach), instead of considering the DRs of period 1 to period M as a single ARegion, we divide the ARegion into multiple subARegions. Each subARegion is further divided into multiple areas based on the number of periods a sensor covers the target if the sensor is in the area. In each step, we count the detection reports generated from a subARegion by using the same method shown in the S-approach. After all the subARegions are processed, we use a Markov chain to assemble these probabilities and generate the final results.

While the ARegion can be divided in many different ways, we divide it based on the NEDR of each period. In this way, the ARegion is divided into M subARegions, each of which is the NEDR of one period. In the M-S-approach, we start from the NEDR of period 1 to the NEDR of period M. That means in the ith step, we compute the probabilities of having different numbers of detection reports generated from the NEDR of the ith sensing period. Based on that, we compute the probabilities of having different numbers of detection reports generated from period 1 to period M based on a Markov chain.

In this way, in each sensing period, we only need to consider the detection reports generated from the NEDR of that period. Compared to the S-approach, since the size of the NEDR that we consider in each sensing period is significantly smaller than the size of the ARegion, we can use a much smaller G value when we compute the probabilities of having different numbers of detection reports generated in that area, given the same user requirements for analysis accuracy. Therefore, we reduce the computation complexity of the S-approach.

In the M-S-approach, we split our modeling into three stages: Head stage, Body stage and Tail stage. The Head stage deals with the NEDR of period 1. The Body stage processes the NEDRs from period 2 to period $M - m_s$ and

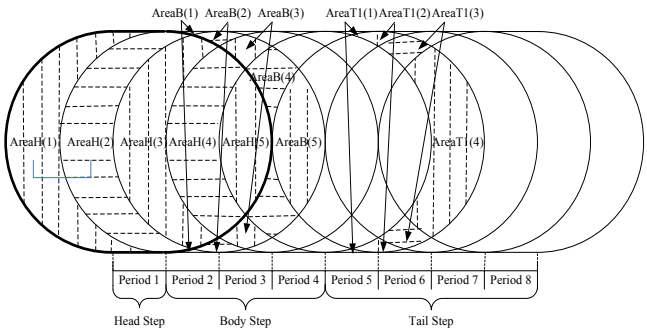


Figure 4. An example of target detections in multiple sensing periods ($M = 8$, $m_s = 4$).

the Tail stage considers the NEDRs from period $M-m_s+1$ to M. While all three stages use the same set of states in the Markov chain, they have different transition matrices. Figure 4 shows an example of target detections in multiple sensing periods. In this example, $M = 8$ and $m_s = 4$. So the Head state contains the NEDR of period 1, the Body stage contains the NEDRs of periods 2, 3 and 4, and the Tail stage includes the NEDRs of periods 5, 6, 7 and 8.

3.4.2 Head Stage

The reason to separate the Head stage from the Body and the Tail stages is that in the Head stage, the NEDR equals the whole DR, while the NEDRs in the following periods are only parts of the DRs. In the Head stage, we compute the probabilities of having different numbers of detection reports generated from the NEDR of period 1. We first divide the DR of period 1 into $m_s + 1$ subareas according to the number of sensing periods a sensor covers the target if a sensor is in the subarea. These subareas are named as AreaH(1), AreaH(2), ..., and AreaH($m_s + 1$), respectively, and are divided by the DRs of the following m_s sensing periods. More formally, AreaH(i) is the area that belongs to the DRs of both period 1 and period i, but not the DR of period $i + 1$. After dividing the DR into these subareas, we can see that if there is a sensor in AreaH(i), the target is covered by the sensor for i sensing periods. This can be easily proved according to the definition of AreaH(i). Figure 4 also shows an example of the subareas in the DR of period 1, where the DR of period 1 is highlighted with the bold line. As shown in this example, $m_s = 4$ and we have 5 subareas.

We use $p_{h:m:n}$ to denote the probability of having m detection reports generated from the NEDR of period 1, when there are n sensors in the NEDR of period 1. With preliminary geometry knowledge, it is easy to compute the size of AreaH(i) as shown in Equation (6). After we obtain the sizes of all the subareas in the DR of period 1, the same method in the S-approach to compute $p_{s:m:n}$ is used

$$AreaH(i) = \begin{cases} 2R_sVt & \text{if } i = 1 \\ \pi R_s^2 - (2R_s^2 arccos(\frac{(i-1)Vt}{2R_s}) - (i-1)Vt\sqrt{R_s^2 - (\frac{(i-1)Vt}{2})^2}) - \sum_{m=2}^{i-1} AreaH(m) & \text{if } 1 < i < m_s + 1 \\ 2R_s^2 arccos(\frac{(i-2)Vt}{2R_s}) - (i-2)Vt\sqrt{R_s^2 - (\frac{(i-2)Vt}{2})^2} & \text{if } i = m_s + 1 \end{cases} \quad (6)$$

to compute $p_{h:m:n}$. Here we do not go through the same process again. Rather, we point out the changes to be made to the equations in the S-approach: the size of the ARegion ($2MR_sVt + \pi R_s^2$) should be replaced with the size of the NEDR of period 1 ($2R_sVt + \pi R_s^2$); Region(i) should be replaced with AreaH(i).

Let $p_{h:m}$ denote the probability for the sensors in the NEDR of period 1 to generate m detection reports, in which $p_{h:m} = \sum_{n=0}^{N} p_{h:m:n}$. Therefore, we also need to limit the value of n to reduce the computation time. Suppose the maximum value of n that is taken into the computation in the Head stage is g_h. In this case, the analysis accuracy in the Head stage, which is denoted by ξ_h, is computed as follows:

$$\xi_h = \sum_{i=0}^{g_h} \left(\begin{array}{c} N \\ i \end{array} \right) (\frac{2R_sVt + \pi R_s^2}{S})^i (1 - \frac{2R_sVt + \pi R_s^2}{S})^{N-i} \quad (7)$$

Based on $p_{h:m}$, we can use the Markov chain shown in Figure 5 to model the target detections in period 1.

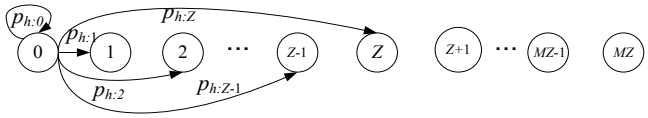


Figure 5. Markov chain used in the Head stage.

In Figure 5, Z denotes $(m_s + 1)g_h$, which is the maximum number of detection reports that can be generated from the DR of period 1, given that we only consider at most g_h sensors in the NEDR of period 1. State n denotes the state that has n detection reports generated in total. We use $MZ + 1$ states to denote all possible numbers of detection reports, since MZ is the maximum number of detection reports that can be generated during M sensing periods. If we are only interested in the probability of having at least k detection reports, we can merge the states from k to MZ. In this case, the transition probability to this merged state is the sum of all the transition probabilities to the states from state k to MZ. In Figure 5, an arrow is used when the transition probability between two states is greater than zero.

We use $\mathbf{T_H}$ to denote the transition matrix in the Head stage, which is easy to obtain based on the Markov chain shown in Figure 5 and thus not shown here. The Markov chain in the Head stage is only executed for one iteration since the Head stage only considers the NEDR of period 1.

3.4.3 Body Stage

The way to model the Body stage is similar to that used to model the Head stage. Yet, there are two major differences: first, the NEDR of a period in the Body stage is only part of the DR of that period and looks like a crescent; second, the Body stage contains $M - m_s - 1$ periods and thus has $M - m_s - 1$ steps, each of which deals with the NEDR of the corresponding period in the Body stage.

In each step of the Body stage, we divide the NEDR of the corresponding period into $m_s + 1$ subareas according to the number of periods a sensor covers the target if we put a sensor in the subarea. These subareas are named as AreaB(1), AreaB(2), ..., and AreaB($m_s + 1$), respectively. These subareas are divided by the DRs of the following m_s sensing periods. AreaB(i) in period l is the area that belongs to the NEDR of period l and the DR of period $l + i - 1$, but does not belong to the DR of period $l + i$. It is also easy to prove that a sensor in AreaB(i) will cover the target for i periods based on the definition of AreaB(i). Figure 4 also gives an example of these subareas in the NEDR of period 2. Since $m_s = 4$, the NEDR of period 2 is divided into 5 subareas, as shown in the figure.

We use $p_{b:m:n}$ to denote the probability of having m detection reports generated from the NEDR of a sensing period in the Body stage, when there are n sensors in the NEDR of that period. Based on AreaH(i) computed in the Head stage, we can compute the sizes of the subareas in the NEDR of a sensing period in the Body stage as follows:

$$AreaB(i) = \begin{cases} AreaH(i) - AreaH(i+1) & \text{if } i \leq m_s \\ AreaH(i) & \text{if } i = m_s + 1 \end{cases} \quad (8)$$

After obtaining the sizes of all the subareas, the same method in the S-approach for computing $p_{s:m:n}$ is used to compute $p_{b:m:n}$. We only need to make two changes: the size of the ARegion ($2MR_sVt + \pi R_s^2$) should be replaced with the size of the NEDR of a period in the Body stage ($2R_sVt$), and Region(i) should be replaced with AreaB(i).

Let $p_{b:m}$ denote the probability for the sensors in the NEDR of a sensing period in the Body stage to have m detection reports. Suppose the maximum value of n that is taken into the computation in each sensing period in the Body stage is g. In this case, the analysis accuracy in each

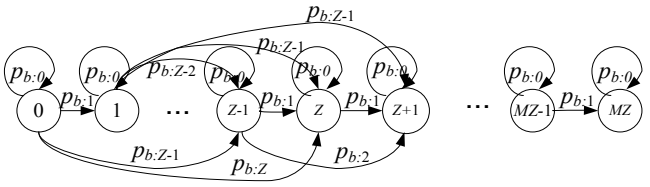


Figure 6. Markov chain used in the Body stage.

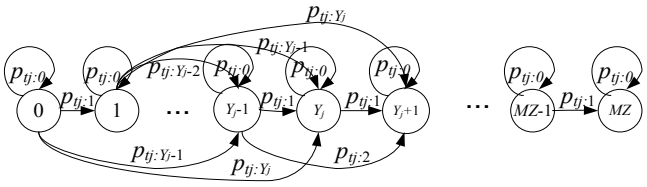


Figure 7. Markov chain used in period T_j in the Tail stage.

step of the Body stage, which is denoted by ξ, is computed as follows:

$$\xi = \sum_{i=0}^{g} \left(\begin{array}{c} N \\ i \end{array} \right) (\frac{2R_sVt}{S})^i (1 - \frac{2R_sVt}{S})^{N-i} \tag{9}$$

Based on $p_{b:m}$, the Markov chain used in each step of the Body stage is shown in Figure 6. We use $\mathbf{T_B}$ to denote the transition matrix in each step of the Body stage, which is not shown here. Note that the Body stage contains $M - m_s - 1$ steps and the Markov chain in each step is the same. Therefore, the Markov chain in the Body stage will be executed for $M - m_s - 1$ iterations.

3.4.4 Tail Stage

The size of the NEDR of a sensing period in the Tail stage is the same as that in the Body stage. However, the NEDRs in the Tail stage are divided into different numbers of subareas in different sensing periods, because the NEDRs of different sensing periods in the Tail stage overlap with different numbers of DRs of the following periods before period $M + 1$.

The Tail stage includes m_s sensing periods, ranging from period $M - m_s + 1$ to period M. So, the Tail stage has m_s steps. For ease of description, we rename period $M-m_s+1$ to period M as period T_1 to T_{m_s}, respectively. The NEDR of period T_j is divided into $m_s + 1 - j$ subareas, which are called AreaT$_j$(1), AreaT$_j$(2), ..., AreaT$_j$($m_s + 1 - j$), respectively. These subareas are divided by the DRs of the following $m_s - j$ sensing periods. AreaT$_j$(i) is the area that belongs to the NEDR of period T_j and the DR of period T_{i+j-1}, but does not belong to the DR of period T_{i+j} if $T_{i+j} \leq T_{m_s}$. We can also prove that a sensor in AreaT$_j$(i) will cover the target for i sensing periods before the end of period M based on the definition of AreaT$_j$(i). Figure 4 also gives an example of these subareas in the NEDR of period T_1. Because $m_s = 4$, the NEDR of period T_1 is divided into 4 subareas, as shown in the figure.

We use $p_{tj:m:n}$ to denote the probability of having m detection reports generated from the NEDR of period T_j in the Tail stage, when there are n sensors in the NEDR of period T_j. Based on AreaB(i) computed in the Body stage, the size of AreaT$_j$(i) in the NEDR of period T_j in the Tail stage is computed as follows:

$$AreaT_j(i) = \begin{cases} AreaB(i) & \text{if } i \leq m_s - j \\ \sum_{m=m_s+1-j}^{m_s+1} AreaB(m) & \text{if } i = m_s + 1 - j \end{cases} \tag{10}$$

Then we can use the same method in the S-approach to compute $p_{s:m:n}$ to compute $p_{tj:m:n}$. In the Tail stage, we need to make three changes when we compute $p_{s:m:n}$: 1) the size of the ARegion ($2MR_sVt + \pi R_s^2$) should be replaced with the size of the NEDR of a period in the Tail stage ($2R_sVt$); 2) Region(i) should be replaced with AreaT$_j$(i); 3) since the NEDR of period T_j is divided into $m_s + 1 - j$ subareas, $m_s + 1$ in the pseudocode should be replaced with $m_s + 1 - j$.

Let $p_{tj:m}$ denote the probability for the sensors in the NEDR of period T_j in the Tail stage to have m detection reports. Since the size of the NEDR of each period in the Tail stage is the same as that in the Body stage, the same maximum value of n is used in the Tail stage, which is g. Therefore, the analysis accuracy in each step in the Tail stage is the same as that in the Body stage, which is ξ.

Based on $p_{tj:m}$, the Markov chain used in period T_j in the Tail stage is shown in Figure 7. In Figure 7, we use Y_j to denote the value of $(m_s+1-j)g$. We use $\mathbf{T_{Tj}}$ to denote the transition matrix in sensing period T_j in the Tail stage, which is not shown here. Note that the Tail stage includes m_s steps and the transition matrices used in different steps are different.

After obtaining all the transition matrices used in different stages, we can compute the probabilities of having different numbers of detection reports in M sensing periods. We use $\mathbf{u}$ to denote the initial probability vector, which has $MZ + 1$ elements. Since we have zero reports before the target enters the field, $\mathbf{u}$ is expressed as follows:

$$\mathbf{u} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \end{bmatrix} \tag{11}$$

Let $\mathbf{Result}$ denote the probability distribution of the Markov chain after M sensing periods. We have

$$\mathbf{Result} = \mathbf{u}\mathbf{T_H}\mathbf{T_B}^{M-m_s-1} \prod_{j=1}^{m_s} \mathbf{T_{Tj}} \tag{12}$$

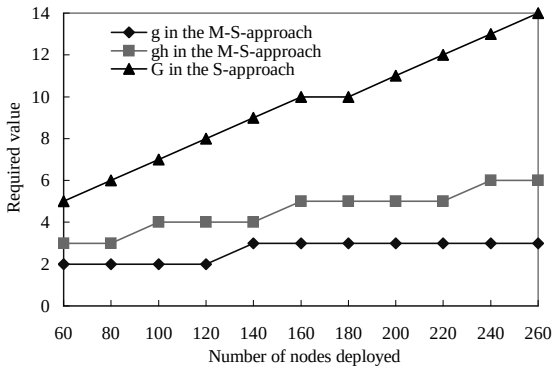


Figure 8. The required value of g, g_h and G to satisfy 99% analysis accuracy.

If the index of the first element in $\mathbf{Result}$ is one, $\mathbf{Result}[m]$ denotes the probability of having $m-1$ detection reports in M sensing periods. Let $sum = \sum_{m=1}^{MZ+1} \mathbf{Result}[m]$. Since both g_h and g are smaller than N, $sum < 1$. So, when we compute the detection probability of a target, we normalize the result as follows:

$$P_M[X \geq k] = \sum_{m=k}^{MZ+1} \mathbf{Result}[m+1] \times \frac{1}{sum} \tag{13}$$

3.4.5 Time Complexity

Based on g_h and g used in the M-S-approach, the analysis accuracy of the M-S-approach, which is denoted by η_{MS}, is computed as follows:

$$\eta_{MS} = \xi_h \xi^{M-1} \tag{14}$$

Based on the above equation, we can also compute the required value for g_h and g in the M-S-approach given the user requirements for analysis accuracy. Suppose the analysis accuracy requirement is η_R. To meet this analysis accuracy, the following condition should be satisfied in the M-S-approach: $\xi_h \xi^{M-1} \geq \eta_R$. Let $\xi_h = \xi$ for simplicity. We have $\xi \geq \eta_R^{\frac{1}{M}}$. Based on Equation (7) and Equation (9), we can compute the value of g_h and g. Based on Equation (6), the value of G used in the S-approach can also be obtained by setting $\eta_S \geq \eta_R$.

Since the time complexity of the S-approach is $O(m_s^{2G})$ and the time complexity of the M-S-approach is $O(m_s^{2g_h}(M-1)m_s^{2g})$, the value of G used in the S-approach and the values of g_h and g used in the M-S-approach are the key factors in determining the computation time. So we compare g and g_h in the M-S-approach, and G in the S-approach to reveal the computation time of both approaches. Figure 8 shows the required value for g, g_h and G to achieve 99% analysis accuracy. The results obtained in the figure are based on the following parameter settings: $S = 32000m \times 32000m$; $R_s = 1000m$; $t = 1min$; $M = 20$; $V = 10m/s$. We can see that G is significantly greater than both g and g_h. When G is large, such as 6 or more, the S-approach may be computationally infeasible depending on the value of m_s. From our experiments, it is common for the S-approach to run for many days to get the results. Several times, we are forced to turn off the program since it shows no sign of completing the computation in the near future. On the contrary, the M-S-approach we propose can significantly reduce the execution time of the analysis from many days to 1 minute.

Figure 8 also shows that g_h is greater than g, because the size of the NEDR of period 1 considered in the Head stage is greater than the size of the NEDR of any period in the Body and Tail stages. In fact, we can reduce the computation overhead in any step by further dividing the computation in that step into multiple substeps and each substep only considers the detection reports generated from a subarea of that step. In this case, g_h or g used in the M-S-approach will be smaller. We do not go into the details of how to further divide the computation in each step into multiple substeps due to page limits. Since both g and g_h are significantly smaller than G, we convert a computationally infeasible solution into a quick solution without losing any accuracy. All our analysis results, when g_h and g are 3, are obtained within one minute.

4 Model validation

We have implemented the group based event detection simulator in Matlab. We compare the detection probability of a target obtained from the analysis to that obtained from simulations to validate our model.

Simulation Configuration: Unless otherwise specified, in all our analysis and simulations, we use the parameter settings suggested by researchers at the Office of Naval Research[3]: 60∼240 sensor nodes are randomly deployed in a physical region of $32000m \times 32000m$; the sensing range of each node is $1000m$ and the communication range is $6000m$; if a target is within a sensor's sensing range, the probability for the sensor to detect the target is 90%; for the group detection algorithm that we abstract, the system level detection decision is made when a sequence of at least 5 detection reports are recorded within 20 sensing periods (the time length of each sensing period is 1 minute); otherwise, detection reports are perceived as false alarms. In the simulation, a target moves from a randomly chosen starting location towards a randomly chosen direction, with the specified speed of 4m/s or 10m/s.

[3] A long sensing range (around $1km$), and a long communication range (around $10km$) with a reasonable data rate (5∼10kHz) in undersea acoustic communication is achievable [10] for supporting of sparse deployments. The short range Mica2/Telos motes are comparatively less appropriate.

For a sensor node to report detection information back to a base state through multi-hop networking, the maximum possible physical distance in this deployment is around $36km$, that is, around 6 hops. With classic Geographic Forwarding routing protocols like GF [11] and GPSR [12], this 6-hop end-to-end communication can be easily finished within a single sensing period, that is, 1 minute. As long as a sensor can send a packet to the base station through multi-hop networking within a single sensing period time (1 minute here), no matter what MAC and Routing combination is used to implement the underlying multi-hop networking, our group detection performance analysis in this paper is still valid. For this reason, we ignore the communication stack in this simulation and only focus on sensing. For each trial of simulation, we randomly generate all nodes' locations and also randomly choose the staring location and moving direction of the target. For each sensing period, we compute the geographical region the moving target passes and compare that with the locations of all sensor nodes. Therefore, we know which sensor can detect the target at what time, and know how many detections are reported. When there are at least 5 detection reports within 20 sensing period, we consider that the target is detected. The simulation is repeated 10000 times, and the detection probability is computed by dividing the number of trials that record at least 5 reports to the total trial of 10000.

Figure 9(a) shows that the analytical model is extremely accurate under various simulation settings when the target travels in a straight line. The analytical results coincide with the simulation results. This means that despite the complicated conditional probabilities involved in the problem, our method successfully models the target detections in multiple sensing periods. Figure 9(a) also reveals that the detection probability increases when more sensor nodes are deployed.

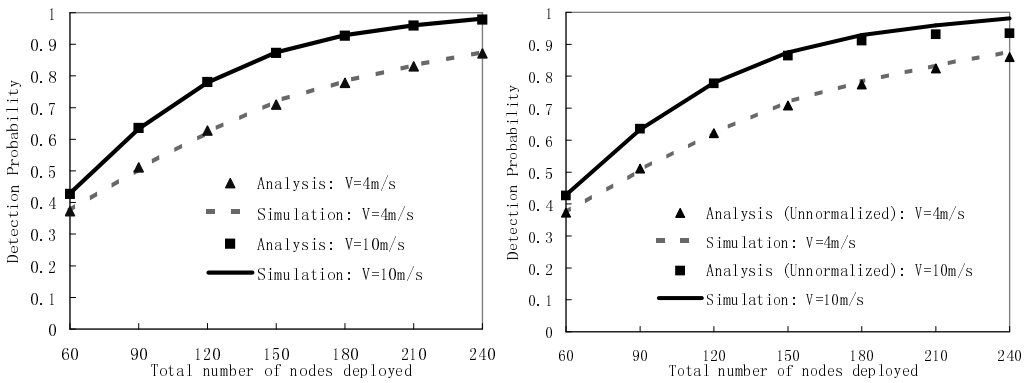
One interesting thing to note is that according to Equation (14), when $N = 240$ and $V = 10m/s$, the analysis accuracy of the M-S-approach is 95.6% when both g_h and g are set to 3. However, Figure 9(a) shows that the analysis accuracy is above 99% when $N = 240$ and $V = 10m/s$. The reason is that when we compute the detection probability in Equation (13), we normalize the result by multiplying $\frac{1}{sum}$. The normalization helps improve analysis accuracy. The reason is that when the analysis accuracy is 95.6% according to Equation (14), it means that in total 4.4% of the probability distribution of having different numbers of reports is not taken into consideration. When the normalization is applied, we assume that the probability distribution of having different number of reports when we only consider limited number of nodes (95.6% probability), $\leq g_h$ in the Head stage and $\leq g$ in both the Body and Tail stages, is the same as that when we consider all the N nodes (4.4% probability).

Another interesting thing in Figure 9(a) is that when the moving target's velocity is 10m/s the detection probability is higher than that when the moving velocity is 4m/s. This is actually one advantage of using group based detection in sparse sensor networks. In a sparse sensor network, within a specified time period, the faster the target moves, the more covered sensing area the target travels through, hence the more detection reports the base station collects within 20 sensing periods, hence a higher chance the base station recognizes the target, that is, collecting 5+ detection reports within 20 sensing periods.

Different from Figure 9(a), Figure 9(b) shows the detection probability of the target when the result is not normalized in the analysis. When the result is not normalized, the analysis accuracy decreases as V or N increases, which conforms to Equation (14). Also, without normalization, the analysis error is above 4% when $N = 240$ and $V = 10m/s$, which is close to the value obtained from Equation (14). The analysis accuracy obtained from Equation (14) can be considered as the lower bound for the analysis accuracy when normalization is applied.

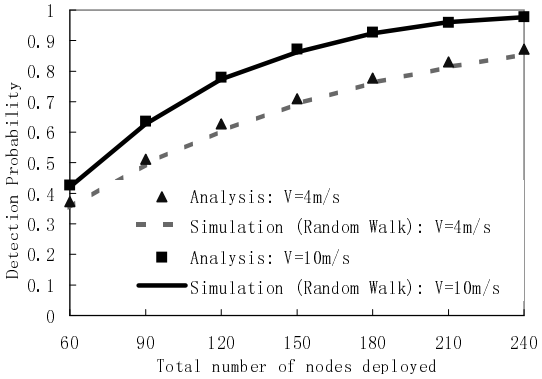
Performance When Target Does Not Move in a Straight Line: In the performance evaluation as shown in Figure 9 (a) and (b), we assume that the target travels in a straight line. In reality, the target may change its direction. If we do not know the target track and the target changes its direction randomly, we are unable to analyze its performance precisely. Nevertheless, the analysis results based on the straight line target track are very close to the simulation results even if the target changes its direction. We use Random Walk to denote the target travel pattern, in which the target randomly choose a new direction within [-$\pi/4$, $\pi/4$] of its current direction, every 1 minute. As shown in Figure 9(c), even though the target changes its direction, our analysis results are still very close to the simulation results, with the maximum error of 2.4%. When the target changes its direction, the probability of having less detection reports increases, because the ARegion is inclined to be smaller than that when the target travels in a straight line. For this reason, the detection probability obtained from our analysis, which is based on the straight target track, is expected to be higher than the actual detection probability when the target changes its direction.

It is worth pointing out that the analytical model can be easily extended if a group based detection algorithm is defined in a different way, such as: the system level detection decision is made when the sensor network generates a sequence of at least k detection reports from at least h nodes within M periods, which can be mapped to a possible target track. The main modification needed for the M-S-approach is to increase the number of states in the original Markov chain model from $MZ + 1$ to $hMZ + 1$. Except state 0, which is the same as that in the original model, each state



(a) Target moves in a straight line

(b) The detection probability is not normalized



(c) Target changes its direction

Figure 9. The results from our analytical model match the results from the simulations very well.

is labeled $m : n$, where $1 \leq m \leq MZ$ and $1 \leq n \leq h$. When $n < h$, state $m : n$ means that we have n nodes that generate m reports in total. When $n = h$, it means that we have at least h nodes that generate m reports.

5 Related Work

Research on performance analysis for wireless sensor networks has been one central topic in the sensor network community in recent years. Most of the detection performance analysis works [13] [14] [15] [16] are based on node scheduling and instantaneous detection. Node scheduling [17] [18] [19] [20] aims to minimize the energy consumption without severely sacrificing the system performance given that the nodes have been deployed. When node scheduling is applied, the field is only partially covered at any time. These analysis works are valuable in understanding the system performance, when the false alarm rate of each node is negligible. However, when the false alarm rate is high, many real systems [1] [2] use group based detection algorithms [6], rather than instantaneous detection, to filter out the false alarms. In this case, the analysis works based on instantaneous detection are no longer valid.

To the best of our knowledge, [9] [21] [22] are the only works that focus on group based detection. [22] estimates the influence fields of the target for classification and tracking. [9] assumes static sensors and mobile target, and analyzes the detection probability. In [21], the authors assume mobile sensors and static target, and focus on the detection latency. However, in all the analysis, the group based detection decision is made based on the detection reports from a single sensing period. In reality, the system level detection decision can be made based on detection reports from multiple sensing periods, rather than a single sensing period.

6 Conclusions and Future Work

In this paper, we focus on analyzing the detection performance of group based detection in sparse sensor networks. Sparse deployment is a key factor to reduce the total cost of a system when it needs to cover a vast area. To filter out the unavoidable false alarms generated by a single sensor in

a network, real sensor network systems tend to use group based detection to reduce the system level false alarms. In this paper, we successfully model the group based detection in sparse sensor networks based on a Markov chain, which is extremely accurate. The analytical model adopts the novel M-S-approach, which resolves the significant complexities involved in the conditional detection probability of a target in each sensing period. The analysis helps a system designer understand the impact of various system parameters in an easy way, without running extensive simulations or deploying real systems, which are costly.

In the future, we plan to study how to obtain the exact lower bound of k based on a specified false alarm model. This exact lower bound can provide statistical guarantee that no possible sequencing of false alarms result in a system level false alarm. We also plan to relax the assumption to address the case when the target travels in varying speeds.

References

[1] T. He, S. Krishnamurthy, J.A. Stankovic, T. Abdelzaher, L. Luo, R. Stoleru, T. Yan, L. Gu, J. Hui, and B. Krogh, "An Energy-Efficient Surveillance System Using Wireless Sensor Networks," in *ACM Mobisys*, 2004.

[2] L. Gu, D. Jia, P. Vicaire, T. Yan, L. Luo, A. Tirumala, Q. Cao, T. He, J. Stankovic, T. Abdelzaher, and B. Krogh, "Lightweight Detection and Classification for Wireless Sensor Networks in Realistic Environments," in *ACM SenSys*, 2005.

[3] "XBOW MICA2 and MICAZ Mote Specifications," http://www.xbow.com.

[4] J. Polastre, R. Szewczyk, and D. Culler, "Telos: Enabling Ultra-Low Power Wireless Research," in *ACM/IEEE IPSN/SPOTS 2005*, April 2005.

[5] "Integrated USBL Acoustic Tracking and Communication Systems," http://www.link-quest.com/html/intro2.htm.

[6] T. Abdelzaher, B. Blum, Q. Cao, Y. Chen, D. Evans, J. George, S. George, L. Gu, T. He, S. Krishnamurthy, L. Luo, S. Son, J. Stankovic, R. Stoleru, and A. Wood, "EnviroTrack: Towards an Environmental Computing Paradigm for Distributed Sensor Networks," in *IEEE ICDCS*, 2004.

[7] C.M. Grinstead and J.L. Snell, "Introduction to Probability," *American Mathematical Society*, 1997.

[8] Z. Yang, M. Li, and Y. Liu, "Sea Depth Measurement with Restricted Floating Sensors," in *IEEE RTSS*, 2007.

[9] T.A. Wettergren, "Performance of Search via Track-Before-Detect for Distributed Sensor Networks," in *submission to IEEE Transactions on Aerospace and Electronic Systems*.

[10] I.F. Akyildiz, D. Pompili, and T. Melodia, "Challenges for Efficient Communication in Underwater Acoustic Sensor Networks," in *ACM SIGBED Review*, 2004.

[11] B. Karp, *Geographic Routing for Wireless Networks*, Ph.D. thesis, Harvard University, Cambridge, MA, 2000.

[12] B. Karp and H. T. Kung, "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks," in *ACM MobiCom*, 2000.

[13] S. Ren, Q. Li, H.N. Wang, X. Chen, and X.D. Zhang, "Probabilistic Coverage for Object Tracking in Sensor Networks," in *ACM MobiCom (Poster)*, 2003.

[14] C. Gui and P. Mohapatra, "Power Conservation and Quality of Surveillance in Target Tracking Sensor Networks," in *ACM MobiCom*, 2004.

[15] C. Hsin and M.Y. Liu, "Network Coverage Using Low Duty-Cycled Sensors: Random and Coordinated Sleep Algorithms," in *IEEE/ACM IPSN*, 2004.

[16] Q. Cao, T. Yan, J.A. Stankovic, and T. Abdelzaher, "Analysis of Target Detection Performance for Wireless Sensor Networks," in *IEEE/ACM DCOSS*, 2005.

[17] T. Yan, T. He, and J.A. Stankovic, "Differentiated Surveillance for Sensor Networks," in *ACM SenSys*, 2003.

[18] X.R. Wang, G.L. Xing, Y.F. Zhang, C.Y. Lu, R. Pless, and C. Gill, "Integrated Coverage and Connectivity Configuration in Wireless Sensor Networks," in *ACM SenSys*, 2003.

[19] Q. Cao, T. F. Abdelzaher, T. He, and J. A. Stankovic, "Towards Optimal Sleep Scheduling in Sensor Networks for Rare Event Detection," in *IEEE/ACM IPSN*, 2005.

[20] L. Wang and S. S. Kulkarni, "Sacrificing a Little Coverage Can Substantially Increase Network Lifetime," in *IEEE SECON*, 2006.

[21] T. Chin, P. Ramanathan, and K.K. Saluja, "Analytic Modeling of Detection Latency in Mobile Sensor Networks," in *IEEE/ACM IPSN*, 2006.

[22] S. Bapat, V. Kulathumani, and A. Arora, "Reliable Estimation of Influence Fields for Classification and Tracking in Unreliable Sensor Networks," in *IEEE SRDS*, 2005.