ASFALT: A Simple Fault-Tolerant Signature-based Localization Technique for Emergency Sensor Networks

Murtuza Jadliwala, Shambhu Upadhyaya and Manik Taneja
State University of New York at Buffalo
Department of Computer Science and Engineering
201 Bell Hall, Buffalo, NY 14260, USA.
{msj3, shambhu, mtaneja}@cse.buffalo.edu

Abstract

We consider the problem of robust node deployment and fault-tolerant localization in wireless sensor networks for emergency and first response applications. Signature-based localization algorithms are a popular choice for use in such applications due to the non-uniform nature of the sensor node deployment. But, random destruction/disablement of sensor nodes in such networks adversely affects the deployment strategy as well as the accuracy of the corresponding signature-based localization algorithm. In this paper, we first model the phenomenon of sensor node destruction as a non-homogeneous Poisson process and derive a robust and efficient strategy for sensor node deployment based on this model. Next, we outline a protocol, called Group Selection Protocol, that complements current signature-based algorithms by reducing localization errors even when some nodes in a group are destroyed. Finally, we propose a novel yet simple localization technique, ASFALT, that improves the efficiency of the localization process by combining the simplicity of range-based schemes with the robustness of signature-based ones. Simulation experiments are conducted to verify the performance of the proposed algorithms.

1 Introduction

Wireless sensor networks (WSN) are being widely used in emergency monitoring and first response applications like natural calamities (storm, hurricanes), forest fires, terrorist attacks, etc. [14, 21, 18, 11]. Such networks are often referred to as *Emergency Sensor Networks (ESN)* [9]. *Localization* is the problem of determining the position of each sensor node (mote) after being deployed at an area of interest. Localization is extremely important in WSNs as the information collected by the sensor nodes is of very little use unless it is associated with the location of occurrence. Distributed localization protocols for WSNs can be divided into two broad categories namely *Beacon-based methods* and *Signature-based methods*. Beacon-based methods [20, 1, 15, 3, 7] require a few special nodes called beacon nodes, which already know their absolute locations via GPS or manual configuration and are fitted with high power transmitters. Remaining nodes estimate their location by first computing distance/angle estimates to the beacon nodes, and then applying triangulation or multilateration to these distance estimates. Signature-based or beaconless schemes [5, 6, 2, 10], on the other hand, assume that nodes are distributed in a non-uniform fashion over the deployment area, and use this non-uniform distribution as a signature to compute location by observing node neighborhoods.

In this paper, we study the problem of localization from the point of view of ESNs. Sensor node deployment in emergency applications is highly localized for each point (over the emergency area) and the size of the node group at each point depends on the intensity of the monitored event at that point. Due to such a non-uniformity in node deployment, signature-based schemes are ideal for localization in ESNs. Moreover, such schemes eliminate the need for costly beacon nodes and GPS devices and thus the “single point of failure” problem. But, one problem with signature-based schemes is that they assume a fixed node distribution over the deployment area (throughout the period of the application) and thus their accuracy is affected by factors that change the existing node distribution. Nodes over a deployment area can be arbitrarily destroyed, disabled or displaced, thus changing the previously fixed node distribution. Signature-based schemes have to take such distribution changes into account before localization, else they will produce inaccurate results.

Here, we attempt to construct signature-based localization schemes that are robust against random node destruction/disablement. We focus on two main factors, namely, 1) the initial node distribution over the deployment area, and 2) random node disablement. To provide an efficient distribution of sensor nodes during an emergency, we need a well-planned deployment strategy that is not only robust against the vagaries of the emergency situation but also helps signature-based localization in a positive way. To achieve this, we outline an emergency level-based deployment strategy that efficiently distributes the sensor nodes over the emergency area by dividing the area into various emergency levels depending on the severity of the emergency at a point. The process of node destruction during an emergency can be modeled as a non-homogeneous Poisson process, and the deployment strategy employs this model to make deployment decisions. Next, to improve the fault-tolerance of existing signature-based localization approaches, we propose an improvement in the form of a Group Selection Protocol (GSP). According to this protocol, only healthy or viable groups of nodes are chosen for participation in the localization process. Although GSP provides improvement in accuracy, it does not simplify the complex localization mechanism of signature-based schemes. To overcome this, we introduce ASFALT, a simple, fault-tolerant localization scheme that combines the salient features of both beacon-based and signature-based scheme. ASFALT uses distance measurements to groups of nodes in its neighborhood and a simple averaging argument to compute location. Using experimental results, we show that the performance and localization accuracy of ASFALT are better than that of standard signature-based algorithms, e.g., [6], especially in situations of arbitrary disablement/destruction of nodes.

The rest of the paper is organized as follows: the next section presents the case study of a signature-based localization technique. Section 3 presents the emergency level-based deployment strategy for ESNs and the Group Selection Protocol (GSP). Section 4 describes ASFALT: our fault-tolerant localization technique. Section 5 presents the evaluation results and in Section 6 we review some earlier research efforts in this direction. Finally, we conclude and present some directions for future research in Section 7.

2 Case Study: A Signature-based (Beaconless) Scheme for Localization

In this section, we present the case study of a signature-based (beaconless) localization technique proposed by Fang et al. Interested readers may refer to the complete article [6] for details.

2.1 Deployment Model and Localization

This localization technique employs a group-based deployment strategy in which the entire deployment area is first divided into a grid of n points. Then, nodes are deployed in groups of equal sizes at each point on the grid. The final position of each node after deployment is assumed to follow some non-uniform distribution, e.g., Normal (Gaussian), with mean as the point of deployment. Thus, the average deployment distribution of any mote over the entire region, if there are n groups, is:

$$f_{overall}(x, y) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{2\pi\sigma^2}e^{-[(x-x_i)^2+(y-y_i)^2]/2\sigma^2}$$

The eventual goal is to get distance estimates from the target node at location $\theta(x, y)$ to each of the fixed point on the grid where nodes are deployed, so that $\theta(x, y)$ can be determined by multilateration. Let $a = (a_1, \ldots, a_n)$ be a vector representing the neighborhood observation of the target node, i.e., a_i number of nodes from group G_i are in the neighborhood of the target node. Given the number m_i of nodes deployed in each group G_i and the probability distribution function (p.d.f) of the deployment, the probability that a is observed by the target node at θ (where X_i is a random variable representing the number of nodes from G_i that are neighbors to the target node and all $X_i's$ are mutually independent) is,

$$f_n(a|\theta) = Pr(X_1 = a_1|\theta)\ldots Pr(X_n = a_n|\theta)$$

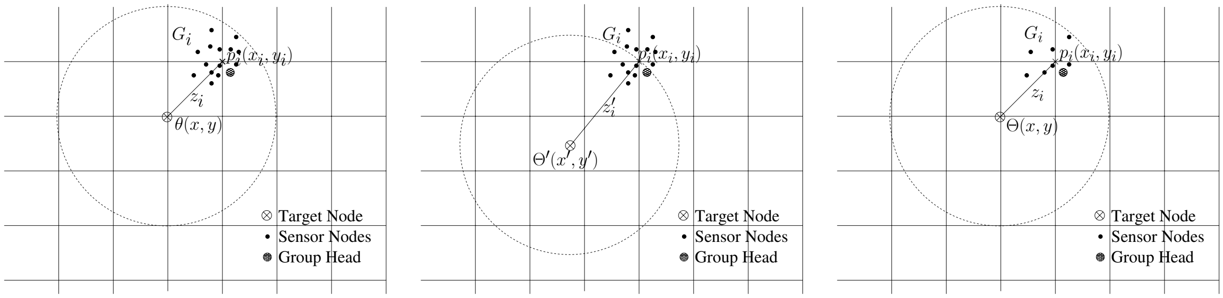
Let, $g_i(\theta)$ be the probability that a mote from group G_i can land within the neighborhood of the point θ. Then,

$$f_i = Pr(X_i = a_i|\theta) = \binom{m_i}{a_i}(g_i(\theta))^{a_i}(1-g_i(\theta))^{m_i-a_i}$$

Let z_i represent the distance from θ to the point where group G_i is deployed. It is clear that $g_i(z_i) = g_i(\theta)$. Using a maximum likelihood analysis it can be shown that the above likelihood function, f_i, is maximized when,

$$g_i(z_i) = \frac{a_i}{m_i}$$

Now, to compute the value of z_i from $g_i(z_i)$ ($z_i = g_i^{-1}(g_i(z_i))$), we need a formulation for $g_i(z_i)$. Fang et al. have used complex geometric techniques to formulate $g_i(z_i)$ (see [6]). As a result, $g_i(z_i)$, which is an extremely complex function, cannot be computed in an online fashion by the low power sensor nodes. To overcome this problem, a table-lookup approach is used to find z_i given a_i and m_i, i.e., $g_i(z_i)$ is pre-calculated (sampled) in an offline fashion for discrete values of z_i, and stored in the form of a table in the mote's memory. Once a_i and m_i are known, a sensor node can find the most likelihood value for z_i by looking up the value of $g_i(z_i)$ $\left(= \frac{a_i}{m_i}\right)$ from the table. Distances to at least three or more known points (z_i's) can then be used to compute $\theta(x, y)$ by atomic multilateration.



(a) (b) (c)

Figure 1. Effect of node destruction on the accuracy of signature-based localization approaches. (a) No nodes destroyed, Node in question at $\theta(x, y)$ and $|G_i| = m_i = a_i = 15$ (b) No nodes destroyed, Node in question at $\theta'(x', y')$ and $|G_i| = m_i = a_i = 8$ (c) 7 nodes destroyed, Node in question at $\theta(x, y)$, $|G_i| = m_i = 15$ and $a_i = 8$

2.2 Disadvantages

In ESNs, node distribution can change due to factors like node destruction/disablement, faulty nodes, etc. contrary to the static node distribution assumption in signature-based localization schemes. Figure 1 shows how random node destruction affects localization in signature-based schemes. Figure 1(a) is the base line scenario. In this case, the distance (z_i) between θ and the point of group deployment p_i can be computed correctly. But, the above signature-based method cannot distinguish between cases (b) and (c), i.e., when a node at θ actually observes just 8 nodes from group G_i it will compute the distance between θ and p_i as z_i' (as shown in Figure 1(b)). But, it may be the case that it just hears from 8 nodes from group G_i because the remaining 7 nodes might be disabled and the correct distance is still z_i and not z_i' (as shown in Figure 1(c)).

Table 1 shows an approximation of the function $g_i(z_i)$, discussed above, as a table of values. Assuming a group size (m_i) of 100, we can see from Table 1 that a difference of even a single observed node can cause an error of roughly $12m$ in distance estimation to the corresponding deployment point. To verify the inaccuracies introduced by such an approximation we conducted simulation experiments using the J-Sim [16] simulation environment for wireless sensor networks. In this experiment, we simulate the signature-based algorithm discussed above and observe the effects of random node disablement on the localization accuracy of the algorithm. The deployment area is a 600m × 600m square grid consisting of 9 points, each having 20 nodes distributed around it. In each run of the simulation, the final position of each node is sampled from a two dimensional Normal distribution ($\mu = 0$, $\sigma = 50$, $R = 200m$) and the transmission range is fixed at $200m$. In each run, k (k varies from 1 up to 15) nodes per group are destroyed in *every* group and the location of every node in a particular group is estimated using the signature-based scheme discussed above. The results of the experiment are outlined in the plot in Figure 2. Performance of the algorithm is measured as an average of the localization errors of all the nodes in that group. From the plot, we can observe that the average localization error increases as k increases. Another trend that we observe in this plot is that at high values of k, the localization inaccuracy increases less steadily. This shows that beyond a certain threshold, the disablement of nodes has little effect on increasing the localization error. Moreover, the average localization error in the case of zero node destruction (i.e., $k = 0$) is just under 30m, which is high. One reason for the low accuracy of this algorithm, even when $k = 0$, is because the complex continuous function $g_i(z_i)$ is approximated by a table of discrete values. Thus, to improve the accuracy and efficiency of signature-based schemes in emergency applications, we need to address two issues: 1) improve fault-tolerance against disabled nodes and 2) reduce complexity. Since the accuracy of signature-based schemes depends on the initial distribution of nodes,

Table 1. Table of $g_i(z_i)$ values, $R = 200$, $\sigma = 50$

z_i	$g_i(z_i) = \frac{a_i}{m_i}$
1.00	0.864611
2.00	0.864449
3.00	0.864178
⋮	⋮
14.00	0.854055
15.00	0.852486
⋮	⋮

we first need to formulate an efficient strategy for sensor node deployment in emergency applications. We address this problem in the following section.

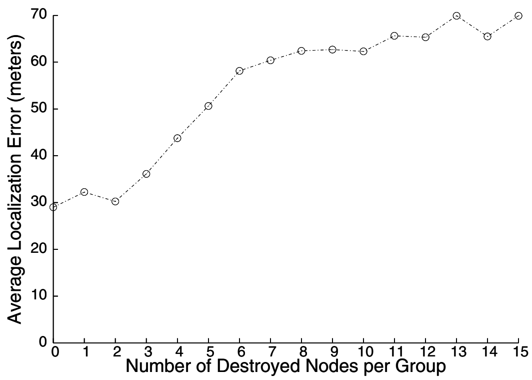


Figure 2. Plot of Localization vs. Number of Disabled Nodes

3 Node Deployment in Emergency Situations

Existing scattering-based (by airplane, fire truck, etc.) deployment strategies have several shortcomings for use in ESNs. First, deployment areas under severe conditions have high probability of node destruction as compared to areas under relatively tranquil conditions. Thus, deploying equal sized groups or in one big group uniformly over the entire area will not be very efficient in emergency situations. Points on the deployment area where the effect of the emergency is high require more nodes as compared to areas where the effect of the emergency is less hostile. But, just randomly deploying high number of nodes at points with greater emergencies is also not a good idea because the network may end up losing more nodes and the application may fail. Also, *manual* deployment is difficult due to the hostility, inaccessibility and unpredictability at the site of the emergency. Another problem is that current localization schemes do not incorporate any nodes or protocols to monitor changes in node distribution after deployment. Thus, a more rigid analysis is required before deploying nodes over the emergency area.

3.1 Model for Node Destruction

We model the phenomenon of node destruction at a point during an emergency as a *stochastic time process*, which is a process that can be described by a probability distribution with domain as the time interval of the process. In other words, it is a collection of random variables indexed by a set T (time). This helps to quantify the expected number of nodes that will be destroyed at a point during the emergency and the initial number of nodes that should be deployed at that point as a result. Assume that the emergency area is divided into a rectangular grid. Each dot in the grid represents a deployment point, say p_i.

Definition 3.1 *A deployment point is a point on the terrain where a node (or group of nodes) is planned to be deployed. The point where a node actually resides after deployment, not necessarily the same as the deployment point, is called the resident point.*

Let (x_i, y_i) be the coordinates of the point p_i. Assuming that there are k deployment points $p_1(x_1, y_1)$, $p_2(x_2, y_2)$, ..., $p_k(x_k, y_k)$, we have k groups of nodes, $G_1, G_2, \ldots, G_k$, where G_i is to be deployed at p_i. Since we are trying to model the effect of external factors on node survival, we assume that sensor nodes can be disabled only by external factors like fire, temperature, force, etc. and not by internal/self factors like battery failures, component malfunction, etc. Let, t_a be the start time of the application and t_b be the end time of the application. Thus, the entire period of the application is, $t_{a,b} = t_b - t_a$. Each deployment point, p_i, is associated with an *emergency level* based on the severity of the emergency condition at that point, as defined below.

Definition 3.2 *An emergency level λ_i at any instance for a deployment point i is defined as the average number of destroyed nodes in group G_i per unit time at that instance and the corresponding function $\lambda_i(t) : t \rightarrow \mathbb{N}$ is called the generalized emergency level function.*

In the above definition, a node is considered destroyed or disabled if it is not capable of communicating with any of the neighboring nodes. The probability of the number of disabled nodes in a group over a fixed period of time can be expressed as a Poisson distribution because these disablements occur with a known average rate (emergency level) during that interval and are independent of the time since the last node disablement. Specifically, the number of nodes disabled in a group during the period of the application can be modeled as a *non-homogeneous Poisson process*. This is because, the average rate of node disablement (emergency level) may change over time (between the start and the end of application) as the effect of the emergency at that point changes. Thus, the number of nodes disabled in a group G_i deployed over a deployment point p_i in the time interval $(a, b]$, given as $N_i(b) - N_i(a)$, is as shown in Eqn. (1),

$$P[(N_i(b) - N_i(a)) = k_i] = f(k_i, \lambda_i^{a,b}) = \frac{e^{-\lambda_i^{a,b}} (\lambda_i^{a,b})^{k_i}}{k_i!} \quad (1)$$

where $k_i = 0, 1, \ldots |G_i|$ and $\lambda_i^{a,b}$ is the overall emergency level for the deployment point i over the time interval $(a, b]$.

As mentioned before, an emergency level at a point cannot be assumed to be constant throughout the time interval $(a, b]$. Emergency level at a deployment point increases over time if the situation at that point worsens or it can decrease as the situation subsides. As a result, the overall emergency level $\lambda_i^{a,b}$ for the deployment point i can be defined in terms of the generalized emergency level function $\lambda_i(t)$ as shown below,

$$\lambda_i^{a,b} = \int_a^b \lambda_i(t)d(t).$$

3.2 Emergency Level-based Deployment

In this section, we describe a deployment strategy, called the *emergency level-based strategy*, that can be used to deploy sensor nodes in an emergency situation. The first issue that we need to address is how to assign an emergency level to each deployment point. An emergency scenario is an accumulation of various events occurring at various points. Each deployment point is associated with a sequence of events; each event produces a different rate of destruction. For example, a forest fire emergency consists of some areas that are directly under a wall of fire where the destruction rate is the highest. Some areas where the fire is out but are still under the effect of burning objects have lower rates of destruction. While others that are just under the influence of smoke might have a much lower rate of destruction. The best way to determine emergency levels for the various deployment points is by repeated controlled experiments. Before actual deployment, an emergency can be carried out in a controlled environment and the sequence of events at any deployment point i can be simulated for a fixed time, say the time of application $t_{a,b}$. A fixed large number of nodes, m_{max} (explained in Section 3.2.1), are deployed initially in groups for each point i and the number of destroyed nodes can be noted. Such experiments can be repeated n times and the number of destroyed nodes (k_i^j) is measured in each run j. Given a sample of n measured values of disabled nodes ($k_i^1, k_i^2 \ldots k_i^n$) for each deployment point i, we wish to estimate the value of the emergency level $\lambda_i^{a,b}$ of point i. Using a Maximum Likelihood Estimation (MLE) analysis, one can derive the most likely value of the emergency level for any deployment point i as shown in Eqn. (2).

$$\lambda_i^{MLE} = \frac{1}{n}\sum_{j=1}^{n} k_i^j \qquad (2)$$

Next, we focus on the group size or the total number of nodes to be deployed at each deployment point.

3.2.1 Determining Deployment Size

Definition 3.3 *The deployment size m_i for any deployment point i associated with an emergency level $\lambda_i^{a,b}$ is the number of sensor nodes to be deployed at that point.*

The deployment size m_i for a deployment point i depends on the emergency level $\lambda_i^{a,b}$ at that point and is determined as follows. The deployment size consists of two components. The first, called the standard deployment (m_i^s), is a fixed application specific constant that is same for every group. The next component, called varied deployment (m_i^v), is determined by the rate of node destruction at the deployment point and is proportional to the overall emergency level at the point i, i.e., $m_i^v \propto \lambda_i^{a,b}$. Thus, the deployment size m_i at a deployment point i is a combination of the standard deployment and the varied deployment components, i.e., $m_i = m_i^s + m_i^v$. Intuitively, more number of sensors are required at deployment points with higher emergency levels as compared to lower ones. According to our quantification of the deployment size, as the varied component of the deployment size is proportional to the emergency level it will make sure that areas with higher emergencies receive a larger deployment size. Moreover, the varied component m_i^v of the deployment size offsets the effects of node destruction at that point. Let m_{max} be an application dependent upper bound on the maximum number of nodes that can be deployed at any point that depends on factors like network density, cost of nodes, priority of coverage etc. Sensor nodes will be deployed at each deployment point in groups of size equal to the deployment size m_i if and only if $m_i \leq m_{max}$.

3.2.2 Hierarchical Deployment

Every group G_i consists of at least one node designated as the *group head* either prior to deployment or post-deployment through voting-based techniques. Group heads (or base stations) have been important components in the design of efficient monitoring applications right from the inception of wireless sensor networks. Due to the low computation power and storage capacity of sensor nodes, sensor network applications normally employ a record and forward paradigm [19]. In this paradigm, sensor nodes forward data to their respective group heads as soon as it develops, which then aggregates it and forwards it up the hierarchy. Because of such a hierarchical design, group heads are aware of all the active nodes in the group. Such a hierarchical design can be used in signature-based localization schemes to decide which groups have sufficient number of nodes to perform localization accurately. But, the group head in such a setting can also be a single point of failure. To overcome this problem, a group can appoint more than one group head depending on factors like size of group, distance between deployment points, type of application, etc. But, to elucidate the current exposition, we assume without loss of generality that each group consists of a single, always on (i.e., it is never disabled) group head.

We now summarize the deployment strategy:

1. Divide the deployment area into a fixed set of deployment points.

2. Assuming that there are k deployment points, assign an emergency level to each deployment point as discussed before. Then, prepare k groups of nodes, each of size determined by the corresponding emergency levels.

3. All of the above information like the group sizes, emergency levels, node distribution (discussed later) etc., called *predeployment information*, is loaded into the memory of every node before deployment.

4. Finally, deploy each group of nodes at the corresponding deployment point using non-manual techniques like aerial scattering, dispersion from a fire truck, etc.

3.3 Deployment Distribution

For a group of nodes thrown at a deployment point, the probability that the final position of a node from the group is at the deployment point is the highest and the probability decreases as we move away from the deployment point. As a result, the final position (resident point) of the nodes after deployment can be modeled as a continuous random variable with a certain fixed non-uniform p.d.f like Normal (Gaussian) distribution as shown in Eqn. (3). Moreover, random variates with unknown distributions are often assumed to be Normal (Gaussian), especially in physics and natural sciences, and thus we can assume that the node distribution around a deployment point is Normal. For a group G_i, the mean (μ) of the p.d.f is the corresponding deployment point $p_i(x_i, y_i)$. The standard deviation (σ) is application specific and depends on the coverage required around the deployment point.

$$f_i(x, y) = \frac{1}{\sqrt{2\pi}\sigma} e^{-[(x-x_i)^2+(y-y_i)^2]/2\sigma^2} \tag{3}$$

Equation (3) gives the probability that a node in the group G_i has a final position (x, y). Let $Pr_i(v)$ be the probability that a node v selected at random belongs to the group G_i. Then,

$$Pr_i(v) = \frac{m_i}{m_1 + m_2 + \ldots + m_k} \tag{4}$$

where $m_i, i = 1 \ldots k$ is the deployment size of the group G_i. Thus, the overall distribution of a randomly selected node v, i.e., the probability that the node v is present at the point (x, y) on the deployment area is:

$$f_{overall}(x, y) = \sum_{i=1}^{k} Pr_i(v) \times f_i(x, y) \tag{5}$$

Equation (5) represents the probability distribution of the final position of nodes just at the moment they are deployed. In theory, the probability that a randomly selected node lies closer to deployment points with higher emergency levels is high. But in practice, this may not be true as nodes in groups near higher emergency levels may also be destroyed with a higher probability and as a result the actual size of such groups may be fairly smaller than their original size at deployment. As discussed in Section 2.2, any scheme that uses this distribution should account for the loss of nodes in each group and use the most current group size. Next, we discuss a very simple and intuitive solution to the above problem, called the Group Selection Protocol (GSP). GSP, which is implemented on top of a signature-based localization algorithm, monitors changes in node distribution over the deployment area and helps to improve the accuracy of the resulting localization schemes.

3.4 Improving Signature-based Localization: Group Selection Protocol (GSP)

Let a_i be the number of nodes from group G_i that the target node at point $\theta(x, y)$ can hear from and let z_i be the distance from the target node to the deployment point of group G_i. The problem with the localization algorithm discussed in Section 2 is that in ESNs, not every observation a_i in $\{a_1, \ldots, a_n\}$ is correct or accurate. Groups where the node destruction rate is high might not be able to provide the correct value of a_i for localization. One way to overcome this problem is by being selective in choosing groups G_i's (and the corresponding observations a_i's) for the localization process. We use a_i's from only those groups that are *healthy*.

Definition 3.4 *The health of a group is quantified by the number of active nodes in the group. A node is active if it is able to communicate with at least one other node in the same group.*

In other words, only observations from those groups are used during localization in which the current health of the group is at least equal to the standard deployment size (m_i^s). This modification will reduce the number of z_i's (distances) available for localization. But, as long as we have at least 3 relatively accurate values of z_i's, localization can be done efficiently. Absence of at least 3 values for z_i will cause localization to fail, but due to the criticality of the applications in emergency situations sometimes no location is better than an incorrect value. In this protocol, group heads are used to monitor the health of their corresponding groups. After deployment, as the ad hoc network comes up, nodes begin sending initial setup information to their respective group heads. Using these communications from members of the group, the group head updates the health of its group.

At regular intervals, the group head broadcasts the current health of its group. These broadcasts are forwarded by all nodes up to a certain hop count so that even nodes farther away can know the health status of a particular group. The communication between nodes and the group head health broadcasts can be synchronized with the sleep-wake cycles of the nodes to save power. The group selection protocol is as outlined in Algorithm 1.

```
Observe the neighborhood, i.e., {a_1, a_2 ... a_k | a_i is the
  number of nodes in group G_i that are in radio range. }
Wait and observe health broadcasts (h_i) from the group
  heads. Update h_i to the latest value for each group.
for all groups G_i for which h_i is known do
  if The group is healthy, say (h_i >= m_i^s) then
    Compute g(z_i) = a_i/h_i.
    Compute z_i from g(z_i) by table look-up.
  end if
end for
if z_i corresponding to at least 3 distinct groups G_i is
  known then
  Compute θ(x, y) by multilateration (using z_i's and
    their corresponding p_i's)
else
  print "Cannot do Localization!"
end if
```

Algorithm 1: Group Selection Protocol (GSP)

Although the GSP proposes only minor and intuitive improvements to the process of signature-based localization, it performs better than existing algorithms in dynamic scenarios. We verify this claim using simulation experiments as outlined in Section 5. Simulation results show that GSP does improve the localization accuracy of signature-based algorithms when nodes over the deployment area are randomly disabled. Despite this improvement, there are some glaring problems with current signature-based approaches that are still left unaddressed by just employing the GSP. Current signature-based schemes are extremely complex involving hard to compute functions. Simplifying the process by using regression-based or table-based approximation techniques results in loss of accuracy in addition to issues like offline computation and storing the function as a table in the memory. The GSP provides some improvement in terms of accuracy relative to standard signature-based approaches, but does not improve on the complexity of such schemes. Moreover, GSP does not work well if node destruction is not localized to only some deployment points in the network. To overcome these problems, we propose a simple fault-tolerant signature-based localization approach called *ASFALT*.

4 ASFALT: A Simple Fault-tolerant Signature-based Localization Technique

In this approach, instead of just observing its neighborhood, the target node computes distances to every node in its neighborhood. The set of distance estimates from the target node to all nodes in a particular group is called the *distance vector* for that group. This distance vector is a sample from the two dimensional Normal distribution with mean as the distance between the target node and the deployment point of the group. Thus, given a distance vector, the distance from the target to a deployment point can be easily estimated by computing the mean of the sample.

4.1 Assumptions

We assume that nodes are deployed over the deployment area using an emergency level-based deployment strategy (Section 3). Also, any node is efficiently able to estimate its distance to its one hop neighbors using techniques like Received Signal Strength Indicator (RSSI), Time of Arrival (ToA), Time Difference of Arrival (TDoA), etc. [8]. Since currently we are not modeling any specific emergency, it is reasonable to assume that nodes are destroyed in a random fashion within a group. This is different from the *number* of nodes destroyed which is still a Poisson process and depends on the rate of destruction at that point. All the symbols and terminology used in this section are same as Section 3.

4.2 Localization Scheme

Let M be the target node for which localization has to be done and let $\theta(x, y)$ be the actual position of M. The ASFALT localization technique is outlined in Algorithm 2. Let z_i be the actual distance between $\theta(x, y)$ and the deployment point i. Let $d_i^1, d_i^2 \dots d_i^{m_i} | d_i^j \in \mathbb{R}$ be the distances of the nodes from the deployment point i ($d_i^j > 0$, if the position of node j is after i on the real line and $d_i^j < 0$ otherwise). Assuming that all m_i ($m_i^s + m_i^v$) nodes in G_i are in the radio range of M, let $z_i^1, z_i^2 \dots z_i^{m_i}$ be the distances of the nodes from M (distance vector). As mentioned before, the distances in the set $\{d_i^1, d_i^2 \dots d_i^{m_i}\}$ follow a Normal distribution and let $\tilde{d}$ be the random variable that takes values in this distribution. Thus,

$$E(\tilde{d}) = 0 \quad (6)$$

In other words, the mean of all distances selected from this distribution is 0. Let $\tilde{Z}_i$ be the random variable that takes values in the distribution followed by the distance estimates in the distance vector for group G_i. Since each z_i^j depends on the corresponding d_i^j, from Eqn. (6) we can claim that,

$$E(\tilde{Z}_i) = z_i \quad (7)$$

In order to compute $\theta(x, y)$, M needs distances z_i's to at least 3 or more deployment points so that multilateration can be done correctly. M first observes its neighborhood $(a_1, a_2 \dots a_k)$. Then, M computes the k distance vectors $\{(z_1^1, z_1^2 \dots z_1^{a_1}), (z_2^1, z_2^2 \dots z_2^{a_2}) \dots (z_k^1, z_k^2 \dots z_k^{a_k})\}$. It then computes the corresponding z_i by taking the mean of the corresponding $z_i^1, z_i^2 \dots z_i^{a_i}$ values, i.e.,

$$z_i = \frac{\sum_{j=1}^{a_i} z_i^j}{a_i} \quad (8)$$

It is obvious that larger the sample size a_i, better is the approximation for z_i. The best approximation is when distances from all the nodes in a group are available. But, an entire distance vector may not be available because of two reasons: 1) the whole group might not be in radio range (Figure 1(b)), or 2) some nodes in a group may be disabled (Figure 1(c)). Thus, we need to distinguish between these two cases and handle them separately. To do this we implement GSP on top of this algorithm to monitor group health. If the group is healthy ($h_i \geq m_i^s$) but still the target node hears from only a few nodes in a group; this would imply that not all nodes in that group are in the radio range. Otherwise, if the group is not healthy ($h_i < m_i^s$), the usefulness of the observation vector is determined by the number of nodes visible (a_i) and a parameter β_i discussed next.

Observe the neighborhood, i.e., $\{a_1, a_2 \dots a_k |\ a_i$ is the number of nodes from group G_i in radio range. $\}$.
for all groups G_i for which $a_i \neq 0$ **do**
  Compute $z_i^1, z_i^2 \dots z_i^{a_i}$.
  Observe health broadcasts (h_i) from the group head. Update h_i to the latest value for the group.
end for
for all groups G_i for which h_i is known **do**
  if The group is healthy, say $(h_i \geq m_i^s)$ **then**
    if $(a_i < \alpha_i)$ **then**
      Continue; {Sufficient samples not available for approximating z_i}
    else if $(a_i \geq \alpha_i)$ and $(a_i < \beta_i)$ **then**
      Compute $z_i = max\{z_i^1, z_i^2 \dots z_i^{a_i}\}$; {Samples for approximating z_i do not cover the entire distribution}
    else if $(a_i \geq \beta_i)$ **then**
      Compute $z_i = \frac{\sum_{j=1}^{a_i} z_i^j}{a_i}$ {Compute mean}
    end if
  else
    if $(a_i < \beta_i)$ **then**
      Continue;
    else
      Compute $z_i = \frac{\sum_{j=1}^{a_i} z_i^j}{a_i}$
    end if
  end if
end for
if z_i corresponding to at least 3 distinct groups G_i is known **then**
  Compute $\theta(x, y)$ by multilateration (using z_i's and their corresponding p_i's)
else
  print "Cannot do Localization!"
end if

Algorithm 2: ASFALT Localization Algorithm

4.2.1 Determining α_i and β_i

The ASFALT algorithm discussed above requires two parameters to determine if a distance sample or vector $(z_i^1, z_i^2 \dots z_i^{a_i})$ for any point i is large enough to approximate the distance z_i correctly. The *mean threshold* β_i for a group G_i is the minimum number of distance values required in the distance sample so that it correctly represents the original distribution of nodes around the deployment point (Normal). If the size of the observed sample is at least β_i then the algorithm computes the distance z_i as the mean of the distance values in the sample. If the size of the observed sample is less than β_i then it means only part of the group can be heard by the target node and z_i is computed as the largest value of the distances in the sample. Generally, $\beta = \frac{|m_i^s|}{2}$ works well for most cases. The *minimum threshold* α_i is the minimum number of distance values in the sample required to make any reasonable estimation of the distance z_i. If the size of the observed sample is less than α_i, we discard that observation from consideration in the localization process. This prevents inclusion of erroneous measurements in the multilateration process. α_i is generally assigned a low value. From our simulation experience, we observe that $\alpha_i \approx \frac{|m_i^s|}{3}$ works well for most cases.

5 Evaluation

In this section, we present a detailed evaluation of the GSP and ASFALT mechanisms using the sensor network simulation tool J-Sim [16] and compare their performance to the signature-based scheme proposed by Fang et al. [6]. In these experiments, deployment is done over a grid of 600m $\times$ 600m, consisting of 9 deployment points 100m apart as shown in Figure 3(a). Each deployment point has around 20 nodes deployed around it, following a 2D Normal distribution with mean (μ) as the corresponding deployment point and standard deviation (σ) as 50. Since we just want to observe the effects of node destruction on the accuracy of localization algorithms, we assume that the deployment size of every group is same, i.e., $m_i = m_i^s = 20\ \forall i$. Each

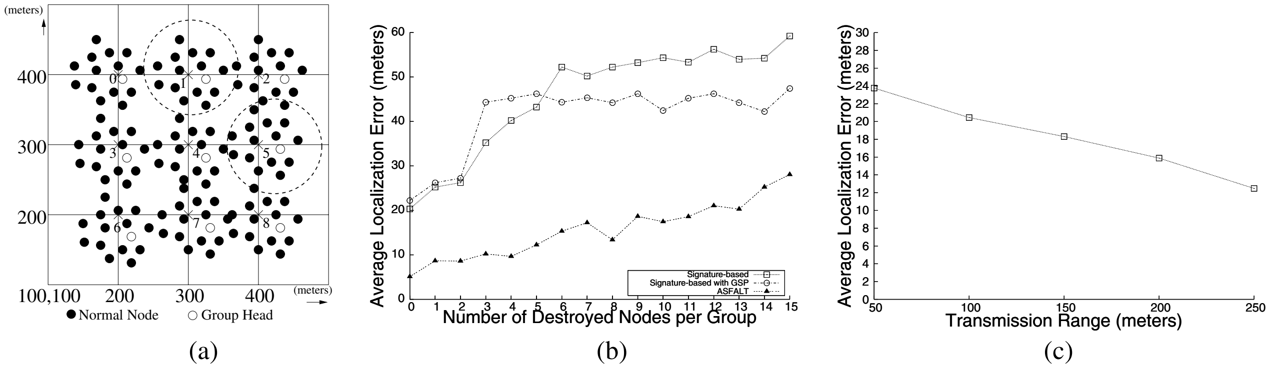


Figure 3. (a) Simulation setup - topology and node deployment (b) Plot of Localization Error vs. Number of Destroyed Nodes (c) Plot of Average Estimation Error vs. Transmission Range

group has a single group head. The estimation error is the Euclidean distance between the actual position and the estimated position of the node and is measured in meters. We plot the average of the estimation error of all the nodes only from group 4. This is done to avoid the boundary nodes, because localization errors in the boundary nodes are generally high due to lack of sufficient samples for localization.

In the first experiment, we simulate the signature-based approach, signature-based approach with GSP and the ASFALT algorithm in dynamic environments. In each simulation run, k nodes are destroyed from groups 1 and 5 (marked with dotted circles in Figure 3(a)), and the value of k varies from 1 up to 15. The simulation setup is not static, i.e, the node positions are not fixed throughout the experiment. Nodes in each group are reassigned new positions according to the 2D Normal distribution at the start of each simulation run. The transmission range of each node is fixed at 200m. The mean threshold β_i is 10 and the minimum threshold α_i is fixed at 5 for each group. We also assume that a group is healthy if its advertised health h_i differs from the original health m_i by at most 2. The simulation results are depicted in Figure 3(b). As we can see from Figure 3(b), the ASFALT localization approach performs much better as compared to the other two approaches and the average localization error of ASFALT increases much less sharply as compared to the other two. As the number of disabled nodes per group (for groups 1 and 5) increases the average localization error for all of the 3 algorithms increases. For lower number of destroyed nodes, the signature-based algorithm outperforms the GSP. This is obvious, as the GSP does not consider samples from groups 1 and 5 even when the number of destroyed nodes is low (but more than 2). The GSP performs marginally better than the signature-based algorithm when the number of disabled nodes is high. We have also conducted similar experiments for $\sigma = 100$, i.e., nodes are sparsely distributed around the deployment point. The trends in the performance of the algorithms is similar to the one shown in Figure 3(b), but the localization error is comparatively higher in this case.

In the second experiment, we observe the effect of radio range on the performance of ASFALT. The results are as expected (see Figure 3(c)). When the radio range increases, each node is able to cover a larger area and thus not only distance samples of a larger size are available from each group, but also more groups become available. As a result, the effect of node destruction is lesser when the node radio range is higher.

6 Comparison with Related Work

Despite the advances in the area of localization techniques for sensor network, the problem of fault-tolerant localization has not received much attention. Robust localization schemes in the presence of malicious nodes and erroneous range measurements exists [12, 13]. But, the problem of localization in the presence of erroneous measurements is different from the one in which entire nodes can be disabled after deployment. The most notable work in fault-tolerant localization was by Tinós et al. [17]. They present a novel fault tolerant localization algorithm developed for a system of mobile robots, called Millibots, that measure the distances between themselves and then use Maximum Likelihood Estimation to determine their location. In another related work, Ding et al. [4] propose a median-based mechanism for reducing the effect of faulty sensor nodes in target detection and localization algorithms. To the best of our knowledge, there has been no previous work specifically addressing efficient and fault-tolerant deployment strategies and signature-based localization schemes for ESNs.

7 Conclusion and Future Work

In this paper, we have addressed the problem of fault-tolerant node deployment and signature-based localization for ESNs. We have outlined an efficient strategy for node deployment in emergency applications, called the emergency level-based strategy. We have also proposed a simple enhancement to existing signature-based approaches, called Group Selection Protocol (GSP), that improves localization accuracy by monitoring changes in node distribution. Finally, we have proposed ASFALT, a novel yet simple, fault-tolerant technique for localization in ESNs that combines the salient features of both traditional range-based and signature-based approaches. Our simulation results have shown that ASFALT performs better compared to other signature-based techniques, especially in situation of high node destruction.

The improvement provided by GSP and ASFALT comes at the cost of extra communication (health status advertisement) and computation (distance estimation) overhead. Further evaluation is needed to compare the complexity and overhead of the proposed techniques against existing schemes and we intend to complete this as a part of future work. Moreover, in this work we assumed an ideal radio model (circular coverage area) which is not practical. As a part of future work, we would like to extend the current work to incorporate more practical radio propagation models like two-ray ground model, shadowing model, etc.

References

[1] P. Bahl and V. N. Padmanabhan. RADAR: an in-building RF-based user location and tracking system. In *IEEE INFOCOM Conference Proceedings*, pages 775–784. IEEE Communications Society, March 2000.

[2] J. Bruck, J. Gao, and A. A. Jiang. Localization and routing in sensor networks by local angle information. In *MobiHoc '05: Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, pages 181–192, New York, NY, USA, 2005. ACM Press.

[3] N. Bulusu, J. Heidemann, and D. Estrin. GPS-less low cost outdoor localization for very small devices. *IEEE Personal Communications Magazine*, pages 28–34, Oct 2000.

[4] M. Ding, F. Liu, A. Thaeler, D. Chen, and X. Cheng. Fault-tolerant target localization in sensor networks. *EURASIP Journal on Wireless Communications and Networking*, 2007:Article ID 96742, 9 pages, 2007. doi:10.1155/2007/96742.

[5] L. Doherty, L. E. Ghaoui, and K. S. J. Pister. Convex position estimation in wireless sensor networks. In *IEEE INFOCOM Conference Proceedings*, Anchorage, AK, April 2001. IEEE Communications Society.

[6] L. Fang, W. Du, and P. Ning. A beacon-less location discovery scheme for wireless sensor networks. In *IEEE INFOCOM'05 Conference Proceedings*, pages 161–171, Miami, FL, March 2005. IEEE Communications Society.

[7] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. F. Abdelzaher. Range-free localization schemes in large scale sensor networks. In *The Ninth Annual International Conference on Mobile Computing and Networking(MOBICOM)*. ACM SIGMOBILE, August 2003.

[8] J. Hightower and G. Borriello. Location systems for ubiquitous computing. *Computer*, 34(8):57–66, August 2001.

[9] M. Jadliwala, S. Upadhyaya, H. R. Rao, and R. Sharman. Security and dependability issues in location estimation for emergency sensor networks. In *The Fourth Workshop on e-Business (WeB 2005)*, Venetian, Las Vegas, Nevada, USA, December 2005.

[10] X. Ji and H. Zha. Sensor positioning in wireless ad-hoc sensor. networks using multidimensional scaling. In *Proceedings of IEEE INFOCOM 2004*, March 2004.

[11] C. A. R. Jr. Sensors bolster army prowess. *SIGNAL Magazine, AFCEA's International Journal*, 2004.

[12] L. Lazos and R. Poovendran. Serloc: secure range-independent localization for wireless sensor networks. In *The 2004 ACM workshop on Wireless security*, pages 21–30, Philadelphia, PA, October 2004. ACM SIGMOBILE.

[13] D. Liu, P. Ning, and W. Du. Attack-resistant location estimation in sensor networks. In *The Fourth International Symposium on Information Processing in Sensor Networks (IPSN '05)*, pages 99–106. ACM SIGBED and IEEE Signal Processing Society, April 2005.

[14] K. Lorincz, D. Malan, T. R. F. Fulford-Jones, A. Nawoj, A. Clavel, V. Shnayder, G. Mainland, S. Moulton, and M. Welsh. Sensor networks for emergency response: Challenges and opportunities. *IEEE Pervasive Computing, Special Issue on Pervasive Computing for First Response*, Oct-Dec 2004.

[15] N. Priyantha, A. Chakraborty, and H. Balakrishnan. The cricket location-support system. In *The Sixth Annual International Conference on Mobile Computing and Networking(MOBICOM)*, pages 32–43. ACM SIGMOBILE, August 2000.

[16] A. Sobeih, W.-P. Chen, J. C. Hou, L.-C. Kung, N. Li, H. Lim, H.-Y. Tyan, and H. Zhang. J-sim: a simulation and emulation environment for wireless sensor networks. *IEEE Wireless Communications*, 13(4):104–119, August 2006.

[17] R. Tinos, L. Navarro-Serment, and C. Paredis. Fault tolerant localization for teams of distributed robots. In *In Proceedings of IEEE International Conference on Intelligent Robots and Systems*, pages 1061–1066, Maui, HI, 2001.

[18] Y.-C. Tseng, M.-S. Pan, and Y.-Y. Tsai. Wireless sensor networks for emergency navigation. *Computer*, 39(7):55–62, 2006.

[19] M. A. Tubaishat and S. Madria. Sensor networks: an overview. *IEEE Potentials*, 2003.

[20] R. Want, A. Hopper, V. Falcao, and J. Gibbons. The active badge location system. *ACM Transaction on Information Systems*, pages 91–102, Jan 1992.

[21] L. Yu, N. Wang, and X. Meng. Real-time forest fire detection with wireless sensor networks. In *Proceedings. 2005 International Conference on Wireless Communications, Networking and Mobile Computing*, volume 2, pages 1214–1217, September 2005.