Beacon Deployment for Sensor Network Localization

Rui Huang and Gergely V. Záruba
Computer Science and Engineering Department
The University of Texas at Arlington
416 Yates, 300NH, Arlington, TX 76019
Email: rxh1725@omega.uta.edu, zaruba@uta.edu

***Abstract*— Localization in sensor networks is the process of obtaining geographical location information for all deployed sensors. Localizing each sensor node is becoming increasingly important as more and more algorithms and protocols in the disciplines of routing, energy management, and security have been proposed that rely on (semi-) accurate location information. In this paper, we study a particular approach to sensor node localization that uses sequentially deployed beacons to localize sensors. Our model assumes that the localization information can be obtained by observing beacons 1-hop away from the sensors. We show that under this simplified model the beacon deployment problem, i.e., to find the minimum number of beacons, is (still) NP-Complete. We then provide an approximation algorithm with a logarithmic approximation ratio based on a set cover greedy heuristic; we study both offline and online versions of this approximation. Finally, we incorporate the localization accuracy into the beacon deployment strategy by explicitly specifying the Cramer Rao Bound (CRB) as the condition. We verify the performance of the proposed algorithms using simulations, and show that accurate localization can be effectively achieved using the proposed algorithms.**

***Index terms:* sensor networks, localization**

I. Introduction

Location discovery for nodes in sensor networks is emerging as one of the more important tasks as it has been observed and shown that (semi-) accurate location information can greatly improve the performance of other tasks such as routing, energy conservation, or maintaining network security. For instance, algorithms such as LAR [1], GRID [2], and GOAFR+ [3] rely on the location information to provide more stable routes during unicast route discovery. The availability of location information is also required for geocast protocols such as LBM [4], GeoGRID [5], and PBM [6]. To minimize the power consumption, the GAF algorithm [7] uses location information to modify the network density by turning off certain nodes at particular instances.

In practice, it is often impossible to strategically deploy sensor nodes one-by-one with localizability in mind. For instance, sensor networks for military applications could be mass deployed from airplanes. In such case, one of the methods to obtain location information is by positioning a number of beacons *after* the sensors have been deployed. Since beacons are often expensive, there is a strong economical incentive to minimize their numbers.

In this paper, we consider the beacon deployment problem according to the following model. We assume that a certain number of unlocalized sensors already reside somewhere in a deployment area. The beacon deployment (BD) problem deals with the question of optimally deploying beacons to localize those nodes. The beacons are assumed to be deployed one at a time in an *online* fashion. After a beacon is deployed, we assume that it can identify the sensor nodes that it covers. This model can be realized in a number of ways. For instance, a mobile robot equipped with a GPS receiver can be sent to move over the deployment area. A computer onboard the robot can run an online algorithm to solve the beacon deployment problem. When this algorithm decides that a beacon should be deployed at a certain location, it will instruct the robot to go to that location and broadcast its coordinates there. Then, the algorithm decides on the next beacon location and moves there, repeating the above process. Alternatively, we can use a data dissemination method such as HEAP [8], where the beacon deployment algorithm is hosted on a centralized node at a fixed location. The result of each beacon deployment is disseminated through a virtual tree imposed on the sensor network, until it reaches the centralized node at the root.

In this work, we consider a simplified version of the beacon deployment (BD) problem where we assume that all sensor nodes in the network are to be localized *directly* from beacons. (By *directly*, we mean that sensor nodes need to be localized from the beacons 1-hop away, i.e., no multihop information is used during the localization.) The goal is to minimize the number of beacons while keeping the network localizable.

II. Previous Work

Localization of nodes in ad hoc and in sensor networks has been extensively studied in various contexts. In this section, we concentrate on the problem of localization by deploying beacons. An adaptive beacon placement algorithm is proposed in [9], where beacons are deployed sequentially based on empirically data of the perceived localization error. The perceived error is obtained by observations among neighboring beacons. When beacons are densely deployed over an area, the perceived localization error among them should reflect the error characteristics of the terrain or environment, which can then be applied to estimate the localization error of the actual sensor nodes. In [9] two beacon deployment algorithms are proposed, MAX and GRID, both of which deploy the beacons to locations where the estimated error is perceived to be at its maximum. The two algorithms differ in the size of the area they consider. Further work in [8] provides a framework to realize the adaptive algorithm in the real world by proposing a distributed algorithm to disseminate the perceived localization error into a centralized location.

The authors of [10] describe a beacon deployment strategy with a different objective: to minimize the number of deployed beacons (cameras) to localize mobile objects. They ask the question: "given the workspace and an error threshold, what is the minimum number, and placement of cameras so that the error in localization is less than the threshold at every point in the workspace? [10]". Two cameras, by the means of angle measurement, are needed to localize nodes. The goal is to minimize the number of cameras so that the overall error is below a threshold. While this method is not designed to localize sensors, it introduces the notion of explicit error thresholds as part of the design.

III. Our Contribution

In this work, we first show some computational complexity results of the beacon deployment (BD). We then propose a number of approximation methods to solve the problem. The approximation algorithms are presented in both offline and online versions. The offline version solves a simplified (although still a theoretically hard) problem by assuming that sensor node locations are known. The offline version is studied because it serves as the performance guideline to the online version. The online version deals with the realistic scenario of progressively deploying beacons to localize sensors of unknown locations. Finally, using simulations, we show that the result of the proposed offline and online algorithms, in terms of the total number of beacons deployed, is reasonably close to the optimum especially at higher sensor densities.

More precisely, we first show that the general optimal beacon deployment problem is NP-Complete (when the number of sensor nodes is the factor). We then propose an approximation algorithm based on the set cover problem to solve the beacon deployment problem that requires each sensor node to be covered by at least k beacons. We also extend the approximation algorithm to explicitly deal with issues that impact localization accuracy such as availability of beacons, measurement errors, and relative position of beacons. Our methods minimize the number of beacons (i.e., minimize equipment cost) under the accuracy constraint explicitly specified as a threshold to the Cramer Rao Bound (CRB).

Our work on the online version of the deployment methods is closest to that of [9], as both algorithms select the beacon location to minimize the localization uncertainty. The primary difference is that while the work in [9] measures the uncertainty using empirical data, we choose to use a more theoretical sound concept of error bounds. Furthermore, the algorithms in [9] use connectivity measurements (i.e., Centroid method) only. Our methods could as well work with ranging measurements (e.g., based on time of arrival or received signal strength measurements).

IV. Problem Definition and Complexity

The *offline* version of the BD problem asks for the minimum number of beacons needed to localize a given network. The network is "given" in the sense that we know all node locations. The offline version seems pointless since by definition all nodes have already been localized. But we study this problem so that we have a base for comparison (an optimal performer) for the online algorithms.

The condition for a node being localizable depends on the measurement type. Consider the trivial case where all nodes have ranging and signal direction measurement capacity; here one measurement from any beacon (i.e., one neighboring beacon) is sufficient to localize a node. If nodes have capabilities for only signal direction measurements, two beacon neighbors are required. Three beacon neighbors are required if only ranging measurements are available. In general, let the number of beacons required for a particular measurement type be k; we denote BD(k) as the version of beacon deployment problem where sensor nodes require measurements from k beacons, i.e., each sensor should have at least k beacon-neighbors for the sensor network to be localizable. In the following discussions we are going to assume that the deployment space for beacons is defined by a grid, i.e., we discretize the location space beacons can occupy from a continuous search space, so that we now deal with a combinatoric optimization problem. We denote V_{sensor} as the set of sensors, and V_{grid} as the set of all grid locations.

BD(k) is clearly NP, since when given a solution candidate (a certificate), a polynomial time algorithm can validate this solution in $O(k \cdot |V_{sensor}| \cdot |V_{grid}|)$. BD($k$), in fact, is a special case of the multiset covering (MSC) problem with special geometric constraints. We define MSC(k) as the problem of finding the minimum number of sets that would cover each element at least k times. The MSC is a general case of the set covering (SC) problem, which is known to be NP-Complete. Here, define a graph consisting of all sensors $\{V_{sensor}\}$ and grid locations $\{V_{grid}\}$ as vertices. An edge exists between a beacon/sensor pair *iff* the sensor can hear the beacon. Each vertex $v_{grid} \in V_{grid}$ can be viewed as a set containing the vertices in V_{sensor} such that there is an edge between v_{grid} and $v_{sensor} \in V_{sensor}$. Thus, the trivial case when $k = 1$ is just the set covering problem. The problem is still NP-Complete when $k > 1$.

V. Approximation Methods

A. *Offline Approximation of BD(k)*

For the set covering (SC) problem, there is a well-known greedy algorithm with approximation ratio to the optimal solution of $O(\ln(|X|))$, where $|X|$ is the size of the largest set . At each step, the greedy algorithm selects a set that covers the maximum number of uncovered members in the superset X, until the entire superset has been covered. The greedy algorithm performs close to the lower bound of the approximation ratio, as it has been shown in [11] that the unweighted set cover problem cannot be approximated within a factor $(1-\varepsilon)\ln(|X|)$, for any $\varepsilon > 0$.

Based on the greedy algorithm for set covering, we can give a pseudo code for the BD(k) offline greedy algorithm as shown in Algorithm 1. At each step, the algorithm selects a beacon location from V_{grid} that covers the most sensor nodes that haven't been k-covered. The algorithm stops once all sensor nodes have been k-covered and returns the set of beacons selected. In terms of complexity, the outer loop at line 5 runs at most $O(min(k \cdot |V_{sensor}|, |V_{grid}|))$ times, and the loop statements between line 6 and 11 can be implemented to run in $O(|V_{sensor}| \cdot |V_{grid}|)$. Thus, the overall runtime complexity is $O(min(k \cdot |V_{sensor}|, |V_{grid}|) \cdot |V_{sensor}| \cdot |V_{grid}|)$

Algorithm 1 Greedy Offline Approximation of BD(k)

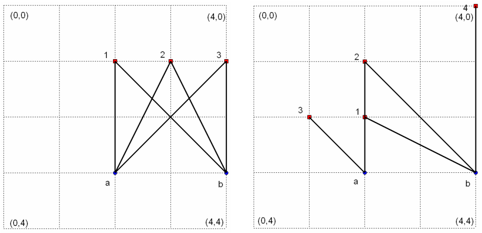
```
 1: operator $i \dashv j$: location $i$ can be covered by broadcast from location $j$
 2: initialize $p_s \leftarrow 0$ for all $s \in V_{sensor}$
 3: $U \leftarrow V_{grid}$
 4: $C \leftarrow \emptyset$
 5: while $\exists s \in V_{sensor} : p_s < k$ do
 6:     select $g_{max} \leftarrow \arg\max_{g \in U} |\{s : p_s < k, s \dashv g\}|$
 7:     for all $s : p_s < k, s \dashv g_{max}$ do
 8:         $p_s \leftarrow p_s + 1$
 9:     end for
10:     $U \leftarrow U - \{g_{max}\}$
11:     $C \leftarrow C \bigcup \{g_{max}\}$
12: end while
13: return $C$
```

Even through BD(k) is a generalization of the set covering problem where $k > 1$, the same logarithmic approximation ratio can be obtained, as stated by the following theorem.

Theorem 5.1: The greedy offline approximation algorithm of BD(k) has an approximation ratio of $\ln(|V_{sensor}|) + 1$.

Proof: Omitted due to page limit. ■

Fig. 1(a) depicts a simple scenario illustrating the operation of the offline greedy algorithm. The scenario consists of two sensor nodes a and b, shown as circles, residing in a deployment area of a 5x5 grid. The task is to deploy the minimum number of beacons, shown as squares, so that each sensor node connects to at least three beacons ($k = 3$). The coverage radius of the beacons is 3 (i.e., a beacon at $(0,0)$ will cover sensors at $(0,1)$, $(0,2)$, $(0,3)$, $(1,0)$, $(1,1)$, $(1,2)$, $(2,0)$, $(2,1)$, $(2,2)$ and $(3,0)$). As seen in Fig. 1(a), a sequence of three beacons are deployed by the offline greedy algorithm. The algorithm keeps a potential variable p_s for each sensor s. Initially, the potentials are $p_a = 0$, $p_b = 0$, and the set U contains all 25 grid points. The first beacon is selected at grid $(2,1)$ as a greedy choice because the beacon would cover both sensor nodes. Evidently, there is more than one grid point that covers both, but we simply choose one of them. After the first beacon location is selected, the potentials of both sensor nodes increase to 1 ($p_a = 1$ and $p_b = 1$). The selected location $(2,1)$ is removed from the set U and added to the cover set C. Since both potentials (p_a and p_b) are not k yet, a second beacon is inserted and its location is greedily selected from the remaining grid points in U. The algorithm continues until both potentials reach k (when both sensor nodes are k covered).



(a) Offline (b) Online

Fig. 1. Example of the greedy approximation of BD(k)

B. Online Approximation of BD(k)

The offline algorithm of solving BD(k) assumes that the locations of the sensor nodes are known, which essentially defeats the purpose of the beacon deployment. However, as shown even this simplified problem is NP-Complete when trying to minimize the number of beacons. In this section, we remove the node position knowledge assumption. We consider this problem to be an *online* problem in the sense that we have to select the next beacon location based on the feedback (i.e., how many sensor nodes have been covered) of the previous beacon locations. (In the offline version the feedback result is known *before* the beacon is deployed.) Our basic assumption of the online version is that sensor nodes are uniformly distributed over the entire deployment area. We also assume knowledge on the population of the sensor nodes. Based on these assumptions, we can design a greedy online deployment algorithm, Algorithm 2, that selects beacon locations by maximizing the coverage probability.

The online greedy algorithm maintains $p_{s,g}$ as the probability of the sensor node s residing at grid location g. Assuming that sensor nodes are uniformly distributed, $p_{s,g}$ is initialized to $1/|V_{grid}|$, where $|V_{grid}|$ is the size of the potential beacon locations. To select the next beacon's location, the algorithm first sums up all the neighborhood probabilities $p_{s,g}$ for each grid point; a greedy choice is then made to pick the grid point with the maximum overall $p_{s,g}$ value. For each of the sensor nodes that are already covered k times, the $p_{s,g}$ value is set to 0, indicating that the k-cover condition of this node is satisfied. For all other sensor nodes, the procedure $AdjustProbability$ (Algorithm 3) is called to adjust the $p_{s,g}$ values based on the latest beacon choice. The purpose of $AdjustProbability$ is to reduce the scope of the possible location for each sensor node and thus help the algorithm to make better choices in subsequent iterations. The algorithm stops when all the probabilities ($p_{s,g}$) become zero, i.e., when all sensor nodes are k covered. Fig. 1(b) depicts a possible outcome of the online greedy algorithm given the same scenario of localizing two sensor nodes in a 5x5 grid and $k = 3$. The shown instance of the online algorithm took four beacons (one more than the offline version).

Algorithm 2 Greedy Online Approximation of BD(k)

```
 1: operator $i \dashv j$: location $i$ can be covered by broadcast from location $j$
 2: initialize $p_{s,g} \leftarrow 1/|V_{grid}|$ for all $s \in V_{sensor}$ and $g \in V_{grid}$
 3: $U \leftarrow V_{grid}$
 4: $C \leftarrow \emptyset$
 5: while $\exists s \in V_{sensor}, g \in V_{grid} : p_{s,g} > 0$ do
 6:     select $g_{max} \leftarrow \arg\max_{g \in U} \sum_{g' \dashv g} \sum_s p_{s,g'}$
 7:     for all $s \in V_{sensor}$ do
 8:         if $s$ is covered by $k$ beacons then
 9:             $p_{s,g} \leftarrow 0$ for all $g$
10:         else
11:             $AdjustProbability(p, s, g_{max})$
12:         end if
13:     end for
14:     $U \leftarrow U - \{g_{max}\}$
15:     $C \leftarrow C \bigcup \{g_{max}\}$
16: end while
17: return $C$
```

Algorithm 3 $AdjustProbability(p, s, g)$

```
sum ← 0
if $s \dashv g$ then
    for all $g' : g' \not\dashv g$ do
        $sum \leftarrow sum + p_{s,g'}$
        $p_{s,g'} \leftarrow 0$
    end for
    $I \leftarrow$ set of all $g' : g' \dashv g, p_{s,g'} > 0$
    for all $g' \in I$ do
        $p_{s,g'} \leftarrow p_{s,g'} + sum/|I|$
    end for
else
    for all $g' : g' \dashv g$ do
        $sum \leftarrow sum + p_{s,g'}$
        $p_{s,g'} \leftarrow 0$
    end for
    $I \leftarrow$ set of all $g' : g' \not\dashv g, p_{s,g'} > 0$
    for all $g' \in I$ do
        $p_{s,g'} \leftarrow p_{s,g'} + sum/|I|$
    end for
end if
```

The initialization phase of Algorithm 2 has a runtime of $O(|V_{sensor}| \cdot |V_{grid}|)$. The condition of the outer loop at line 5 can be validated in $O(|V_{sensor}| \cdot |V_{grid}|)$. Within the outer loop, the greedy selection step at line 6 has a runtime of $O(|V_{sensor}| \cdot |V_{grid}|)$, and the *for* loop at line 7 also has a runtime of $O(|V_{sensor}| \cdot |V_{grid}|)$ (the runtime of the function $AdjustProbability$ is $O(|V_{grid}|)$. Thus, the total run time of each iteration of the outer loop is $O(|V_{sensor}| \cdot |V_{grid}|)$. In the worst case, the algorithm will cover all grid points, in which case the outer loop will execute in $O(|V_{grid}|)$ time. Overall, the worse case runtime complexity of the algorithm is $O(|V_{sensor}| \cdot |V_{grid}|^2)$.

One note on the actual implementation: although the online greedy algorithm assumes that the sensor nodes are uniformly distributed, it is equally effective if the spatial distribution of sensors is known before the deployment process starts. One can simply adjust the initial probability distribution to match the perceived location distribution. If it is expected that the sensor nodes are more likely to concentrate on a smaller area in the deployment region then the probability distribution should be adjusted to give higher probability to the locations within this area. A more accurate initial probability distribution will enable the online algorithm to make better decisions.

C. Measurement Quality and CRB

Our prior analysis makes the simple assumption that for a particular measurement model (i.e., ranging, angle, or both), k measurements are required to localize the node. Each measurement is treated equally. In reality, however, noisy measurements imply that not all measurements should be treated equally. Certain measurement might contain more useful information than others. Take localization using ranging for instance, the angle between the measurements can be important. Intuitively, we would prefer the ranging measurements to come from more "spread out" beacons. Fortunately, a convenient metrics is available to numerically qualify the perceived localization error in relation to the measurements.

In the context of localization, the Cramer Rao Bound (CRB) [12] is often used to mathematically qualify the lower bound on the localization error. For the localization problem, the CRB is a function of the following: 1) the relative locations of the sensor nodes and the beacons, 2) the measurement model determined by the measurement type, and 3) the noise model characterizing the terrain and the environment. Previous works have derived the CRB formulas for a number of measurement types including RSSI ranging, TOA ranging, and AoA [12]. Note that the actual localization algorithm being used has no impact on the CRB. Thus, the CRB is essentially a bound determined solely by the particular localization scenario. It gives an indication of how difficult a particular localization scenario is and what is the best *any* localization algorithm can do given the scenario.

D. CRB-Based Approximation of BD

We raised the issue that the localization accuracy is impacted by the beacon location and the measurement noise. Depending on the particular beacon location, the actual localization result can vary greatly because of the measurement noise. Thus, in practice it might not be sufficient to only require that each sensor node is covered by k beacons ($k = 3$ for 2-dimensional ranging localization), since the localization accuracy varies depending on how the k beacons are placed. In this section, we address the accuracy issue in the BD problem. In particular, we modify the offline and online approximation algorithms presented earlier to use the CRB as the stoppage and greedy selection criteria.

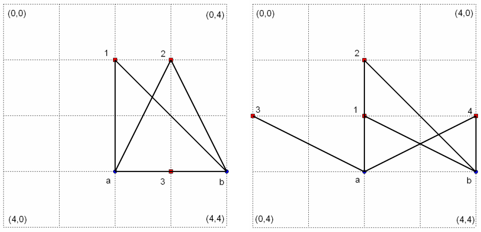
For the greedy offline algorithm, we can modify it such that it will continue to deploy beacons until the CRBs of all sensor nodes are reduced below a given threshold crb_T. For each sensor node s, we track the location of beacons that cover s. Let function $CRB(s)$ return the CRB of the node s with the already deployed beacons. We also define a function called $CRB^P(s,g)$ that returns the potential of the new CRB with a new beacon g added. (The CRBs of initially unlocalized sensor nodes are set to infinity.) At each iteration, a greedy choice, g_{max}, is made to pick the next beacon location that will cause the maximum reduction to the CRB of the sensor nodes, i.e., maximizing $CRB^P(s,g) - CRB(s)$:

$$g_{max} \leftarrow \arg\max_{g \in U} \sum_{s : s \dashv g} (CRB^P(s,g) - CRB(g))$$

For the online version, since the exact location of each sensor node s is unknown, we present it as a probability distribution over the entire deployment area, where $p_{s,g}$ is the probability of node s residing close to the grid location g. Furthermore, the CRB calculation needs to based on a perceived instead of the exact sensor location. Thus, if a sensor s is perceived to be located at g', $CRB_{g'}(s)$ returns the CRB of s at g'. The greedy beacon selection is then based on maximizing the sum of $p_{s,g'} \cdot (CRB^P_{g'}(s,g) - CRB_{g'}(s))$ for all locations g' within the range of the beacon at location g. In other words, the greedy location choice is the one that maximizes the overall CRB reduction of all sensor nodes considering their probabilities of residing within the beacon range:

$$g_{max} \leftarrow \arg\max_{g \in U} \sum_{g' \dashv g} \sum_{s} p_{s,g'} \cdot (CRB^P_{g'}(s,g) - CRB_{g'}(s))$$

Fig. 2 shows results of the offline and online greedy algorithms of BD(crb_T) running on a simple scenario of two sensor nodes (with the same RSSI parameters as in [13]). The CRB threshold is set to $crb_T = 1$. The sample run for the offline version selected three beacons while the online version picked four. Comparing to the results in Fig. 1, which were obtained using k-cover ($k = 3$) as the objective, it is clear that the CRB requirement gives better spread of the beacon locations. In the case of the offline version, the three beacons selected in Fig. 1(a) constitute a spread of 45° angle for both sensor nodes, while the beacons in Fig. 2(a) constitute a spread of 90° for node a and 63° for node b. Wider angle spread is also observed in the online version. The better spread of the beacon locations dictated by the CRB requirement would ultimately result in more accurate localization.



(a) Offline (b) Online

Fig. 2. Example of the greedy approximation of BD(crb_T)

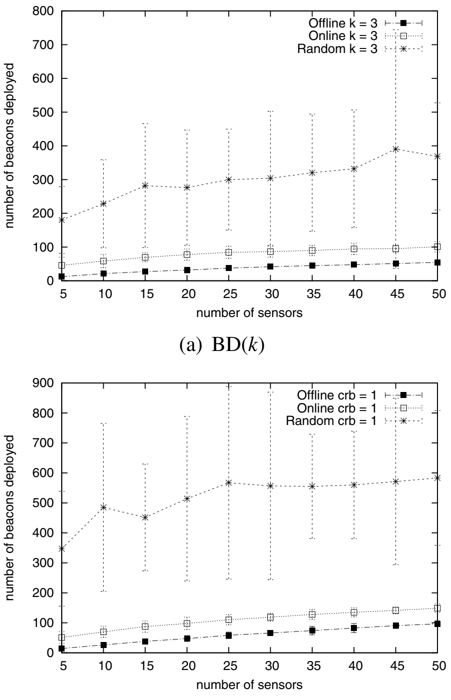
In practice, the threshold crb_T is a system configurable variable, which should be adjusted according to the desired localization accuracy. As expected, a lower crb_T would result in better localization accuracy at the expense of more deployed beacons. The choice of crb_T also affects the runtime complexity. In the worst case, when crb_T is close to zero, the outer loops of both offline and online versions would continue executing until all potential beacon locations are used, i.e., $O(|V_{grid}|)$. This worst case runtime complexity of the offline and online algorithm is $O(|V_{grid}|^3 + |V_{sensor}| \cdot |V_{grid}|^2)$ and $O(|V_{grid}|^4 + |V_{sensor}| \cdot |V_{grid}|^2)$, respectively.

VI. Simulation Results

We evaluate the performance of the various BD approximation methods using computer simulations. The simulation environment consists of a number of sensor nodes uniformly distributed over a deployment area of a 50 by 50 grid. For every deployment scenario, we run each of the approximation methods and collect the number of beacons needed to localize all sensor nodes. The beacons are assumed to have a transmission range of 5 grid radius. For BD(k), we assume a 2-D localization which requires $k = 3$. In the case of BD(crb_T), we assume the range is obtained via RSSI readings with a constant Gaussian noise added. We use the same signal propagation and noise model as in [13], with the reference distance $d_0 = 1m$, the path loss exponent $n_p = 2$, and the noise variance $\sigma^2_{dB} = 1.7$. In addition to BD(k) and BD(crb_T), we also include the result of a simple random deployment algorithm, in which the beacon locations are i.i.d. uniform randomly selected until the stoppage criteria ($k = 3$ or $CRB < crb_T$) is satisfied. The simulation result presented in the figures is the average of 30 different runs with the 95% confidence interval shown as the vertical lines.

Fig. 3 compares the total number of beacons deployed for each of the approximation methods while varying the number of sensor nodes residing in the deployment area. Fig.3(a) shows the result of using $k = 3$ as the stop condition, and Fig.3(b) uses CRB threshold of $crb_T = 1$. As expected, in both the k and the crb_T cases, the online approximation methods deploy more beacons than the offline versions. However, the difference does not seem to increase as the number of sensor nodes increases. This indicates that the online version would be preferable with a denser deployment area, since the relative difference between the online and offline version becomes smaller as the number of sensor nodes increases.

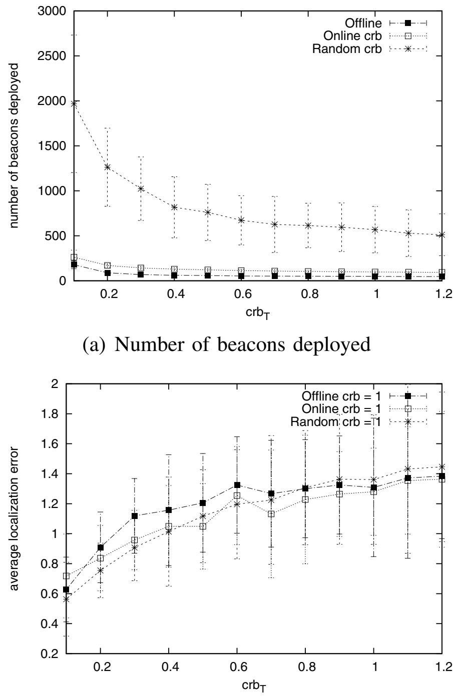
To demonstrate the effect of the threshold crb_T in beacon deployment, we use a scenario of 20 sensor nodes in the deployment area while varying crb_T from 0.1 to 1.2. The number of beacons deployed by the random algorithm, and offline and online version of the BD(crb_T) approximation methods are shown in Fig.4(a). Note that as the crb_T threshold decreases, the increase in the beacon population is faster than linear. As shown in Fig.4(b), the actual localization error decreases as the crb_T lowered. Furthermore, Fig.4(b) shows that the localization error stays relatively constant for all three deployment algorithms, even though the random algorithm deploys a significantly higher number of beacons than online and offline BD(crb_T). This result demonstrates that the selected threshold crb_T effectively controls the localization error. *Thus, by selecting the appropriate crb_T, we can balance the trade-off between the localization accuracy and the overhead associated*



(a) BD(k)

(b) BD(crb_T)

Fig. 3. Number of beacons deployed with various BD approximation methods



(a) Number of beacons deployed

(b) Average localization error

Fig. 4. Using the threshold crb_T in BD(crb_T) as a factor

with the number of beacons deployed.

VII. Conclusions

In this paper, we have studied the beacon deployment problem for localizing sensors (i.e., sensor nodes). The objective is to deploy the minimum number of beacons to localize all sensor nodes. We have proposed a number of approximation algorithms to solve this NP-Complete problem. An offline version of the approximation algorithm greedily picks the beacon location that covers the largest number of uncovered sensor nodes and is used as a basis for result comparison. The online version maintains a probability distribution of the estimated node locations, and selects a location for the beacon by maximizing the potential of reducing the overall variances of the location distributions. We further describe a variation of the proposed algorithms that uses the Cramer Rao Bound (CRB) as the evaluation criteria, which incorporates the localization accuracy into the deployment problem.

While our simulations only consider the ranging using RSSI as the measurement, the proposed framework will work for any other measurement types such as ToA and AoA. Essentially, the k-cover and crb_T requirements are independent of the measurement types except for the model from which k and CRB are calculated. While in this work the sensor nodes are localized via the beacons directly, we are working on extending our framework to allow collaboration among sensor nodes. Intuitively, such the collaboration could result in smaller number of beacons required, as sensor nodes would need to hear less of the beacons directly.

References

[1] Y. Ko and N. H. Vaidya. “Location-Aided Routing (LAR) in Mobile Ad Hoc Networks,” *Wireless Networks*, vol. 6(4), pp. 307 - 321, July, 2000.

[2] W.-H. Liao, Y.-C. Tseng, and J.-P. Sheu. “GRID: A Fully Location-Aware Routing Protocol for Mobile Ad Hoc Networks,” *Telecommunication Systems*, vol. 18(1), pp. 37-60, 2001.

[3] F. Kuhn, R. Wattenhofer, Y. Zhang, and A. Zollinger. “Geometric Ad-Hoc Routing: Of Theory and Practice,” in Proc. of *ACM PODC’03*, pp. 63-72, 2003.

[4] Y. Ko and N. Vaidya. “Geocasting in Mobile Ad Hoc Networks: Location-Based Multicast Algorithms,” in Proc. of *IEEE WMCSA’99*, 1999.

[5] W.-H. Liao, Y.-C. Tseng, K.-L. Lo, and J.-P. Sheu. “Geogrid: A Geocasting Protocol for Mobile Ad Hoc Networks Based on Grid,” *Journal of Internet Technology*, vol. 1(2), pp. 23-32, 2000.

[6] M. Mauve, H. Fuler, J. Widmer, and T. Lang. “Position-Based Multicast Routing for Mobile Ad-Hoc Networks,” *Technical Report TR-03-004, Department of Computer Science, University of Mannheim*, 2003.

[7] Y. Xu, J. Heidemann, and D. Estrin. “Geography-Informed Energy Conservation for Adhoc Routing,” in Proc. of *ACM/IEEE MOBICOM’01*, pp. 70-84, 2001.

[8] N. Bulusu, J. Heidemann, V. Bychkovskiy, and D. Estrin. “Density-adaptive Beacon Placement Algorithms for Localization in Ad Hoc Wireless Networks,” in Proc. of *IEEE INFOCOM’02*, 2002.

[9] N. Bulusu, J. Heidemann, and D. Estrin. “Adaptive Beacon Placement,” in Proc. of *ICDCS-21*, pp. 489-498, April 2001.

[10] V. Isler. “Placement and Distributed Deployment of Sensor Teams for Triangulation Based Localization,” in Proc. of *IEEE ICRA’06*, pp. 3095-3100, May, 2006.

[11] U. Feige, “A Threshold of $ln(n)$ for Approximating Set Cover,” *Journal of the ACM*, vol. 45, pp. 634-652, 1998.

[12] N. Patwari, A. Hero, J. Ash, R. Moses, S. Kyperountas, and N. Correal, “Locating the Nodes: Cooperative Geolocation of Wireless Sensors,” *IEEE Signal Processing Magazine*, vol. 22(4), pp. 54-69, July 2005.

[13] N. Patwari, A. O. Hero III, M. Perkins, N. S. Correal, and Robert J. O’Dea. “Relative Location Estimation in Wireless Sensor Networks,” *IEEE Trans. on Signal Processing*, vol. 51(8), pp. 2137-2148, August 2003.