# Mobile Node Localization Using Cooperation and Static Beacons

Troy Johnson
Department of Computer Science
Central Michigan University
Mount Pleasant, MI 48859
johns4ta@cmich.edu

Patrick Seeling
Department of Computer Science
Central Michigan University
Mount Pleasant, MI 48859
Email: pseeling@ieee.org

***Abstract*—Location-based services are one of the demanding applications on mobile devices that account for significant battery usage. In this paper, we evaluate the potential average savings over using approaches such as GPS for localization that can be achieved by ($i$) using local low-energy mobile node cooperation and/or ($ii$) additional local infrastructure (fixed) beacon nodes that broadcast their position. Using a random walk simulation approach, we find that significant savings could be attainable; we determine above 20% for 50 nodes in a 100x100 area in the presence of beacons and 18% without. We additionally note that the simulated nodes are able to determine their location fairly accurately within the simulated area. We finally observe that random background refreshes (such as those triggered by other applications) have only little impact on the potential for savings.**

***Index Terms*—Localization, cooperation, energy savings, simulation**

## I. Introduction

Location-based services have garnered a tremendous success both in research and implementation, especially due to the now ubiquitous availability of mobile devices. In addition, several location-based and context-based services rely on an accurate localization. Mobile users today enjoy location-specific information at their fingertips, whereby the actual location information is traditionally contributed by ($i$) satellite-based positioning systems, such as GPS and GLONASS, ($ii$) cellular network triangulation through base stations, and ($iii$) location fingerprinting where the location is determined by a cloud-based service based on local radio signals. Some of these services are complimentary to another and based on the current usage scenario, which typically is given by an application the user interacts with on the mobile device. In certain contexts, however, the utilization of an individual localization service is not possible, e.g., satellites are typically not visible when being indoors or fingerprinting is not possible when no connection to a centralized server is available. In addition, power consumption considerations play into the selection of the individual localization services as well; typically a relationship between the accuracy, time to obtain a location, and the actually required battery consumption during the time all are in interplay with each other. With recent communication advances considering cooperative approaches, several generic approached to cooperative localization were evaluated in the past, see, e.g., [1]. The authors found for a trace-based evaluation that if half of the nodes shared their location and GPS was usable half the time, the overall localization costs (as determined by energy used and accuracy) could be significantly reduced. Other localization techniques and mechanisms, geared towards indoor localization have been evaluated in the past, see, e.g., [2] for a summarizing overview. Without significant optimization, we found the overall accuracy to be in the sub-meter range, which is granular enough for indoor localization, as also reported by Nokia in [3], [4]. Other cooperative location exchanges were shown to successfully operate in, e.g., [5], where the authors describe a novel architecture that uses local mobile terminal cooperation to exchange multiple cell IDs of cellular networks to increase accuracy. As described in [6], [7], sharing among nodes can be accommodated either via device names or lower level protocol functions as well as traditional services. The advent of broad deployment of BLE (Bluetooth v4 Low Energy mode, also referred to as Bluetooth v4 SmartDevice mode) devices and their incorporation into mainstream mobile consumer devices, such as smartphones and tablets, leans itself towards location sharing among mobile nodes using the Bluetooth LE beacon mode. The Bluetooth beacon mode is realized through the Proximity (PXP) and Find Me (FMP) profiles in the Bluetooth standard, see, e.g., [8], which allows this feature to be implemented in most of the recently available commercial off-the-shelf (COTS) mobile consumer devices. With typical ranges of the low power Bluetooth devices of about 10m, this approach is suitable to convey messages with almost no energy impact amongst mobile devices in a peer-to-peer manner.

In this paper, we extend these prior works by closer evaluation of the cooperative location sharing among mobile nodes with and without supportive location beacons. The remainder of this paper is structured as follows. In the following Section II, we present our overall scenario (model) and underlying assumptions while we provide the corresponding details for the NetLogo simulation environment we utilized in Section III. In Section IV, we present evaluation results of our model. We conclude in Section V with an outlook on future research activities.

## II. Cooperative Localization Scenario

In a cooperative localization scenario, nodes determine their location either by using their own equipment or by obtaining location information through neighboring nodes in transmission range of short-range radios, such as Bluetooth. These neighboring nodes can be either fixed location beacons or other mobile devices.

In this approach, a mobile node $n$ out of $N$ total mobile nodes tries to determine its location with the help of $i$ neighbors, with $i = 0, \ldots, 3$, using trilateration based on simulated signal strength and accuracy. As outlined in prior works, e.g., in [6], in addition to location and estimated distance, an accuracy estimator is assumed to be available as well.

We assume that nodes move freely within the environment, using the random walk model with a maximum speed of $\Delta$. Every time a node $n$ wishes to calculate its location, it attempts to do so by initially utilizing information obtained from the surrounding fixed (beacons) or other mobile nodes. If the determined accuracy $a_i$ is not within a certain numeric limit $A$, e.g., $A = 5$ meters, the node utilizes its own GPS in order to determine its location. Without loss of generality, we assume that the mobile nodes tend to periodically refresh their GPS (e.g., based on other applications in case of smartphones) with a random refresh rate $R$.

## III. NetLogo Model Details

We simulate the overall behavior of our model using the NetLogo simulation toolkit [9]. The NetLogo toolkit is well suited for social interaction modeling of agent behavior on a macro (population) and micro (individual agent) level. Within NetLogo, we created the models for simple user mobility (we use the random walk model) and the location determination, as well as performance evaluation over time. We illustrate our developed simulation environment in Figure 1.

Each NetLogo simulation starts with a general setup phase, whereby the mobile nodes $n$ are randomly placed on a square area with length $l$ together with a number of fixed (beacon) nodes $B$. We note that NetLogo uses the notion of a patch, which we normalize to the equivalent of $1\text{m}^2$. Each node calculates a random time to call its GPS based on the refresh rate $R$, whereby the time $r_i$ is drawn randomly from an exponential distribution with a mean of $\lambda^{-1} = R$. Finally, each node's self-calculated position values are initialized to their actual positions within the simulation environment.

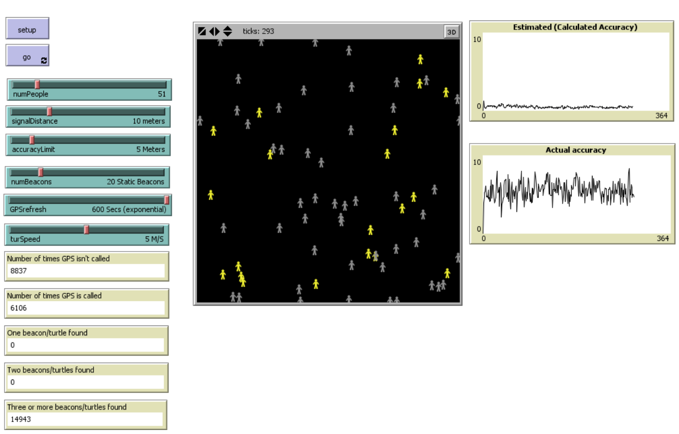


Fig. 1. Screenshot of the Netlogo simulation environment evaluating the cooperative localization scheme.

After this setup phase, the actual simulation takes place for a time of $T$ ticks, which we normalize to seconds to provide a simplified real-world mapping. While the simulation is ongoing, each node performs the following four main algorithmic steps move, scan, calculate and update.

---

**Move** Rotate node uniformly distributed $\varphi \in [0, 360)$ and move uniformly distributed $d \in [0, \Delta)$.

**Scan** Determine count of other nodes $n_i, i \in [0, 3]$ in range, sorted by accuracy $a_i$ (reported by $n_i$ as self-calculated with added random factor).

**Calc.** Calculation of new position $p$ and movement $\delta$ based on $n_i$ positions $p_i$:

- **if** $i = 1$ **then**
  - Set $a = |p - p_1| + a_1$ (distance)
  - Set $p = p_1$
- **else if** $i = 2$ **then**
  - Set $p = \frac{p_1 + p_2}{2}$ (midpoint)
  - Set $a = \left| p - \frac{p'_1 + p'_2}{2} \right|$, where $p'_i = p_i + \frac{a_i}{2}$
- **else if** $i \geq 3$ **then**
  - Set $p = \frac{p_1 + p_2 + p_3}{3}$ (trilateration)
  - Set $p' = \frac{p'_1 + p'_2 + p'_3}{3}$, where $p'_i = p_i + a_i$
  - Set $a = |p - p'|$ (distance)
- **else**
  - Utilize GPS: set $p$ to actual position and $a$ to 1
- **end if**

Determine calculated moved distance $\delta$

**Upd.**
- **if** $a > A$ **then**
  - Utilize GPS
- **else**
  - **if** $\delta = 3\Delta$ **then**
    - Utilize GPS
  - **end if**
- **end if**

Random GPS update, set next GPS call time $r$

---

We utilize a random walk model, whereby each mobile node draws uniformly distributed random numbers for rotation $\varphi \in [0, 360)$ and distance $d \in [0, \Delta)$ for each time tick $t, t = 0, \ldots, T$. Next, the node scans for neighboring fixed or mobile nodes within range (assumed 10m in our approach). Found neighbors are sorted in ascending order based on accuracy; if no neighbors are found within the signal distance, the node under consideration utilizes its GPS. Every time the GPS is utilized, the accuracy value of the node calling GPS is set to 1m and its calculated position is set to the actual position in the simulated area. We note that each of the simulated nodes performs the outlined algorithms independently, while global observation is used to determine the performance achieved.

In the case of one or more available references, the mobile node under consideration determines its location and accuracy as follows. Initially, the distance between the simulated nodes is determined, similar to the typically RSSI-based distance calculations used for real devices. To account for fluctuations and inaccuracies in using real radio signals, a randomly drawn error value $\varepsilon$ is added to the distances calculated in the following three scenarios, whereby $\varepsilon$ is normal distributed $N(M = 1, \sigma = 1)$, as motivated by [3], [4], [6].

If one neighbor is found within the radius of the signal range, the mobile node sets its position to the position reported by that reference and sets its own accuracy to the distance between itself and the reference. If exactly two neighbors are found within the radius of the signal range, then the latitude, longitude, and distances of the two references are used to estimate the location of the mobile node under consideration. This estimation uses the intersections that are generated by the intersections of the two circles. The accuracy will then be the distance from the midpoint to one of the intersections. In the ideal situation, at least three neighboring nodes are found within the signal range and trilateration is used to determine a node's location. To determine the accuracy, trilateration is used again, but the accuracy of each of the three references is added to their calculated distance from the node determining its location to simulate a worst-case scenario. This determines the coordinates of the alternate position $p'$. Subsequently, the distance between the two positions is determined as the accuracy $a$ for this node's location.

Two cases may require a node to utilize its GPS, namely $(i)$ if the determined accuracy value $a$ is larger than the required accuracy threshold $A$ and $(ii)$ if the detected displacement is larger than three times the maximum allowable node speed. Finally, a node might periodically refresh its GPS based on other applications. If a random refresh occurs, the node reschedules a new time to refresh similar to the initial setup.

## IV. Evaluation Results

In this section, we provide an evaluation of the impact of different factors on the potential for savings. We count the number of localizations that are due to usage of the GPS as $C_{\text{GPS}}$ and the ones due to cooperation or beacons as $C_{\text{Coop}}$. The savings are calculated as a fraction of the local location updates from beacons or other mobile nodes over all localizations consisting of local and GPS updates as

$$S = \frac{C_{\text{Coop}}}{C_{\text{GPS}} + C_{\text{Coop}}}$$

and averaged over all nodes $N$. Furthermore, we note that we assume continuous localization needs, i.e., we assume that nodes determine their location at every "tick" $t$. Specifically, we use the simulation parameters described previously in Section III as provided in Table I. We provide the results obtained from 100 simulation runs, noting that the 95% confidence interval for the observed savings in the presented cases is less than 0.5% (with a maximum observed standard deviation of 2.4%).

TABLE I
Overview of simulation parameters.

| Variable | Symb. | Values/Units |
|---|---|---|
| Square area length | $l$ | 100 [m] |
| Total time | $T$ | 7200 [s] |
| Desired accuracy | $a$ | 5 [m] |
| Random refresh rate | $R$ | $[200, \ldots, 600, \infty]$ [s] |
| Random speed max. | $\Delta$ | $[1, 3, 5, 7, 10]$ [m/s] |
| Number of mobile nodes | $N$ | $[10, 20, \ldots, 50]$ |
| Number of beacons | $B$ | $[0, 2, \ldots, 10]$ |

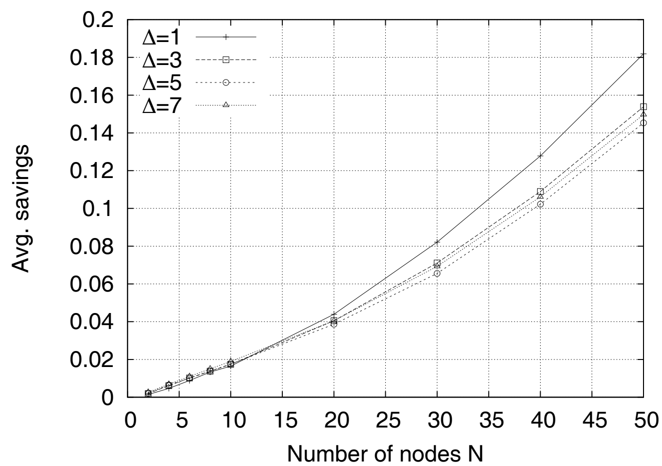


Fig. 2. Impact of the number of nodes $N$ and the maximum mobile node speed $\Delta$ on the average savings without random refresh rates ($R = \infty$) or beacons ($B = 0$).

### A. Maximum Mobile Node Speed $\Delta$ and Node Density

Initially, we evaluate the impact that the number of nodes $N$ (node density) and maximum speed $\Delta$ have on the attainable savings. We illustrate this impact in Figure 2 for a scenario without random background refreshes ($R = \infty$) and without fixed beacons ($B = 0$). We observe that for a low node density, e.g., $N \leq 10$, the savings are very low and almost linearly increasing with an increase in the number of nodes. As the node density increases, we observe a faster than linear increase in the potential savings, whereby lower nodal speeds correspond to higher attainable savings, up to approximately 18%. With an increased node population density, a more frequent availability of neighboring nodes with acceptable accuracy values allows a particular node to determine its position based on cooperation rather than utilizing the GPS. As nodes move faster, their location accuracy decreases as does the chance of being in range of other nodes, e.g., from $\Delta = 1$ to $\Delta = 3$. With an additional movement speed increase, the chance of finding suitable neighboring nodes decreases again, as nodes travel further.

We next illustrate the effect the cooperative exchange has on the average self-calculated node accuracy (SCA) and offset from real position (Off) for the same scenario in Figure 3. We initially note that the average self-calculated node accuracy is fairly close to the actual position and remains rather independent of the maximum random speed or number of nodes,

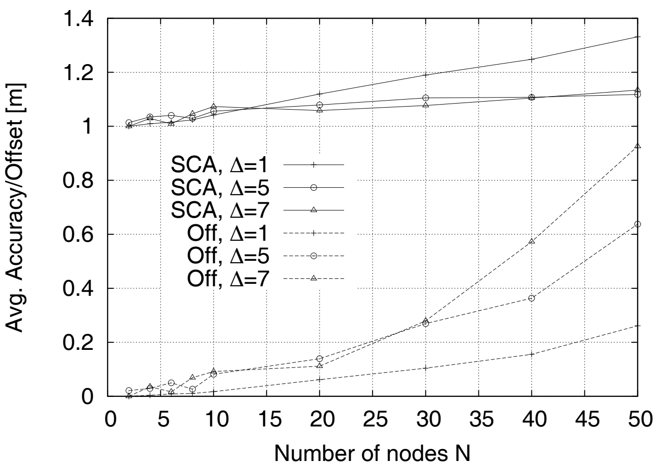


Fig. 3. Impact of the maximum mobile node speed $\Delta$ on the average self-calculated node accuracy (SCA) and externally calculated offset from real position (Off) for different number of nodes $N$ without random refresh rates ($R = \infty$) or beacons ($B = 0$).

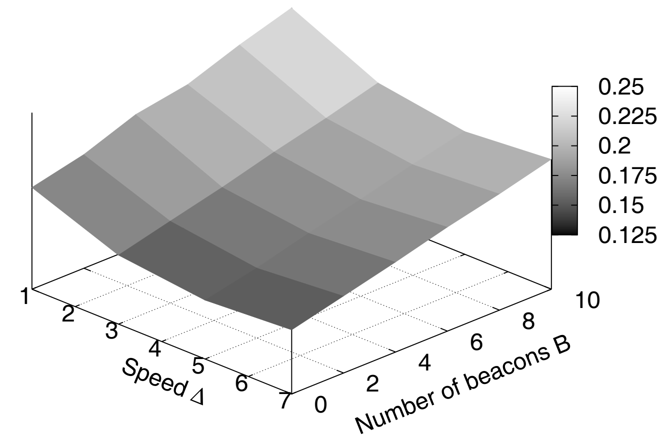


Fig. 4. Impact of the maximum mobile node speed $\Delta$ and different number of randomly placed beacons $B$ on the average savings for fixed number of nodes ($N = 50$) without random refresh ($R = \infty$).

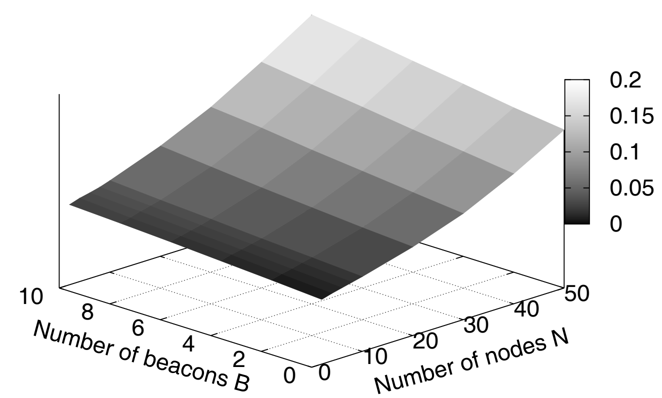


Fig. 5. Impact of the number of nodes $N$ and different number of randomly placed beacons $B$ on the average savings for fixed maximum mobile node speed ($\Delta = 5$) without random refresh ($R = \infty$).

increasing only for higher node densities. In other words, nodes assume that they have their location closely calculated and the low offset corroborates this observation. The overall achieved accuracy for low node densities is in the 1m range, which (when comparing with Figure 3 and the corresponding offsets) can be attributed to the frequent invocation of the GPS (and low savings in return). Together with an increase for higher densities, we observe an increase in the offset with the nodal speed, while the accuracy remains fairly high in the meter range. The increasing difference between actual and estimated position can be attributed to the more frequent exchange of inaccurate positions between nodes while a node's self-determined accuracy is only slowly increasing.

### *B. Random GPS Refresh Rate $R$*

We now shift our view to the impact that the random background refresh rate (e.g., GPS usage triggered by other applications if considering a cellphone) has on the overall savings. We find (with greater details omitted due to space limitations) that all evaluated background refresh rates, ranging from $R = \infty$ to 600 simulated seconds, yield similar average savings. If we regard the scenario outlined in Section III with a standard normal error distribution, we observe ranges from around 17% to around 24%, depending on the number of nodes and their maximum movement speed, but independent of the configured random refresh values. We conclude that the maximum background refresh rates have only a negligible impact on the savings obtainable.

### *C. Number of Fixed Beacons $B$*

We now evaluate the impact of randomly placed position beacons $B$ while maintaining the number of nodes ($N = 50$) in the absence of random refreshes ($R = \infty$) and illustrate the results in Figure 4. We initially observe the characteristic relationship between average savings and speed in the absence of beacons as previously noted. With an increase in the number of location beacons, however, we observe an increase in the overall level of the average savings. This increase in potential savings corresponds to an approximately linear increase of approximately 1% of average savings for each randomly placed beacon. The maximum average savings observed in turn rise to around 22.5%.

Next, we evaluate the impact of the number of nodes in the absence of random refreshes ($R = \infty$) for a maximum random speed of $\Delta = 5$ in Figure 5. We observe that the average savings increase slightly faster than linear with an increase of the number of nodes. We additionally note that the impact of a placed beacon is approximately linear as well, with approximately 0.5% – 1% increases in average savings, whereby the effect is greater with lower node densities.

Overall, we note that beacon placements have a directly observable linear effect on the average savings. This impact can be explained through the wandering of nodes which now are increasingly able to $(i)$ use these beacons to obtain a local location fix or $(ii)$ use these beacons to derive a more accurate

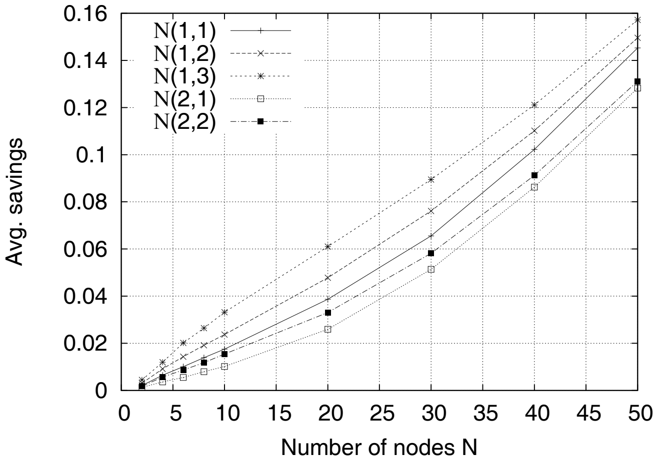


Fig. 6. Savings impact of the number of nodes $N$ and different estimation error distribution parameters for maximum mobile node speed ($\Delta = 5$) without random refresh ($R = \infty$).

position, which in either case results in less frequent uses of their simulated GPS.

### D. Impact of Range Estimation Errors

The previously presented results assumed a normal distribution of $N(1,1)$ for the individual range estimation processes within the simulation. We illustrate the impact of different normal distribution parameters on the overall savings in Figure 6. We observe that the savings increase with the variability of the introduced ranging error while they decline with an increase in the error average. This behavior can be explained in conjunction with the globally calculated average nodal offset from the actual position (omitted due to space constraints), which increases with the variability as well. In addition, the self-evaluated nodal position accuracy declines with the average estimation error, causing more calls to the simulated GPS.

### E. Small World Scenario

In smaller areas, such as office spaces, where device broadcasts could be simply tied to database entries of positions, we consider a considerably smaller evaluation area of 10m x 10m. Utilizing the initial random range estimation errors $N(M = 1, \sigma = 1)$, we combine those with a smaller desired accuracy of 3m.

With a single user and a single beacon, we derive savings around four percent (almost independent of movement speed), which increase to over 60% when regarding three beacons. We additionally note that in the absence of beacons and only depending on interchange with other simulated nodes, we observe worst-case (dependent on max. movement speed) savings over three percent for two nodes, over 10% for three, up to over 50% when regarding five nodes. Combining beacons and sharing nodes yields savings of 60–70%.

## V. Conclusion

We find that using a random walk model and sharing locations among nodes (with an exchange of self-estimated accuracies) results in significant savings of calls to a simulated GPS module. of above Considering 50 mobile nodes in a 100m $\times$ 100m area with lower movement speeds, we found around 18% savings without infrastructure beacons and 20% with beacons. For a smaller area of 10m $\times$ 10m, the savings can be significantly larger and beyond 40% when regarding four or more entities. These significant savings are only marginally impacted by random background location updates and more significantly impacted by node density, number of beacons, and speed. We find an almost linear relationship between the number of beacons and savings in our evaluated scenarios.

In future works, we will evaluate other mobility models in our simulations, as well as more randomized approaches to localization (which will deviate from a constant monitoring to fully random localization requests). Furthermore, we intend to perform evaluations on actual implementations to translate our findings here into actual power savings achievable.

## Acknowledgment

This work was supported in part by an Early Career grant from the Office of Research and Sponsored Programs at Central Michigan University.

## References


[1] P. Seeling, "Power consumption evaluation for cooperative localization services," in *Innovations and Advances in Computer, Information, Systems Sciences, and Engineering*, ser. Lecture Notes in Electrical Engineering, K. Elleithy and T. Sobh, Eds. Springer New York, 2013, vol. 152, pp. 267–275. [Online]. Available: http://dx.doi.org/10.1007/978-1-4614-3535-8_22

[2] H. Liu, H. Darabi, P. Banerjee, and J. Liu, "Survey of wireless indoor positioning techniques and systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 37, no. 6, pp. 1067–1080, 2007.

[3] F. Belloni, V. Ranki, A. Kainulainen, and A. Richter, "Angle-based indoor positioning system for open indoor environments," in *Proc. of the 6th Workshop on Positioning, Navigation and Communication (WPNC)*, Hannover, Germany, March 2009, pp. 261 –265.

[4] C. di Flora, M. Ficco, and S. Russo, "An architecture for providing java applications with indoor and outdoor hybrid location sensing," in *Proc. of the 4th IEEE Workshop on Software Technologies for Future Embedded and Ubiquitous Systems (SEUS)*, Gyeongju, Korea, April 2006, p. 6 pp.

[5] C. Sammarco, F. Fitzek, G. Perrucci, A. Iera, and A. Molinaro, "Localization information retrieval exploiting cooperation among mobile devices," in *Proc. of the IEEE International Conference on Communications Workshops*, Beijing, China, May 2008, pp. 149 –153.

[6] T. A. Johnson and P. Seeling, "Localization using bluetooth device names," in *Proc. of the 13th ACM international symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, Hilton Head Island, SC, USA, June 2012, pp. 247–248. [Online]. Available: http://doi.acm.org/10.1145/2248371.2248408

[7] A. C. Champion, B. Zhang, J. Teng, and Z. Yang, "D-card: A distributed mobile phone based system for relaying verified friendships," in *Proc. of the IEEE Conf. Computer Communications Workshop*, Shanghai, China, Apr. 2011, pp. 696–701.

[8] Bluetooth SIG, "Specification of the bluetooth system, core version 4.0," 2010.

[9] U. Wilensky, "Netlogo," Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL., USA, 1999. [Online]. Available: http://ccl.northwestern.edu/netlogo/