Performance of Collaborative GPS Localization in Pedestrian Ad Hoc Networks

Vladimir Vukadinovic
Disney Research Zurich
8092 Zurich, Switzerland
vvuk@disneyresearch.com

Stefan Mangold
Disney Research Zurich
8092 Zurich, Switzerland
stefan@disneyresearch.com

ABSTRACT
In ad hoc networks without static nodes that could be used as reference points, mobile handhelds must rely on their GPS receivers to enable location-aware services. By sharing their position estimates using short range radios, neighboring devices may suppress unnecessary GPS activations in order to reduce energy consumption. We describe and evaluate two collaborative GPS localization protocols based on substitution and averaging of position estimates. The evaluation focuses on entertainment park scenarios and relies on realistic simulations to capture the mobility of park visitors. We demonstrate that the simple collaboration protocols, which do not require distance estimation between the neighbors, may provide significant energy savings. We discuss the impact of device density and provide guidelines for choosing the transmission range of their radio interfaces.

Categories and Subject Descriptors
C.2.1 [**Computer–Communication Networks**]: Network Architecture and Design–*Wireless communication.*

General Terms
Algorithms, Performance, Experimentation.

Keywords
GPS localization; energy consumption; ad hoc networks; mobility.

1. INTRODUCTION
Unlike in infrastructure-based wireless networks, communication in ad hoc networks relies on dynamically-created multi-hop routes composed of direct links between neighboring nodes, which forward data on behalf of each other. This communication mode is useful when network infrastructure (cellular base stations, WLAN access points) is not available or cannot be used due to the cost or logistic reasons. It can also be used to supplement sparsely deployed infrastructure networks with partial coverage. Examples of ad hoc network architectures include mobile ad hoc networks (MANETs) and delay/disruption tolerant networks (DTNs). Today, the use of the ad hoc mode on handheld devices, such as smartphones, is hindered by the high energy consumption of Wi-Fi radios, the short range and slow pairing procedure of Bluetooth radios, the complexity of ad hoc network setup, and lack of support in popular operating systems (e.g. iOS and Android). However, as Wi-Fi Direct [1] and Bluetooth 4.0 (aka Bluetooth Smart) [2] are starting to penetrate the market, it is likely that such obstacles will diminish. Nokia's Instant Community [3], Apple's iGroups [4] currently rely on proprietary solutions to enable ad hoc communication between smartphones.

Depending on the density and mobility of nodes, an ad hoc network may exhibit varying degrees of partitioning and, therefore, cannot provide stringent quality of service guarantees. However, if carefully designed, a variety of application can be provided in such networks. For example, ad hoc networks can support mobile multi-player games, mobile advertising, multimedia sharing, and participatory sensing. Some of these applications are location-based and require knowledge of user's location with different levels of accuracy. In infrastructure-based networks, the location can be determined using multilateration algorithms, using infrastructure nodes at known locations as reference points. In ad hoc networks, static infrastructure nodes are not available or they are too sparse to be used for multilateration. Without WLAN access points, GSM/UMTS cell towers can only provide very coarse localization, with errors as large as 300 m. GPS localization becomes often the only way to localize a handheld device. However, GPS is power hungry: A continuously active GPS receiver alone may drain a battery on a smartphone in a few hours. Our results show that its consumption is significant even if sampled every few minutes.

In this paper, we consider collaboration between neighboring devices to reduce the energy consumption of GPS receivers. A device may combine position information received from the neighbors with its own to obtain a new position estimate without relying on GPS. Only if a position estimate with acceptable error/confidence cannot be obtained from the neighbors, a device will trigger its GPS receiver. We refer to this type of localization as *collaborative GPS localization*. The collaboration may not only reduce the GPS energy consumption, but it may also increase the positioning accuracy: Even static and collocated devices often obtain different position information from GPS due to multipath effects and/or because their receivers have different sensitivities and lock-on different sets of satellites. If the position information is shared between the devices, each device may refine its original position estimate. However, it is not straightforward to conclude that collaborative GPS localization is always beneficial. While it may reduce the consumption of GPS receivers, the collaboration increases the consumption of wireless interfaces. Also, if a GPS receiver is not sampled regularly, it may require a new satellite lock-on procedure, which is a lengthy and power consuming procedure. Furthermore, the overall energy consumption and accuracy of collaborative GPS localization depends strongly on the density and mobility of devices in an area and on the transmission range of their radios.

Our target scenarios assume large crowds of people/tourists visiting areas such as entertainment theme parks, zoos, open-air

MobiOpp'12, March 15–16, 2012, Zürich, Switzerland.

archeological parks, or nature reserves. A number of location-based services offered to the visitors rely on wireless communication [5]. However, it cannot be assumed that such areas are fully covered with a wireless (e.g. Wi-Fi) infrastructure. Rolling out extensive infrastructure in a theme park, for example, is not an easy task: The largest parks are comparable in size with big cities (Walt Disney World Resort in Florida spans over ~100 km^2). Problems go beyond the obvious deployment and maintenance costs. For example, access points and antennas may be too visible to guests and, therefore, interfere with artistic intentions. Therefore, services must rely on spotty Wi-Fi coverage (if any), ad hoc communication between visitors' devices, and GPS localization. We describe a protocol for collaborative GPS localization that decides when a device should provide/request location information to/from its neighbors. It then calculates a new position estimate based on the input from the neighbors. We explore if such protocol can be used to reduce the energy consumption of GPS receivers. Our evaluation is based on detailed simulations, where the mobility of people/devices is driven by mobility traces collected in a theme park [5].

The reminder of this paper is organized as follows: Section 2 provides an overview of related work. Energy consumption models for GPS receivers and wireless interfaces are described in Section 3. Section 4 describes methods to measure and track position errors on smartphones and introduces an error model used in our simulations. Section 5 presents our collaborative localization protocols. The simulation setup and performance results are presented in Sections 6 and 7, respectively. Section 8 concludes the paper.

2. RELATED WORK

Collaborative (cooperative) localization has been first proposed for robot and sensor networks. Only recently it has been considered for other types of wireless networks. A general overview of cooperative localization techniques is provided in [6]. In robotics literature, collaborative localization refers to the problem of fusing relative position measurements between mobile robots with their odometry measurements. Most of the proposed solutions require centralized processing [7][8], although distributed algorithms have also been proposed [9].

In sensor networks, cooperating devices are typically densely deployed and static. A subset of nodes may be equipped with GPS receivers and/or able to obtain their absolute locations by other means. These devices serve as anchors for other devices, which rely on ranging and multilateration to determine their locations. Most of the solutions proposed for static sensor networks are not applicable to mobile networks due to frequent changes in topology. Therefore, [10] proposes a collaborative localization scheme by which a mobile sensor node may refine its location estimate through sporadic encounters with other nodes. In [11], the authors evaluate the benefits of using radio ranging and location sharing among GPS-enabled sensors mounted on cow collars for cattle tracking and virtual fencing applications. Two collaborative localization algorithms that rely on accurate ranging between neighbors using ZigBee and UWB radios have been described in [12]. Similarly, [13] considers cooperative WLAN-based indoor positioning for groups of people moving in clusters. ZigBee radios are used for proximity detection and communication within the clusters.

In sensor networks, energy-efficient communication between cooperating nodes can be achieved by means of ultra-low power radios. However, if communication relies on radios available on commodity phones, such as Wi-Fi and Bluetooth radios, the collaboration may incur significant energy overhead. Cooperative positioning for Wi-Fi devices is considered in [14]. Received signal strength (RSS) measurements towards access points, which are used as reference points, are supplemented with RSS measurements towards neighboring mobile terminals and used as inputs to an algorithm that calculates the position. A similar scheme is proposed and evaluated in [15]. In [16], the authors consider collaboration among mobile nodes capable of localizing themselves using either GPS or pedestrian dead reckoning. The focus in [14]-[16] is on localization accuracy; energy consumption is not addressed. Simulations in [15], [16] employ simple random walk mobility models. Our work is closely related [17], which considers Bluetooth communication to synchronize GPS positions of neighboring devices to reduce the number of GPS activations. To evaluate the benefits of the proposed Bluetooth-based Position Synchronization (BPS) protocol, two phones were placed in a bag and carried around for two days. Therefore, both phones had the same GPS signal availability and were constantly within each other's range. Depending on the mobility and density of devices, this simplistic evaluation could underestimate or overestimate the potential energy savings of the protocol in real-world scenarios.

Several works have proposed to use accelerometers, compasses, microphones, and other low-power sensors on phones [17]-[21] as well as the knowledge of habitual mobility [22] to adapt the duty cycles and suppress unnecessary activations of GPS receivers. Such approaches are complementary to the collaborative GPS localization and can be used to further reduce the energy consumption. However, they can be very unreliable: For example, theme park visitors engage in many activities that activate the accelerometers (e.g. ride a roller-coaster), but do not result in mobility that needs to be tracked.

3. ENERGY CONSUMPTION

In this section, we describe models of the energy consumption of GPS receivers and wireless interfaces, which we later use to evaluate the performance of collaborative GPS localization.

3.1 Energy Consumption of GPS Receivers

If sampled continuously, a GPS receiver may quickly drain a battery on a mobile phone [17]. An obvious strategy to reduce the energy consumption is to periodically sample ("duty cycle") the receiver. The energy consumption of a periodically sampled GPS receiver depends on the time needed to obtain a location fix, which depends on the sampling interval. We carried out a number of experiments to measure the energy consumption of continuous and periodic GPS sampling. For the measurements, we used HTC Desire smartphones running Android v2.2.

First, we measured the energy consumption of continuous GPS sampling under various satellite visibility conditions. The consumption is measured based on the electric current drained from the battery, whose milliampere value is obtained from the battery driver. In each test, we calculated the average power draw over a one-hour period. Regardless of the satellite visibility, this average power draw P_{GPS} was close to 305 mW (~78 mA @ 3.9 V) on top of the measured base consumption of the phone. This is consistent with the 80 to 85 mA measured on Android phones in [23] and somewhat less than 370 mW on Symbian Nokia N95 phone in [17].

Second, we measured the time-to-fix (TTF) for various GPS sampling intervals. Before each test, up-to-date satellite almanac and ephemeris data was downloaded using Wi-Fi connection. Hence, whenever sampled, phones' GPS receivers performed so-called "hot start" with valid satellite information. After a hot start, TTF depends on how fast a GPS receiver can tune to the carrier frequencies of visible satellites and synchronize with their signals. The carrier frequencies are constantly shifted due to the Doppler effect. In environments with lots of shadowing and multipath scattering, it may take a few tens of seconds to acquire a fix. The measurements were performed in one of the Disney's theme parks, where phones were carried at the walking speed of a typical theme park visitor. TTFs were measured for four sampling intervals T_S (30, 60, 120, and 300 seconds). The sampling interval is the time elapsed since the last fix was obtained until the next one is requested. The TTFs for each sample, as well as the average TTFs for each of the sampling intervals are shown in Fig 1. Based on the results, we construct a model that describes the average TTF as a function of the sampling interval T_S:

$$\mathrm{TTF}(\mathrm{T_S}) = \begin{cases} 10, & \mathrm{T_S} \le 30 \\ 10 + 8 \cdot \log_{10}(\mathrm{T_S}/30), & 30 < \mathrm{T_S} < 120 \\ 15, & \mathrm{T_S} \ge 120 \end{cases} \quad (1)$$

The function $\mathrm{TTF}(\mathrm{T_S})$ is also plotted in Fig. 1. The model (1) assumes typical pedestrian walking speed. We performed another set of measurements where phones were static. As expected, TTFs were significantly shorter because Doppler and multipath effects were less prominent. Based on the results, which we omit for brevity, we construct a model that describes the average TTF when phones are static:

$$\mathrm{TTF}(\mathrm{T_S}) = \begin{cases} 4, & \mathrm{T_S} \le 30 \\ 4 + 2 \cdot \log_{10}(\mathrm{T_S}/30), & 30 < \mathrm{T_S} < 300. \\ 6, & \mathrm{T_S} \ge 300 \end{cases} \quad (2)$$

The energy consumed to acquire a fix is then given by $E_{FIX}(T_S) = P_{GPS} \cdot TTF(T_S)$. Although models in (1) and (2) are not thoroughly validated (i.e. by comparing results from various theme parks), we believe that they better reflects the reality of theme park scenarios than models described in literature. For example, the model in [18] assumes a constant TTF of six seconds for sampling intervals longer than 30 seconds.

3.2 Energy Consumption of Wireless Interfaces

Collaborative GPS localization relies on ad hoc communication between neighboring devices using their wireless (e.g. Wi-Fi) interfaces. This communication incurs energy overhead. The overhead includes the energy spent to transmit/receive location information to/from the neighbors. We assume that the energy spent for idle listening is not a part of this overhead because, in ad hoc networks, devices anyway need to listen for incoming traffic from their neighbors. In the following, we assume that devices are equipped with Wi-Fi interfaces, which are nowadays available on most phones. To calculate the energy overhead in our simulations, we adopt the per-packet energy consumption model introduced and validated in [24]. The model assumes that the energy spent on top of idle listening (*energy*$^+$) to send or receive a packet in ad hoc mode is given by

$$energy^+ = m \times size + b,$$

where *size* is measured in bytes. Hence, the consumption has a fixed component associated with the power state changes and channel acquisition, and an incremental component, which is proportional to the size of the packet. The values of coefficients *m* and *b* depend on whether a packet is broadcasted or sent point-to-point. Table 1 summarizes the values of coefficients *m* and *b* measured in [24]. As an example, in the second-to-last column of the table, we calculate the energy consumption (on top of idle listening) to send/receive a 100-byte packet. The total energy consumption, assuming idle listening power of 741 mW, is given in the last column of the table. It is in the order of a few hundreds of microjoules. We use the model in our simulations to make a rough estimate of the energy consumed by a Wi-Fi interface whenever it transmits or receives one of the protocol messages of the collaborative GPS localization protocol. In practice, the consumption depends on particular Wi-Fi chipset, transmission rate, and other factors that we do not consider: The goal is to indicate general orders of magnitude.

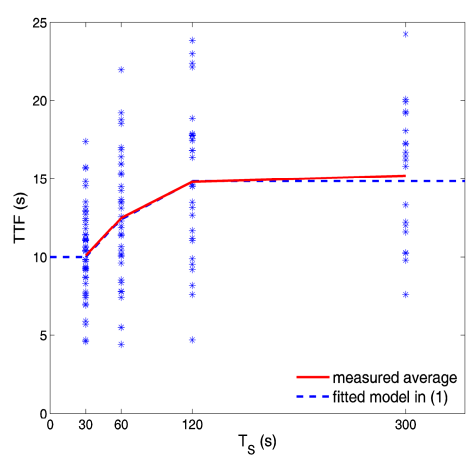

Figure 1. TTFs measured for various sampling intervals on phones carried by theme park visitors.

4. POSITION ERROR MODEL

Let us first assume that a mobile handheld device is sampling its GPS receiver to position itself without help from neighboring devices. The device maintains its position estimate $p = (x, y)$, which is updated with each new GPS sample. Our position error model for this scenario takes into account two sources of errors: position uncertainty of the last GPS update and the distance traveled since the update. Let τ be the time elapsed since the last position update (τ is the age of the position estimate p). Let $e(\tau)$ be the position error of p with respect to the current true position $P(\tau) = (X(\tau), Y(\tau))$:

$$e(\tau) = |P(\tau) - p| = \sqrt{(X(\tau) - x)^2 + (Y(\tau) - y)^2}$$

The model assumes that the expected position error $\mathrm{E}[e(\tau)]$, which measures the uncertainty of position p is given by

$$\mathrm{E}[e(\tau)] = \mathrm{E}[e(0)] + \tau \cdot \overline{v(\tau)}. \quad (3)$$

where $\mathrm{E}[e(0)] = \mathrm{E}[e_{GPS}]$ is the expected horizontal position error of the GPS update and $\overline{v(\tau)}$ is the estimated average speed of the phone during the period $(0, \tau)$. Ways to estimate $\mathrm{E}[e_{GPS}]$ and $\overline{v(\tau)}$ are discussed in the following. Note that (3) may overestimate the actual error since the displacement of the phone (with respect to its position at $\tau = 0$) depends on the movement trajectory and it is often smaller than $\tau \cdot \overline{v(\tau)}$.

Table 1. Wi-Fi energy consumption measurements in [24] assume data rate of 11 Mb/s. Idle listening power is 741 mW.

	m (µJ/byte)	b (µJ)	energy$^+$ 100 bytes (µJ)	tot. energy 100 bytes (µJ)
point-to-point send	0.48	431	479	533
broadcast send	2.1	272	482	536
point-to-point receive	0.12	316	328	382
broadcast receive	0.26	50	76	130

4.1 Position Error of GPS

The horizontal position error of GPS (e_{GPS}) depends on many factors: number and constellation of visible satellites, satellite clock and ephemeris data errors, atmospheric propagation delay, multipath fading, and GPS receiver quality. It can be written as

$$e_{GPS} = \sqrt{e_{xGPS}^2 + e_{yGPS}^2}$$

where e_{xGPS} and e_{yGPS} are random variables that correspond to the errors in x (longitude) and y (latitude) directions, respectively. The two random variables are often assumed to follow a normal distribution with zero mean, which has been confirmed in [25]. If we assume the same variances in both directions, then the horizontal error e_{GPS} is Rayleigh distributed:

$$P(e_{GPS} \leq \varepsilon) = 1 - e^{-\frac{e^2}{2\sigma^2}} = 1 - e^{-\frac{e^2}{\sigma_{rms}^2}},$$

where $\sigma_{rms} = 2/\sqrt{\pi} \cdot \mathrm{E}[e_{GPS}]$ is the root-mean-squared (RMS) error. When $\varepsilon = \sigma_{rms}$, then $P(e_{GPS} \leq \varepsilon) = 1 - e^{-1} \approx 0.63$. Hence, 63% of errors fall within a circle of radius σ_{rms}. Therefore, the RMS error σ_{rms} is often referred to as "63% error distance" and denoted by $dRMS$. This is the value what most GPS receivers estimate and report as their accuracy. Some receivers report "95% error distance" denoted by $2dRMS$. There is a simple relationship between the two measures: $2dRMS \approx 1.73 \cdot \sigma_{rms} \approx 1.95 \cdot E[e_{GPS}]$. Hence, $\mathrm{E}[e_{GPS}]$ can be easily estimated from the accuracy reported by GPS receivers. In our simulations, $\mathrm{E}[e_{GPS}]$ is drawn from a GPS dataset, which is described in Section 6.

4.2 Speed Estimation

Accurate estimation of $\overline{v(\tau)}$ on a phone is not a trivial problem. One option would be to assume that $\overline{v(\tau)}$ is equal to the speed reported by the GPS receiver at $\tau = 0$. GPS receivers typically estimate the speed from their position samples and from the measured Doppler shift of satellite signals. It has been shown, however, that these estimates are very unreliable at pedestrian speeds [18]. Another possibility is to adopt an upper bound on the walking speed, which may be $v_{max} = 1.5$ m/s for a vigorous park visitor, and assume that $\overline{v(\tau)} = v_{max}$. Clearly, this may grossly overestimate the expected position error $\mathrm{E}[e(\tau)]$ in (3). Inertial sensors on the phone, such as the accelerometer, can be used to refine the estimate. Readings from the accelerometer can be interpreted (e.g. based on a threshold crossing) as a binary indicator if a person is moving or not. The speed $\overline{v(\tau)}$ is then estimated as a product of v_{max} and the average value of the binary indicator over the period $(0, \tau)$. Evaluation of different methods to estimate $\overline{v(\tau)}$ is out of the scope of this paper. In our simulations, we assume that $\overline{v(\tau)}$ is estimated from the distance between the last two position updates and the time elapsed between them. The drawback of this method is that positioning errors introduce errors in the speed estimate. We put a lower cap on $\overline{v(\tau)}$ to 0.2 m/s. Otherwise, location updating would stall when $\overline{v(\tau)} = 0$.

5. COLLABORATIVE LOCALIZATION

Our collaborative localization protocol works as follows: Each mobile phone i maintains an up-to-date position error estimate $\mathrm{E}[e_i(\tau_i)]$ according to (3). If its error estimate exceeds the maximum tolerable position error e_{max}, the phone $i = 0$ broadcasts a *location update request* using its wireless interface. The request contains the phone's last position estimate $p_0 = (x_0, y_0)$ and position error estimate $\mathrm{E}[e_0(\tau_0)]$. Every neighbor i within the transmission range r who receives the request will estimate its distance d_i to the sender and compare the sender's error estimate $\mathrm{E}[e_0(\tau_0)]$ to its own error estimate $\mathrm{E}[e_i(\tau_i)]$. If $\mathrm{E}[e_i(\tau_i)] + d_i < \mathrm{E}[e_0(\tau_0)]$, the neighbor i will send a *location update response* with its position and position error estimates p_i and $\mathrm{E}[e_i(\tau_i)]$. Hence, a neighbor will respond to the request if it has a more accurate position estimate that the originator of the request, taking into account the distance between the two. The distance d_i can be estimated using received signal strength (RSS) or time of flight (TOF) based ranging techniques. RSS ranging is supported on commodity phones, but it suffers from low accuracy. TOF ranging may provide more accurate distance estimation, but typically requires changes to the phones' hardware and/or protocol stack. A software-based TOF ranging technique for Wi-Fi is described in [26]. Unfortunately, it can only be implemented on phones with reconfigurable open-source Wi-Fi drivers, which are presently rare. We focus on scenarios where the transmission range r is relatively small compared to the target location accuracy e_{max}. In such scenarios, ranging capabilities are not essential. In our protocol, neighboring phones assume that $d_i = r$, which accounts for the worst-case scenario.

Assume now that the phone obtains *location update responses* from N neighbors:

- If $N = 0$, the phone triggers its GPS receiver and waits for a position fix. After it obtains a fix, the phone updates its position estimate p_0, resets the age of the position estimate to $\tau_0 = 0$, calculates the expected position error $\mathrm{E}[e_0(0)]$ according to (3), and broadcasts a *location update notification* containing p_0 and $\mathrm{E}[e_0(0)]$ to its current neighbors.

- If $N > 0$, the responses from multiple neighbors are combined in order to update the phone's position estimate p_0. We consider the following two combining schemes:

Substitution: Position estimate p_0 is substituted with p_k, where

$$k = \arg\min_{1 \leq i \leq N} (\mathrm{E}[e_i(\tau_i)] + r)$$

Hence, among the neighbors' positions p_i, $1 \leq i \leq N$, the one with the smallest expected error is used as a new position estimate p_0. Correspondingly, the new position error is

$$\mathrm{E}[e_0(0)] = \mathrm{E}[e_k(\tau_k)] + r.$$

Averaging: position p_0 is calculated as a weighted average of $p_i(x_i, y_i)$, $1 \leq i \leq N$:

$$p_0 = \left(\sum_{i=1}^{N} w_i x_i, \sum_{i=1}^{N} w_i y_i\right), \tag{4}$$

where weights w_i are chosen so to minimize the expected squared error σ^2 of the position estimate p_0 with respect to the current true position $P_0(\tau_0)$:

$$\sigma^2 = \mathrm{E}[|P_0(\tau_0) - p_0|^2] =$$

$$= \mathrm{E}\left[\left(X_0(\tau_0) - \sum_{i=1}^{N} w_i x_i\right)^2 + \left(Y_0(\tau_0) - \sum_{i=1}^{N} w_i y_i\right)^2\right].$$

Assuming that $(X_0(\tau_0) - x_i, Y_0(\tau_0) - y_i)$, $1 \leq i \leq N$ are uncorrelated random variables, it is easy to show that σ^2 is

$$\sigma^2 = \sum_{i=1}^{N} w_i^2 \sigma_i^2, \tag{5}$$

where $\sigma_i^2 = \mathrm{E}[|P_0(\tau_0) - p_i|^2] = \mathrm{E}[(X_0(\tau_0) - x_i)^2 + (Y_0(\tau_0) - y_i)^2]$ is the expected squared error of p_i with respect to $P_0(\tau_0)$. From (5), we obtain that σ^2 is minimized for

$$w_i = \frac{1/\sigma_i^2}{\sum_{k=1}^{N} 1/\sigma_k^2} \tag{6}$$

However, w_i cannot be calculated from (6) because σ_i^2, $1 \leq i \leq N$ are unknown. In order to express w_i in terms of $\mathrm{E}[e_i(\tau_i)]$, we use the following approximations: Since $|P_0(\tau_0) - p_i| = |P_0(\tau_0) - P_i(\tau_i) + e_i(\tau_i)|$ and $|P_0(\tau_0) - P_i(\tau_i)| \approx r$, it follows

$$\sigma_i^2 \approx \mathrm{E}[(e_i(\tau_i) + r)^2] =$$
$$= \mathrm{E}[(e_i(\tau_i) + r - \mathrm{E}[e_i(\tau_i) + r])^2] + \mathrm{E}[e_i(\tau_i) + r]^2 =$$
$$= \mathrm{E}[e_i(\tau_i) + r]^2 \cdot \left(\frac{\mathrm{E}[(e_i(\tau_i) + r - \mathrm{E}[e_i(\tau_i) + r])^2]}{\mathrm{E}[e_i(\tau_i) + r]^2} + 1 \right) =$$
$$= (\mathrm{E}[e_i(\tau_i)] + r)^2 \cdot (\gamma_i^2 + 1)$$

where γ_i is the ratio of the standard deviation and the mean of $e_i(\tau_i) + r$. Given that $e_i(\tau_i)$, $1 \le i \le N$ follow the same distribution, their γ_i ratios are equal ($\gamma_i = \gamma$). By substituting σ_i^2 in (6), we obtain

$$w_i = \frac{\dfrac{1}{(\mathrm{E}[e_i(\tau_i)] + r)^2}}{\sum_{k=1}^{N} \dfrac{1}{(\mathrm{E}[e_k(\tau_k)] + r)^2}}$$

Now the weighted average position p_0 can be calculated by substituting w_i in (4). Starting from (5) and following the same reasoning, it can be shown that the expected error $\mathrm{E}[e_0(0)]$ is

$$\mathrm{E}[e_0(0)] = \sqrt{\sum_{i=1}^{N} w_i^2 (\mathrm{E}[e_i(\tau_i)] + r)^2}.$$

Once p_0 and $\mathrm{E}[e_0(0)]$ are determined, either by substitution or by averaging, a *location update notification* is broadcasted to the neighbors, which in turn update their position estimates p_i if $E[e_0(0)] + r < E[e_i(\tau_i)]$.

6. EVALUATION SETUP

We analyze and compare the energy consumptions of non-collaborative and collaborative GPS localization of theme park visitors using simulations. The scenario assumes that visitors' phones maintain estimates of their current position errors $\mathrm{E}[e(\tau_i)]$ defined in (3) and, when the error exceeds e_{max}, they either trigger their GPS receiver (in case of non-collaborative localization) or execute the protocol described in Section 5 (in case of collaborative localization). We measure the number of GPS activations, number of protocol messages exchanged between the phones, and the deviation of the position estimates from the ground truth positions.

The performance of collaborative localization depends strongly on the density and mobility of the visitors/phones. The density affects the number of potential collaborators. The mobility affects the position errors $\mathrm{E}[e(\tau_i)]$ and the frequency of collaboration opportunities. In our simulations, density and mobility of visitors are driven by real-world data: As a part of a research study unrelated to this paper, we collected 910 GPS traces by handing out GPS-enabled phones to the visitors of the Epcot theme park in Florida [5]. The layout of the park is shown in Fig. 2 (left). The phones were handed out between 8am and 1pm at the entrance gate, and collected when the visitors were exiting the park. The spatial distribution of the phones at different times of the day is shown in Fig. 2 (right). The bell-shaped curve in Fig. 3 shows the number of phones in the park as a function of time. The phones were sampling their GPS receivers on average every two minutes. In addition to the geo-coordinates, GPS accuracy was also logged. We discarded waypoints whose accuracy was worse than 25 m. We also discarded tracks shorter than two hours or containing less than 50 waypoints. Results presented in the following sections are based on the remaining 825 out of 910 tracks. We interpolated the movements of visitors between the waypoints assuming straight-line movements. The interpolated trajectories are used as "ground truth" location data to drive the mobility of nodes in our simulations. The original dataset, which includes waypoints with accuracy is worse than 25 m, is used to generate the expected horizontal position error $\mathrm{E}[e_{GPS}]$ of the simulated GPS receivers.

Whenever a simulated node triggers its GPS receiver, the time-to-fix (TTF) and the expected error of the fix $\mathrm{E}[e_{GPS}]$ are generated: TTF is generated based on the time T_S elapsed since the receiver was sampled last time and the visitor's average *ground-truth* walking speed during that time: If the speed is higher than 0.2 m/s, TTF is generated according to (1). Otherwise, it is generated according to (2). The energy consumption of the GPS receiver E_{GPS} increases by $E_{FIX}(T_S) = P_{GPS} \cdot TTF(T_S)$ with every new sample. $\mathrm{E}[e_{GPS}]$ is drawn from the empirical distribution observed in the traces. The average 95 % error distance in the traces is $\overline{2dRMS} = 21.3$ m, which corresponds to the average $\overline{\mathrm{E}[e_{GPS}]} = \overline{2dRMS}/1.95 \approx 11$ m. The phones were equipped with GPS chipsets based on Qualcomm's gpsOne technology, which is world's most widely used GPS technology for handsets [27].

We also keep track of the number of protocol messages (location update requests, location update responses, and location update notifications) exchanged between the nodes. In order to estimate the energy overhead of a Wi-Fi interface E_{WiFi}, we introduce the following simplifications:

- The size of each protocol message is 100 bytes at the physical layer. This is more than sufficient to accommodate

Figure 2. Left: The layout of the Epcot theme park in Florida. Right: spatial distribution of park visitors at 11am, 13pm, 16:30pm, and 17:30pm (clockwise starting from the top left image).

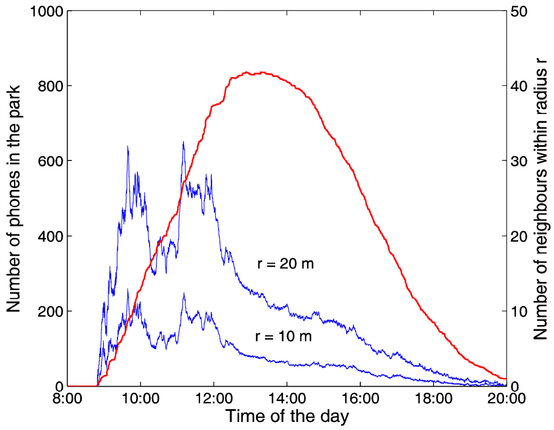

Figure 3. Number of phones and the average number of neighbors within radius *r* at different times of the day.

location information payload (position estimate p_i and position error estimate $\mathrm{E}[e_i(\tau_i)]$) and all upper-layer headers.

- Sending or receiving a protocol message consumes 600 µJ of energy, regardless if whether the message is sent/received in broadcast mode (location update request, location update notification) or unicast mode (location update response). According to Table 1, this overestimates the actual consumption of Wi-Fi interfaces.
- The energy consumed to send a protocol message does not depend on the transmission range *r*. It has been shown in [28] that transmit power has a minor impact on the energy consumption of Wi-Fi interfaces.

Therefore, the energy consumption of a Wi-Fi interface E_{WiFi} increases by 600 µJ whenever it sends or receives a protocol message. The total energy consumption is given by the sum of E_{GPS} and E_{WiFi} for the collaborative GPS localization, and by E_{GPS} only for the non-collaborative GPS localization.

7. PERFORMANCE RESULTS

We present the performance results for non-collaborative and collaborative GPS localization (with position substitution and averaging) for various values of the maximum tolerable position error e_{max} and transmission range *r*. Each simulated node, whose arrival/departure and mobility in the park are driven by one of the ground-truth traces, maintains the following statistics: number of GPS activations, energy consumed by the GPS receiver, number of sent/received protocol messages, energy consumed by the Wi-Fi interface, and deviation of the position estimate from the ground-truth position. At the end of each simulation run, we calculate for each node the average number of GPS activation and protocol messages per hour, average GPS and Wi-Fi energy consumption per hour, and average deviation from the ground-truth position. These values are then averaged over all devices and shown in Table 2.

In case of non-collaborative localization, the number of GPS activations decreases from 59.4/h to 12.4/h as e_{max} increases from 25 m to 75 m, and so does the energy consumption, although the energy per activation increases (3.6 J/activation for $e_{max} = 25$ m vs. 4.5 J/activation for $e_{max} = 75$ m). This is because TTF increases when GPS is sampled less frequently. The average consumption of GPS receivers for $e_{max} = 25$ m is 213.6 J/h. For a comparison, a smartphone in the suspended state consumes ~100 J/h (HTC Dream: 26.6 mW or 95.6 J/h, Google Nexus One: 24.9 mW or 89.6 J/h) [29]. When e_{max} increases to 75 m, the GPS consumption drops to 56 J/h, which is however still significant. Surely, this consumption is dwarfed by the idle listening consumption of Wi-Fi, but this is likely to change (e.g. with the use of Power Save Mode and Opportunistic Power Save protocol in Wi-Fi Direct). Table 2 also shows that the average deviation from the ground-truth position is well below e_{max}.

Collaborative localization based on position substitution with $r = 10$ m reduces the number of GPS activations and, therefore, the GPS energy consumption. The reduction depends on e_{max}. For $e_{max} = 25$ m, the GPS consumption is 151 J/h, which is 71% of 213 J/h consumed by non-collaborative localization. For $e_{max} = 75$ m , it decreases to 24.3 J/h, which is 43% of 56 J/h consumed by non-collaborative localization. The energy overhead of Wi-Fi is negligible (< 1 J/h). The average deviation from the ground-truth position is almost the same as with non-collaborative localization. The results show that collaboration significantly reduces the energy consumption, especially when the maximum

Table 2. Performance results for non-collaborative and collaborative GPS localization for various values of e_{max} and r. The results are averaged over time spent in the park and over all devices.

GPS localization method	r (m)	e_{max} (m)	GPS activations (1/h)	GPS energy (J/h)	Protocol messages (1/h)	Wi-Fi energy (J/h)	GPS+Wi-Fi energy (J/h)	Deviation (m)
non-collaborative	n.a.	25	59.4	213.6	n.a.	n.a.	213.6	9.9
		50	21.7	92.3	n.a.	n.a.	92.3	16.4
		75	12.4	56.0	n.a.	n.a.	56.0	23.0
collaborative (substitution)	10	25	43.5	151.1	983.4	0.6	151.7	9.8
		50	11.6	44.8	323.3	0.2	45.0	16.5
		75	5.9	24.3	176.6	0.1	24.4	23.6
	20	25	64.6	220.9	4300.8	2.6	223.5	10.3
		50	11.7	42.0	866.0	0.5	42.5	16.9
		75	4.5	17.3	515.9	0.3	17.6	24.3
collaborative (averaging)	10	25	41.0	143.8	764.3	0.5	144.3	9.8
		50	10.2	40.4	196.2	0.1	40.5	16.8
		75	5.0	20.9	111.8	0.1	21.0	25.3
	20	25	59.4	205.6	3608.4	2.2	207.8	10.2
		50	9.5	36.2	490.6	0.3	36.5	17.6
		75	3.1	12.6	201.1	0.1	12.7	26.2

tolerable error increases. With an optimal selection of the transmission range, the energy consumption can be further reduced. The optimal selection depends on an the interplay between r and e_{max}: On one hand, a larger range increases the number of neighbors and, therefore, the chance to obtain a location estimate without GPS. On the other hand, the accuracy of position estimates obtained from the neighbors decreases (since every neighbor is assumed to be r meters away) and might not be sufficient to suppress GPS activations if r approaches e_{max}. The results for $r = 20$ m show that collaboration actually increases the energy consumption when $e_{max} = 25$ m. However, when $e_{max} = 75$ m, the consumption is only 31% of 56 J/h consumed by non-collaborative localization, down from 43% with $r = 10$ m. Results achieved with the position averaging algorithm are also shown in Table 2. This algorithm utilizes location information from multiple neighbors to obtain more accurate position estimates. Therefore, it further reduces the number of GPS activations, hence, the energy consumption. The results from Table 2 are summarized in Fig. 4. The figure shows that 1) the energy savings of collaborative localization increase with e_{max}, 2) the averaging algorithm provides additional savings compared to the substitution algorithm, 3) the optimal transmission range r depends on e_{max}, and 4) even for ranges that are marginally smaller than e_{max} (e.g. $r = 20$ m, $e_{max} = 25$ m) collaborative localization may provide energy savings without distance estimation between neighbors.

The results presented so far show the time-average performance for a random device for the entire duration of its stay in the park. A typical visit to the Epcot lasts 5-6 hour. During this time, the number of visitors and their spatial distribution change, and so does the number of neighbors within the radius r, as shown in Fig. 3. Notice that the number of neighbors is weakly correlated with the number of devices in the park because visitors tend to gather at certain locations in the park at certain times of the day. For example, in the morning hours, they crowd in the front section of the park, as shown in Fig. 2 (right). Later they may gather to watch a street performance or similar event. Once they spread across the park area in the late afternoon, the correlation becomes stronger. To show the impact of the number of neighbors on the performance of collaborative localization, we observe three one-hour periods listed in Table 3 when the number of neighbors is relatively constant. Energy consumption of collaborative localization based on position averaging for the three periods and various e_{max} is shown in Fig. 5 as a percentage of energy

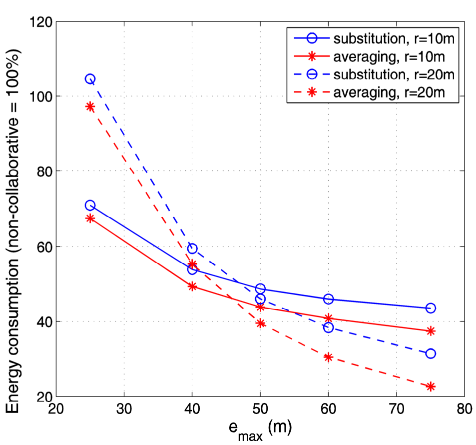

Figure 4. Energy consumption of collaborative GPS localization for various e_{max}.

consumed by non-collaborative localization. As expected, the consumption decreases with the number of neighbors. The decrease is not uniform: When the number of neighbors within 10 m (20 m) increases beyond 6.5 (18.9), which roughly corresponds to the density of 1.5-2 devices per 100 m^2, the consumption decreases only marginally. Interestingly, the results in Fig. 5 suggest that optimal r does not depend on device density. Clearly this is not true in general because optimal r tends to zero when device density tends to infinity. It, however, implies that, for scenarios of interest, close-to-optimal r (fine-tuning of r is anyway not possible) can be chosen solely based on e_{max}. Based on Figs. 3 and 4, it appears that $0.2e_{max} < r < 0.4e_{max}$ is a good rule of thumb for choosing the range in the considered scenario.

8. CONCLUSION

We evaluated two collaborative GPS localization protocols based on position substitution and position averaging. The evaluation is based on realistic simulations where the mobility of people is driven by real-world data from a theme park. Since the energy overhead of wireless interfaces is negligible in ad hoc networks where devices need to listen for incoming traffic anyway, the proposed schemes can provide significant energy savings compared to the non-collaborative GPS localization. One parameter that can be engineered to maximize the energy savings is the transmission range. In the absence of distance estimation between neighbors, the optimal range depends heavily on the maximum tolerable position error. We provide guidelines for choosing the range in the considered scenario. If distance estimation would be available on commodity phones, the consumption would always decrease with the range as long as the transmission power is a negligible part of that consumption.

The important problem of security (i.e. position information integrity) has not been addressed in this work. Without complex trust/reputation schemes, the collaborative localization protocols are prone to malicious announcements of incorrect positions. However, a coordinated effort of a significant number of malicious users is needed to introduce a persistent positioning error. We have not considered use case scenarios for collaborative GPS localization that would attract such coordinated attacks.

9. REFERENCES

[1] Wi-Fi Alliance: Wi-Fi Direct. [Online]. http://www.wi-fi.org/Wi-Fi_Direct.php. [Accessed: Oct. 30, 2011].

[2] Bluetooth Low Energy. [Online]. http://www.bluetooth.com/Pages/Low-Energy.aspx. [Accessed: Oct. 30, 2011].

[3] Nokia Instant Community gets you social. [Online]. http://conversations.nokia.com/2010/05/25/nokia-instant-community-gets-you-social/. [Accessed: Oct. 30, 2011].

[4] D. M.-K. Low et al., "Group formation using anonymous broadcast information," U.S. Patent 233 358, 2010.

[5] V. Vukadinovic and S. Mangold, "Opportunistic Wireless

Table 3. Average number of neighbors at different periods of the day.

Period	Neighbors (r =10m)	Neighbors (r = 20m)
14h - 15h	2.8	9.3
10h - 11h	6.5	18.9
11h - 12h	9.1	25.8

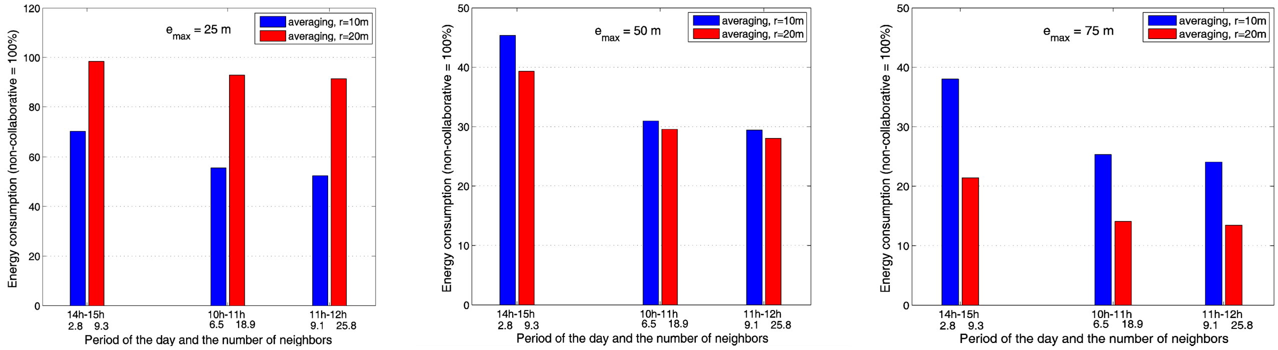

Figure 5. Energy consumption of collaborative localization depends on the average number of neighbors at different periods of the day.

Communication in Theme Parks: A Study of Visitors Mobility," *Proc. ACM MobiCom Workshop on Challenged Networks* (*CHANTS*), Las Vegas, USA, 2011.

[6] H. Wymeersch, J. Lien, and M. Z. Win, "Cooperative localization in wireless networks," *Proc. IEEE*, vol. 97, no. 2, pp. 427–450, Feb. 2009.

[7] R. Kurazume, S. Nagata, and S. Hirose, "Cooperative positioning with multiple robots," *Proc. IEEE Int. Conf. Robotics and Automation*, San Diego, USA, 1994.

[8] A. I. Mourikis and S. I. Roumeliotis, "Performance analysis of multirobot cooperative localization," *IEEE Trans. Robotics*, vol. 22, no. 4, pp. 666–681, 2006.

[9] P. Barooah, W. J. Russell, and J. P. Hespanha, "Approximate distributed Kalman filtering for cooperative multi-agent localization," *Proc. Int. Conf. Distributed Computing in Sensor Systems*, Santa Barbara, USA, 2010.

[10] P. Zhang and M. Martonosi, "LOCALE: Collaborative Localization Estimation for Sparse Mobile Sensor Networks," *Proc. Int. Conf. Information Processing in Sensor Networks*, St. Louis, USA, 2008.

[11] R. Jurdak, P. Corke, D. Dharman, and G. Salagnac, "Adaptive GPS Duty Cycling and Radio Ranging for Energy-Efficient Localization," *Proc. ACM Sensys*, Zurich, Switzerland, 2010.

[12] F. Zorzi et al., "Exploiting opportunistic interactions for localization in heterogeneous wireless systems," *Proc. NEWCOM++ / COST 2100 Joint Workshop on Wireless Communications*, Paris, France, 2011.

[13] L. Chan et al., "Collaborative Localization: Enhancing WiFi-Based Position Estimation with Neighborhood Links in Clusters," *Proc. Int. Conf. Pervasive Computing*, Dublin, Ireland, 2006.

[14] F. Della Rosa et al., "Experimental Activity on Cooperative Mobile Positioning in Indoor Environments," *Proc. IEEE Worshop on Advanced Experimental Activities on Wireless Networks & Systems*, Helsinki, Finland, 2007.

[15] C. Mensing et al., "Performance Assessment of Cooperative Positioning Techniques", *Proc. ICT Future Networks and Mobile Summit*, Florence, Italy, 2010.

[16] J. Hemmes, D. Thain, and C. Poellabauer, "Cooperative Localization in GPS-Limited Urban Environments," *Proc. Int. Conf. Ad Hoc Networks* (*AdHocNets*), Niagara Falls, Canada, 2009.

[17] J. Paek, J. Kim, and R. Govindan, "Energy-efficient rate-adaptive gps-based positioning for smartphones," *Proc. ACM MobiSys*, San Francisco, USA, 2010.

[18] M. B. Kjaergaard, J. Langdal, T. Godsk, and T. Toftkjaer, "Entracked: energy-efficient robust position tracking for mobile devices," *Proc. ACM MobiSys*, Krakow, Poland, 2009.

[19] Z. Zhuang, K.-H. Kim, and J. P. Singh, "Improving energy efficiency of location sensing on smartphones," *Proc. ACM MobiSys*, San Francisco, USA, 2010.

[20] F. Ben Abdesslem, A. Phillips, and T. Henderson, "Less is more: energy-efficient mobile sensing with SenseLess," *Proc. ACM MobiHeld*, Barcelona, Spain, 2009.

[21] D. H. Kim, Y. Kim, D. Estrin, and M. B. Srivastava, "SensLoc: sensing everyday places and paths using less energy," *Proc. ACM SenSys*, Zurich, Switzerland, 2010.

[22] I. Constandache, S. Gaonkar, M. Sayler, R.R. Choudhury, and L. Cox, "EnLoc: energy-efficient localization for mobile phones," *Proc. IEEE INFOCOM Mini Conference*, Rio de Janeiro, Brazil, 2009.

[23] J. Sharkey, "Coding for life – battery life, that is," *Google IO Developer Conference*, San Francisco, USA, 2009.

[24] L. M. Feeney and M. Nilsson, "Investigating the energy consumption of a wireless network interface in an ad hoc networking environment," *Proc. IEEE Infocom,* Anchorage, USA, 2001.

[25] P. A. Zandbergen, "Positional accuracy of spatial data: Non-normal distributions and a critique of the National Standard for Spatial Data Accuracy," *Transactions in GIS*, vol 12, pp. 103–130, 2008.

[26] D. Giustiniano and S. Mangold, "Demo: Distance Tracking Using WLAN Time of Flight," *Proc. ACM MobiSys*, Washington, USA, 2011.

[27] Navigating LBS Platform. [Online]. http://developer.qualcomm.com/file/617/navigatinglbsplatformspdf. Qualcomm Inc., 2010. [Accessed May 17, 2011].

[28] D. Halperin, B. Greensteiny, A. Shethy, and D. Wetherall, "Demystifying 802.11n power consumption," *Proc. USENIX HotPower*, Vancouver, Canada, 2010.

[29] A. Carroll and G. Heiser, "An analysis of power consumption in a smartphone," *Proc. USENIX Annual Technical Conference*, Boston, USA, 2010.