On Improving the Energy Efficiency and Robustness of Position Tracking for Mobile Devices

Mikkel Baun Kjærgaard

Aarhus University, Denmark,
mikkelbk@cs.au.dk

Abstract. An important feature of a modern mobile device is that it can position itself and support remote position tracking. To be useful, such position tracking has to be energy-efficient to avoid having a major impact on the battery life of the mobile device. Furthermore, tracking has to robustly deliver position updates when faced with changing conditions such as delays and changing positioning conditions. Previous work has established dynamic tracking systems, such as our EnTracked system, as a solution to address these issues. In this paper we propose a responsibility division for position tracking into sensor management strategies and position update protocols and combine the sensor management strategy of EnTracked with position update protocols, which enables the system to further reduce the power consumption with up to 268 mW extending the battery life with up to 36%. As our evaluation identify that classical position update protocols have robustness weaknesses we propose a method to improve their robustness. Furthermore, we analyze the dependency of tracking systems on the pedestrian movement patterns and positioning environment, and how the power savings depend on the power characteristics of different mobile devices.

Key words: energy-efficiency, positioning, mobile devices, power consumption, GPS, position update protocols

1 Introduction

An important feature of a modern mobile device is that it can position itself. Not only for use locally on the device but also for remote applications that require tracking of the device. Examples of such applications are geo-based information applications [2] or proximity and separation detection for social networking applications [9] just to mention a few. To be useful, such position tracking has to be energy-efficient to avoid having a major impact on the power consumption of the mobile device. Optimizing the operation of mobile devices for energy efficiency is an important issue and research is trying to address it from many angles as surveyed in [5], for instance, by trying to lower the impact of network traffic on power consumption [8] or by optimizing the execution at the operating system level [1]. Furthermore, tracking has to be robust in order to deliver position updates within limits when faced with changing conditions such as delays due to positioning and communication, and changing positioning accuracy.

As a basis for this work we divide the responsibility of remote tracking into *sensor management strategies* that on the device decides how to use available position sensors to estimate the current position and *position update protocols* that controls the interaction between the device and remote services. Such a division enables us to analyze the different combinations of sensor management strategies and position update protocols. Position update protocols has previously been studied, e.g., by Leonhardi et al. [11]. We will denote the combination of a sensor strategy and a protocol with *Strategy*:*Protocol*.

To quantify the impact of remote position tracking on power consumption, we have emulated the power consumption of a Nokia N95 phone in four different setups using the emulation tools and residential neighborhood dataset presented in Kjærgaard et al. [7]. In the first setup *(Periodic:Simple)* a periodic sensor management strategy every T_{period} seconds positions the phone using the built-in GPS receiver and then uses a simple protocol that immediately sends the position data using UMTS to a remote service hosted on an internet-connected server[1]. In the second setup *(Default:Distance)* a default sensor strategy positions the phone continuously by the update rate of the built-in GPS receiver (1Hz) and then uses a protocol that tries to minimize the number of position updates by only sending position data when the phone has moved more than a distance threshold T_{dist} meters from the last reported position. The third setup *(Default:Dead)* also uses a default strategy to position the phone by the update rate of the built-in GPS receiver and then uses a dead-reckoning protocol that sends an update when the distance between the current position and a server-side predicted position from the last reported heading, speed and position becomes greater than a given threshold T_{dist} meters. The fourth setup *(Dynamic(EnT):Simple)* uses a dynamic sensor strategy implemented by the EnTracked system [7] that tries to minimize the needed GPS fixes based on an accuracy limit T_{acc} meters and then uses a simple protocol that immediately sends the position data to the remote service.

The average power consumption for each setup with different accuracy threshold parameters are plotted in Figure 1 together with a robustness plot of the percentage of time the distance between the real position and the server known position is greater than the threshold. Comparing *Default:Distance*, *Default:Dead* and *Dynamic(EnT):Simple* we can notice that all three are able to lower the power consumption with between 560mW to 734mW compared to *Periodic:Simple* for the same accuracy threshold. The EnTracked system both minimize GPS and radio consumption whereas the distance-based and dead-reckoning protocols only save on radio consumption. Therefore we hypothesize that more power can be saved by combining EnTracked with either a distance-based or a dead-reckoning reporting protocol. However, a problem with either protocols is that they are less robust than *Periodic:Simple* and for most accuracy thresholds also *Dynamic(EnT):Simple* as shown on Figure 1. This drawback of

[1] 10 m/s is used as a conservative upper bound on the speed of pedestrian movement.

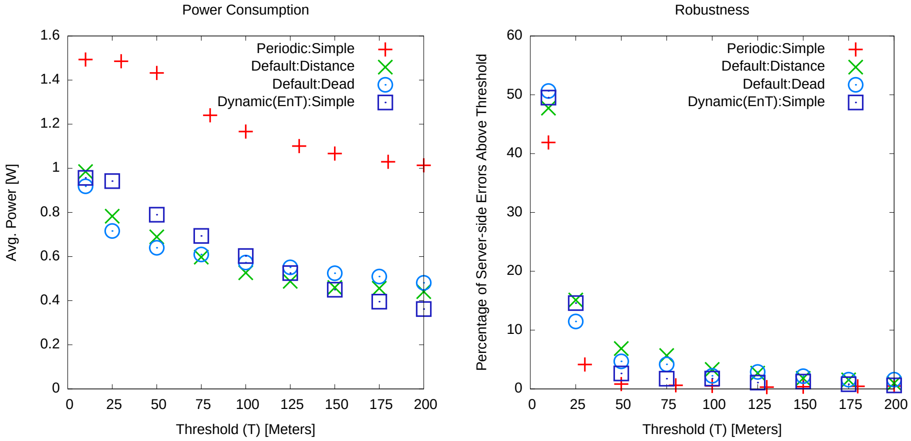

Fig. 1. Comparison of average power consumption and robustness for Periodic:Simple ($T = 10m/s * T_{period}$), Default:Distance ($T = T_{dist}$), Default:Dead ($T = T_{dist}$) and Dynamic(EnT):Simple ($T = T_{acc}$).

the protocols has been overlooked by previous work on position update protocols as they did not consider the ground truth accuracy [3, 11, 10].

We make the following contributions in this work: First of all, we propose a responsibility division for position tracking into sensor management strategies and position update protocols and combine the sensor management strategy of EnTracked with position update protocols which enables the system to further reduce the power consumption with up to 268 mW extending the battery life with up to 36% compared to the original system presented in [7]. Secondly, we propose a solution for improving the robustness of distance-based and dead-reckoning position update protocols that only marginally increases the power consumption. Thirdly, we evaluate how the power savings and robustness depend on the movement characteristics of pedestrian targets and the environment. Fourthly, we analyze how the power savings and the optimal system setup depend on the power characteristics of different mobile devices by deploying the system on the newer Nokia N97 phone which has significantly different parameters and power consumption levels.

2 Related Work

Previous work such as [3, 10, 11] has studied position update protocols to minimize communication and to minimize the load on server nodes by lowering the number of position updates. Leonhardi et al. [11] study time-based and distance-based protocols that takes a constant positioning accuracy and target speed into account. They study by simulation the number of updates each protocol produces and the average and maximum uncertainty of the server-known position.

They have later extended this work to consider dead-reckoning protocols [10]. Systems that tries to minimize the number of position updates for a specific application such as GeoPages have also been proposed [2].

A later work focusing both on sensor management strategies and position update protocols is Farrell et al. [4]. They propose strategies and protocols that take into account a constant positioning delay, target speed, and stress the importance of the fact, that it is not energy-free to use the GPS constantly as assumed by earlier work. Their solutions have been evaluated by simulation, where they can save around 50% energy in the evaluated scenarios. They have later extended this work for area-based tracking where they also take constant position accuracy and communication delays into account. For an indoor sensor network setting, You et al. [12] propose strategies and protocols that take into account a constant positioning accuracy and delay, target speed and acceleration to detect if the target is moving or not. They evaluate the techniques by emulation for IEEE 802.15.4 signal-strength-based indoor positioning and one of their results is that considerable energy savings can be gained from the use of an accelerometer to detect if the target is stationary or not. In our previous work Kjærgaard et al. [7] we proposed the EnTracked system that take into account dynamically estimated position accuracy and delays, communication delays, power constraints, target speed and acceleration (to detect if the target is moving or not). Furthermore the techniques was evaluated both by emulation and in real-world deployments.

In comparison, in this work we extend EnTracked with position update protocols to further reduce the energy consumption. Furthermore, we propose a method to improve the robustness of position update protocols and analyze the dependency of tracking systems on the pedestrian movement patterns and positioning environment, and how the power savings and optimal system setup depend on the power characteristics of different mobile devices.

3 Overview of Strategies and Protocols

As introduced earlier we divide the task of remote tracking into sensor management strategies and position update protocols. Sensor management strategies decide how to use available position sensors to estimate the current position. Sensor strategies could be implemented considering relevant properties such as position accuracy, power consumption, the availability of positioning in different environments (e.g., outdoor versus indoor) and privacy (e.g., WiFi positioning reveals a target's existence). Position update protocols control the interaction between the device and remote services which have to consider relevant properties such as server-side position accuracy, power consumption, data carrier availability and privacy.

In this paper we focus on outdoor GPS positioning of pedestrian targets and strategies and protocols for this setting. Figure 2 gives an overview of the considered strategies and protocols. A basic sensor management strategy is the default strategy that delivers position updates with the rate of a position sensor.

This strategy was implicitly assumed by prior work such as Leonhardi et al. [11]. The second strategy is a periodic strategy that with a frequency $T_{frequency}$ Hz requests a new GPS position fix. The third strategy is a dynamic strategy that dynamically changes the sampling rate depending on requests and availability. A system implementing a dynamic strategy is the EnTracked system [7] which consists of several elements but in this work we will consider it as a whole and refer the reader to the evaluation of the individual elements presented in [7]. The system consists of the following elements: distance-based scheduling, device-aware power minimization and movement awareness. The distance-based scheduling estimates when the next GPS fix is needed according to an error model that takes into account the positioning accuracy and requested accuracy. The device-aware power minimization uses a power minimization algorithm implemented using dynamic programming to predict when sensors has to be turned on and off. The algorithm uses a profiled device model to ensure that the system will correctly minimize the consumption and take into account, e.g., the delays associated with powering on and off the GPS and the radio. The movement awareness enable the system to switch between GPS and sensing motion using accelerometer readings. If the system can sense that a mobile phone is not moving, there is no reason to update the position on the server and the GPS can be switched off. But as soon as motion is sensed, the system switches the GPS back on.

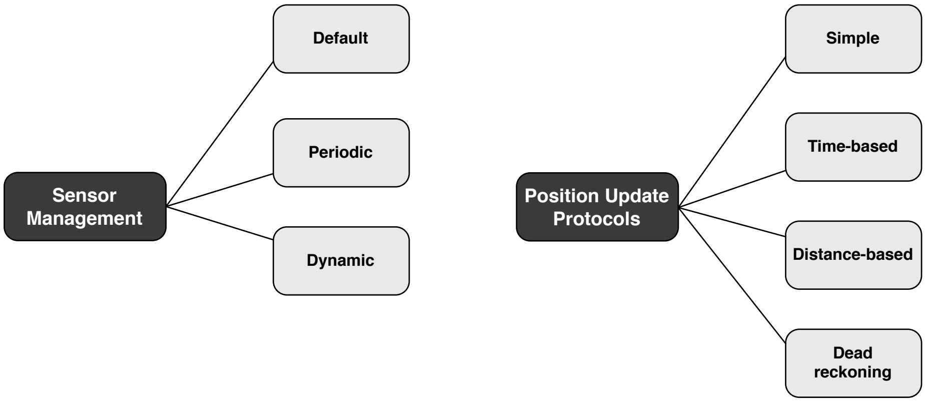

Fig. 2. Overview of sensor management strategies and position update protocols

In terms of position update protocols we restrict ourselves to protocols for a pedestrian scenario and device-controlled reporting protocols. Leonhardi et al. [11] list the following four types of reporting protocols that applies to the pedestrian scenario. Simple reporting which sends an update each time a position sensor provides a new position fix. Time-based reporting which sends an update each time a certain time interval of T_{period} seconds has elapsed. Distance-based reporting which sends an update when the distance between the current position and the last reported position becomes greater than a given threshold $T_{distance}$

in meters. Dead-reckoning which sends an update when the distance between the current position and the server-side predicted position from the last report position extrapolated with the reported speed and heading becomes greater than a given threshold $T_{distance}$ in meters. We refer the reader to [11] for a more detailed discussion and analytical analysis of these protocols in terms of their accuracy guaranties and communication efficiency. In this paper we will focus on the power consumption and robustness of these protocols on actual mobile devices. We will not consider Time-based combinations as any *Default:Time* combination is more efficiently implemented as a *Periodic:Simple* combination because this will lower both the GPS and radio usage.

The above discussion might indicate that out of the box any sensor strategy can be combined with any protocol, however, one have to take care of implementation pit falls. An example is the dead-reckoning protocol which assumes that the server can extrapolate the position as long as it does not receive new updates from the mobile device. In the classic protocol the threshold is tested continuously because a default strategy is implicitly assumed. The problem is what to do when a movement-aware dynamic strategy avoids to provide new updates because the device is detected not to move. In this case the server will continue to extrapolate the position which might violate the threshold. To address this issue we have extended the dead-reckoning protocol to test periodically if the server predicted position is about to violate the threshold and in this case send an extra position update with the last reported position and zero speed to stop the extrapolation.

A problem when implementing distance-based and dead-reckoning protocols is that they have robustness problems because they might not be able to keep the maximum error below $T_{distance}$ due to delays and positioning errors. Previous research such as Leonhardi et al. [11] did not observe this problem as they only considered data collected with a highly accurate differential GPS with a dedicated antenna in good signal conditions. In our work we focus on the more common case of less accurate GPS receivers found in mobile phones with smaller embedded antennas and in the non-optimal signal conditions found in urban and residential areas. To improve the protocols' robustness we propose to use the GPS receiver's estimates of it's current accuracy a_{gps} in meters and take this into account when evaluating if a threshold has been passed, e.g., for distance-based reporting the threshold equation then become: $d_{traveled} + a_{gps} < T_{distance}$ where $d_{traveled}$ is the distance between the last reported position and the current estimated position. This is an optimistic solution to strike a balance with power consumption as the pessimistic solution would be to also include the estimated accuracy of the last reported position because both the current and the last will be effected by GPS positioning errors.

4 Improving EnTracked using Position Update Protocols

To evaluate the extension of EnTracked with position update protocols we will consider several datasets in the following sections. This section considers the residential neighborhood dataset presented in Kjærgaard et al. [7] which was recorded on Nokia N95 phones for three pedestrian targets walking a 1.7 km tour in a residential neighborhood with several stops. The dataset consists of ground truth positions and 1 Hz GPS and 35Hz acceleration measurements collected from the built-in sensors. The ground truth was collected manually by walking a known route and collecting timestamps on a mobile device when reaching known points on the route. As mentioned earlier we will denote the combination of a sensor strategy and a protocol with *Strategy*:*Protocol*. For the dynamic strategy we will add "(EnT)" to mark that it is the EnTracked system that is used and the protocols that implements the accuracy extension have a "+" attached to their name, e.g., *Distance+* or *Dead+*.

The results for combinations of sensor strategies and protocols are shown in Figure 3. From the figure one can observe how the results for *Default:Distance+* slowly approaches it's lower limit which is equal to the background power consumption (62mW) plus the GPS consumption (324mW) because it only saves radio consumption. When comparing *Dynamic(Ent):Simple* with *Dynamic(Ent):Distance+* we can see, as hypothesized, that the distance extended version is able to further decrease the power consumption with between 95mW to 268mW for the evaluated thresholds which equals a 17% to 36%[2] increase in battery lifetime. To highlight that *Dynamic(Ent):Distance+* both save GPS and radio usage we have split up the consumption for the 200 meter threshold. For this threshold the average power consumption of the radio was 45mW, for the GPS it was 176mW, and for the background, CPU and accelerometer it was 63 mW. We have also evaluated *Dynamic(Ent):Dead+* which also provides an improvement comparable to *Dynamic(Ent):Distance+*. The main difference is that the dead-reckoning version is a few mW better for thresholds smaller than 100 meters and a few mW worse for larger thresholds. One reason behind the negligible improvement over the distance-based protocol is that if one compares with ground truth the average accuracy for the speed and the heading estimates are 0.35 m/s and 52° and therefore the server predictions will often be extrapolated in a non-optimal direction. It can also be linked to the movement style of a pedestrian which often take turns. For the protocol part previous work [10] has in terms of communication efficiency for a pedestrian movement style also observed only a small decrease in the number of position updates specially for large thresholds.

A problem with the original position update protocols as discussed in the introduction is that they might not satisfy the given thresholds if the magnitude of GPS errors are significant compared to the length of the threshold. Therefore we have evaluated our proposal of extending the protocols to take the estimated accuracy into account. Figure 3 shows a robustness plot of the percentage of time the distance between the real position and the server known position is greater than the threshold. In most cases the *Dynamic(Ent):Simple* protocol has the lowest values, often below two percent. For the ten and twenty-five thresholds the percentage is higher because the GPS errors alone often are enough

[2] The largest percentage is for the 200 meter threshold.

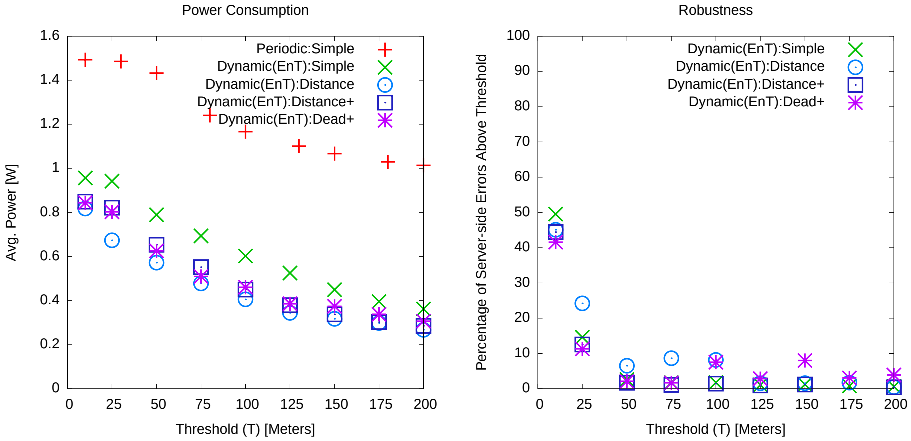

Fig. 3. Comparison of average power consumption and robustness for combinations of sensor strategies and position update protocols for the residential neighborhood dataset.

to violate the smaller thresholds as the average GPS error for the dataset is 11.8 meters. Comparing *Dynamic(Ent):Distance* with *Dynamic(Ent):Distance+* the proposed extension is able to lower the percentage of violations with between six to eighteen percentage points. The *Dynamic(Ent):Dead+* generally performs similar to *Dynamic(Ent):Distance+* except for two cases where it performs worse.

5 Power Consumption and Mobility

The previous section presented results that provided evidence that the combination of EnTracked with position update protocols can lower the power consumption and that the proposed extension can improve the robustness. In this section we would like to consider how the combination performs given urban positioning conditions and a pedestrian movement pattern with no stops. To test this we have collected a dataset with Nokia N95 phones for three pedestrian targets walking a 4.85 km tour in a urban environment with no stops. The dataset consists of ground truth positions and 1 Hz GPS and 35Hz acceleration measurements collected from the built-in sensors. The ground truth was collected at 4Hz with a high accuracy u-blox LEA-5H receiver with an dedicated antenna placed on the top of a backpack carried by the collector. The ground truth measurements were manually inspected to make sure they followed the correct route of the target. Using an urban setting instead of an residential setting tripled the magnitude of average GPS errors to 29,1 meters. That the dataset does not include any stops is adding to the difficulty because it means that EnTracked cannot save power using motion detection, it can only save power by distance-based scheduling and device-aware power minimization.

The results from running different combinations of sensor strategies and protocols is shown in Figure 4. Due to the fact that the dataset does not contain any stops the average power consumption for *Dynamic(Ent):Simple* is higher than for the residential dataset especially for smaller thresholds. The combination of EnTracked with distance-based or dead-reckoning protocols provides a decrease in power consumption between 102mW to 274 mW. The difference between dead-reckoning and distance-based is again insignificant. For the robustness the percentage of threshold violations are twice as high as for the residential dataset. This difference can be explained by the magnitude of GPS errors in the urban dataset but again the accuracy extended version is able to lower the percentage of threshold violations with ten percentage points except for the smallest threshold of ten meters. Therefore even in more difficult conditions the combination can provide savings while improving the robustness.

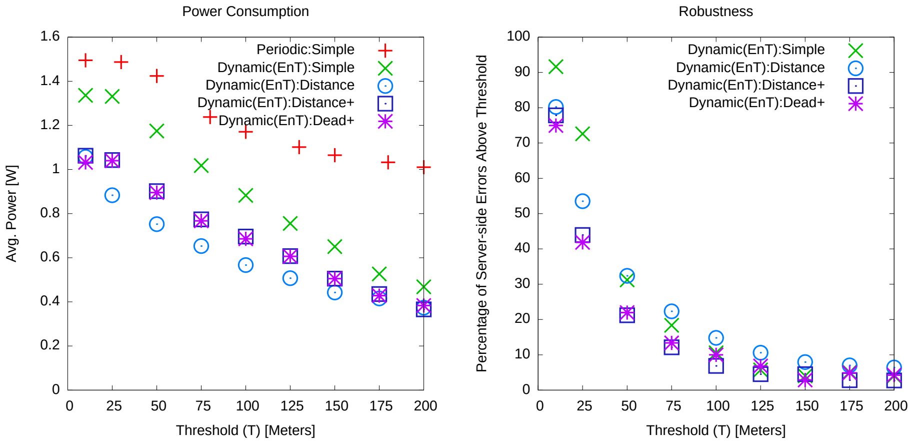

Fig. 4. Comparison of average power consumption and robustness for combinations of sensor strategies and position update protocols in the urban setting.

6 Power Consumption and Hardware Characteristics

An interesting question is how much the preceding results depends on the characteristics of the specific device. To answer this we collected a dataset with Nokia N97 phones during the same walks as for the urban N95 datasets (the collector carried both N95 and N97 phones). For the emulation and for the parameters needed by EnTracked we have profiled the delays, power consumption and needed thresholds of the N97 which are quite different from the N95 as can be seen from Table 1 which list values for both phones. To illustrate the values as they impact the combination of *Periodic:Simple* with $T_{period} = 60s$ we have collected energy measurements on both phones which is shown in Figure 5. The plot clearly illustrates that the power consumption for using both the GPS and radio is lower on the N97.

Table 1. Comparison of parameters for N95 and N97

Power Consumption		
	N95 [mW]	*N97 [mW]*
GPS	324	255
Radio idle	466	-
Radio active	645	753
Accelerometer	50	51
Idle	62	32

Delays		
	N95 [s]	*N97 [s]*
GPS Off	30.0	1.00
Radio idle Off	31.3	-
Radio active On	1.00	3.06
Radio active Off	5.45	4.75

Thresholds		
	N95	*N97*
$T_{Movement}$	1000	20
A_{norm}	3.71	2.53

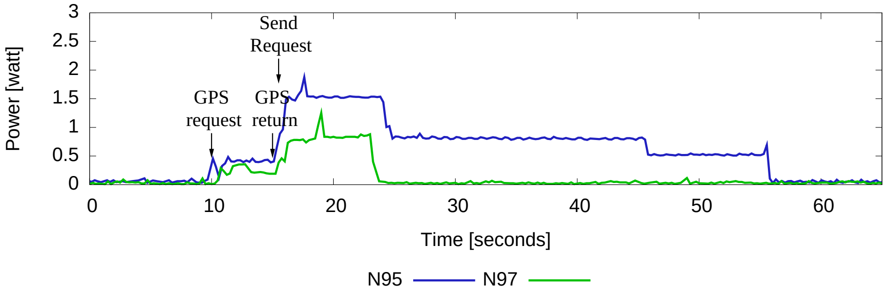

Fig. 5. Power consumption for N95 and N97 for *Periodic:Simple* with $T_{period} = 60s$

The new platform also significantly impacts the tracking results as shown in Figure 6. Generally the power consumption of all combinations are lower. The Periodic:Simple strategy is in this case a much better option than for the N95 data and is even better than Default:Distance which is limited by the GPS and background consumption as noted earlier. Considering the improvements of combining EnTracked with a distance-based or a dead-reckoning protocol there are significant savings for the thresholds below 125 meters, these savings are between 50mW-300mW and above between 20mW-30mW which extends the battery life with between 26% to 52%. Even though the savings are smaller in absolute numbers they are at least as significant as for the N95 due to the lower general power consumption. The robustness plot is given in Figure 6 for data collected in urban conditions with an average GPS error of 20.6 meters. The magnitude of GPS errors can again explain that values are higher for the ten meter threshold. The accuracy extensions are in this case able to lower the violation percentage with 2-7 percentage points for the distance-based combination. Therefore we can conclude that the combined system is able to improve the power consumption and robustness given another hardware platform.

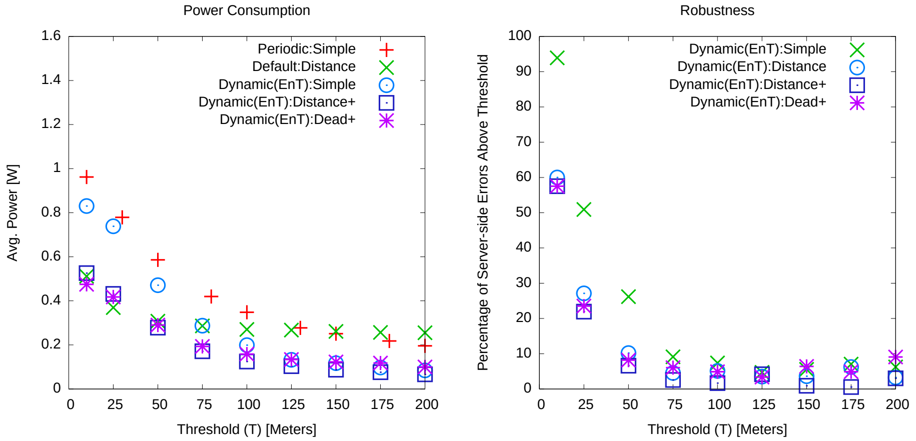

Fig. 6. Comparison of average power consumption and robustness for combinations of sensor strategies and position update protocols for N97 data.

7 Conclusions

The primary contribution of this paper is the responsibility division of position tracking into sensor management strategies and position update protocols and the combination of the sensor management strategy of EnTracked with position update protocols which enables the system to further reduce the power consumption with up to 268 mW extending the battery life with up to 36% compared to the original system presented in [7]. Furthermore, we proposed a solution for improving the robustness of distance-based and dead-reckoning position update protocols that only marginally increases the power consumption. The experimental results also provided evidence that the system could save power and improve robustness for pedestrians with a high mobility level in urban positioning conditions and when deployed on a new hardware platform.

In our ongoing work we are trying to address several issues. These are: First, propose methods for automatically determine the parameters of our device model for new devices. Secondly, apply the proposed methods and findings to other positioning technologies such as location fingerprinting [6].

Acknowledgements

The authors acknowledge the financial support granted by the *Danish National Advanced Technology Foundation* under J.nr. 009-2007-2.

References

1. M. Anand, E. B. Nightingale, and J. Flinn. Ghosts in the machine: Interfaces for better power management. In *Proceedings of the Second International Conference on Mobile Systems, Applications, and Services*, 2004.
2. Y. Cai and T. Xu. Design, analysis, and implementation of a large-scale real-time location-based information sharing system. In *Proceedings of the 6th International Conference on Mobile Systems, Applications, and Services*, 2008.
3. A. Civilis, C. S. Jensen, and S. Pakalnis. Techniques for efficient road-network-based tracking of moving objects. *IEEE Trans. Knowl. Data Eng.*, 17(5):698–712, 2005.
4. T. Farrell, R. Cheng, and K. Rothermel. Energy-efficient monitoring of mobile objects with uncertainty-aware tolerances. In *Proceedings of the Eleventh International Database Engineering and Applications Symposium*, 2007.
5. M. B. Kjærgaard. Minimizing the power consumption of location-based services on mobile phones. *IEEE Pervasive Computing*.
6. M. B. Kjærgaard. A Taxonomy for Radio Location Fingerprinting. In *Proceedings of the Third International Symposium on Location and Context Awareness*, 2007.
7. M. B. Kjærgaard, J. Langdal, T. Godsk, and T. Toftkjær. Entracked: energy-efficient robust position tracking for mobile devices. In *Proceedings of the 7th International Conference on Mobile Systems, Applications, and Services*, pages 221–234, 2009.
8. M. B. Kjærgaard, G. Treu, and C. Linnhoff-Popien. Zone-Based RSS Reporting for Location Fingerprinting. In *Proceedings of the 5th International Conference on Pervasive Computing*, pages 316–333. Springer, 2007.
9. A. Küpper and G. Treu. Efficient proximity and separation detection among mobile targets for supporting location-based community services. *Mobile Computing and Communications Review*, 10(3):1–12, 2006.
10. A. Leonhardi, C. Nicu, and K. Rothermel. A map-based dead-reckoning protocol for updating location information. In *Proceedings of 16th Int. Parallel and Distributed Processing Symposium*, 2002.
11. A. Leonhardi and K. Rothermel. A comparison of protocols for updating location information. *Cluster Computing*, 4(4):355–367, 2001.
12. C. wen You, P. Huang, H.-H. Chu, Y.-C. Chen, J.-R. Chiang, and S.-Y. Lau. Impact of sensor-enhanced mobility prediction on the design of energy-efficient localization. *Ad Hoc Networks*, 6(8):1221–1237, 2008.