SenSearch: GPS and Witness Assisted Tracking for Delay Tolerant Sensor Networks

Lun Jiang[1], Jyh-How Huang[2], Ankur Kamthe[1], Tao Liu[1], Ian Freeman[1], John Ledbetter[2], Shivakant Mishra[2], Richard Han[2], and Alberto Cerpa[1]

[1] University of California, Merced
[2] University of Colorado, Boulder

Abstract. This paper describes the design, implementation and performance evaluation of SenSearch, an outdoors, GPS assisted personnel tracking system using MICA motes. SenSearch is a mobile wireless ad-hoc network comprised of sensor nodes worn by users. These nodes store and forward information about the location of other nodes in an environment that lacks communication infrastructure. A key feature of SenSearch is that it does not require a continuously connected network for its operation. It is designed for a delay tolerant network that provides only occasional connectivity between nodes. It uses the distributed storage available through multiple nodes and the mobility provided by users to propagate the history of nodes' GPS locations to the processing center. The main contribution of this paper is an extensive, experimental evaluation of the system under controlled as well as uncontrolled environments. The paper discusses in detail the effects of a number of experimental parameters on the performance of the system.

1 Introduction

A delay tolerant sensor network is one in which sensor nodes are mobile and remain disconnected from each other most of the time. These nodes take advantage of occasional connectivity established when two or more nodes[1] wander within close range of one another to transmit data to a base station, which is typically a resource-rich node in a well-connected network, e.g. the Internet. Generally speaking DTNs can be characterized on the basis of the trajectory of the sensor node into open trail and closed trail DTNs. In open trail DTN, a node does not have a specific path and may never come with in close range of a base station. In this case, mobility patterns are unpredictable. Wildlife tracking is an example of this kind of DTN. In closed trail DTN, there are a small number of well-known paths that a node may take and most of the nodes in the system eventually arrive at a base station. Moving patterns are more predictable and nodes usually stay on the path. Hiking and vehicle tracking are examples of this kind of DTN.

---

[1] Node: An object (person, animal) that carries a wireless sensor module capable of communicating with other nodes. This term is used interchangeably along with entity or mote throughout the paper.

In this paper, we describe the design, implementation and performance evaluation of SenSearch, a GPS and witness-assisted tracking system for hikers in outdoor environments. Our application belongs to close trail DTN , which means the trajectory of the experiment subject is known and it is relatively easy to predict its current position based on prior knowledge. In this system, GPS modules are used to enable a sensor unit to infer its location information. The system uses the concept of witnesses to convey a node's movement and location information to the outside world. This helps overcome the constraint of having a constantly connected network. Different nodes exchange their location data when they encounter each other along the way. This information is subsequently routed to a base station through a series of subsequent data transfers between nodes. This data consists of the history of past locations of a node, and can be used to estimate the location of a missing node using the history information acquired by other nodes. The use of the GPS data drastically reduces the search space, by increasing the accuracy of the system. However, the use of GPS along with the transmission/reception of data in an energy-constrained system also reduces the lifetime of the system. To save energy, SenSearch uses a duty cycling scheme for the GPS and radio units. We discuss in detail the trade-offs between lowered accuracy due to duty recycling and the resulting energy saving in the performance evaluation section of this paper. This paper makes three important contributions:

1. SenSearch is the first sensor-based DTN system designed to locate missing persons and assist in search and rescue in wilderness environment that totally lack communication infrastructure. Using novel coordination strategies and adaptive mechanisms, we drastically improve the lifetime of the system.
2. Two separate implementations of SenSearch on two different computing platforms as well as a simulation in a discrete event simulator are provided. A comparison between the two implementations provides important insights into the effects of the underlying platform on the performance of the system.
3. An extensive performance evaluation of the SenSearch system including experiments in controlled outdoor environments such as a university campus as well as uncontrolled, wilderness environments such as US state and national parks. Overall, a comprehensive analysis of different parameters that affect SenSearch's performance is provided from over 65 different hiking experiments that we performed over a period of seven months in UC Merced campus, Mt. Sanitas and Chautauqua Park in Boulder and Boulder reservoir.

2 Related Work

A large majority of networks [3], [4], [5] used for tracking movements of mobile entities are comprised of hundreds of small, densely distributed wireless sensor nodes deployed in the field. However, it is not well-suited for tracking the movement of entities with random mobility patterns on paths going through large area as it would require a prohibitively large number of nodes to cover all possible locations.

SenSearch is based on CenWits [6], a connection-less sensor-based tracking system using witnesses. CenWits is comprised of mobile nodes that receives its location information periodically from location points and passes it to other nodes during subsequent encounters. This information is then transmitted to access-points distributed at various locations. The improvement of SenSearch compared to CenWits include the followings. Using the state of art low power GPS sensor board with the wireless sensor to improve the accuracy of the system; implementing a group layer protocol to solve and take advantage of the tagging along problem of the human movement. BikeNet [2] is an opportunistic sensor networking system, wherein customized sensor modules and mobile phones are used for real-time and delay tolerant uploading of data. Electronic Shepherd [8] is a low-power, low-bandwidth application for tracking the movement of animals with flock- behavior. It requires a GPRS/GSM or 802.11 network to retrieve data form the sensors. All of the above systems depend on the presence of some communication infrastructure backbone such as location points (CenWits) and cell towers (BikeNet) to convey information to the base station. In many outdoor scenarios (or hostile environments), it is not practical to deploy location and access points for information storage and recovery. SenSearch does not need a communication infrastructure as it relies on people carrying data (muling) collected during encounters with other nodes to the base station. Also, we provide real world experimental results in addition to a complete evaluation of the parameters in the design space that was lacking in earlier papers [6].

SenSearch is similar, in intent, to the ZebraNet [7, 10], which is designed for the tracking of wild animals. SenSearch differs from ZebraNet in a number of areas. In ZebraNet, hardware comprises of a CPU, a low power radio for short range communication, a long range radio for communication with the base station and a solar cell array along with the Li-Ion batteries. The total weight is around 1,151 grams. The hardware is powered by rechargeable solar cells supplying 13.5 Ampere-hour of energy. On the other hand, SenSearch consists of MICAz sensor nodes powered by 2 AA batteries having a limited battery life of 5.2 Ampere-hour of energy and weigh around 150 grams. SenSearch is intended for use in outdoor activities for human tracking and hence, the weight restrictions are more severe in comparisons to ZebraNet. For SenSearch, it would be impractical to have solar cells to power such a system, because carrying anything larger than a pager would be burdensome. Also, in ZebraNet, the use of high power, long range radios allows for transfer of data to reach the base station even in a single-hop communication. As opposed to that the use of low power, short range 802.15.4 radios in SenSearch which requires the use of a different strategy, relying on the encounter between nodes as the method of forwarding information back to base station. The disconnected nature of SenSearch network denies the possibility of using an muilihop routing protocol, as used by ZebraNet which has a routing path from each node to the base station all the time. These distinct differences in purposes of the applications result in different hardware as well as software infrastructures used between these two projects.

Name	Explanation	Value
P_{CPU}	Base Power (CPU only)	11.04 mW
P_{GPS}	GPS in Hot Start mode	170.07 mW
P_{TX}	Radio in TX state	78.49 mW
P_{RX}	Radio in RX state	74.85 mW
P_{LPL}	Power used by the Radio in LPL Listen state	21.95 mW
$T_{GPS_{HS}}$	Avg. Time for GPS to get coord. in Hot Start mode	3.42 sec
R_{BW}	CC2420 TX bandwidth	250 Kbps
B_{SIZE}	Beacon packet size	120 bits

Table 1. Factors affecting the power consumption and data transmission of a MICAz mote with a GPS Module.

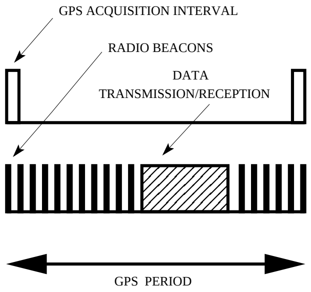


Fig. 1. Nodes have overlapping periods in which they acquire GPS and send radio beacons, even data exchanges with other nodes.

In this paper, the design problems that are unique to systems like SenSearch are highlighted. We propose solutions in the form of two implementations for slightly different hardware platforms and perform extensive performance evaluation on points of interest for system designers.

3 System Description

SenSearch uses Berkeley MICAz motes equipped with an RF transmitter and a GPS receiver to track locations of entities (example: people, animals, etc.) in natural or urban environments in the absence of a communication infrastructure.

Each node has a unique ID. It keeps records of its locations by periodically using the GPS receiver and saving the information in an internal database. Energy savings by duty-cycling the GPS results in decreased accuracy in the sense of the localization of missing entities. Continuously sampling the GPS data increases accuracy but results in prohibitively large power consumption. Each node emits periodic radio beacons to detect the presence of other nodes in its vicinity. When any two nodes are within radio range of each other (encounter), they exchange their databases in response to the radio beacons. An encounter can be classified as successful or missed, depending on whether the nodes are able to exchange their databases when sending periodic radio beacons. The information exchanged in an encounter is propagated among multiple nodes in subsequent encounters during database exchanges. It is possible to estimate the expected position or area of any missing node by examining the history of its past locations from GPS data muled by other nodes. Fig. 1 shows an illustration of the GPS acquisition in hot start mode, radio beacons and data transmission during a node encounter.

3.1 Hardware

Wireless Sensor Module The MICAz is based on the ATmega128L microcontroller with ChipCon CC2420 radio with a a 250kbps data rate. The MICAz has 4KB of RAM and 512KB of external flash memory.

GPS Module In the first implementation (*SenSearch-A*), the MTS40CA sensor board was used. It has an 12-channel integrated Leadtek 9546 GPS module with a SiRFstart chipset having position accuracy up to 10m. It has times of 45s, 38s and 8s for cold, warm and hot starts respectively, with a reacquisition time of 0.1s. It draws current of 60mA@3.3V [1]. In the second implementation (*SenSearch-B*), the MTS420CC has a uBlox LEA-4A GPS module with the ANTARIS 4 chipset. It has 16 channels and position accuracy of 3m CEP. It has times of 34s, 33s and 3.5s for cold, warm and hot starts respectively, with a reacquisition time of <1s. It draws current of 35mA@3V.

Power Measurements In Table 1, we provide the power consumption for the MICAz mote and the MTS420CC sensor board in different modes of operation for the SenSearch-B implementation. We measured the power consumption using the National Instruments ELVIS platform. Each measurement is averaged over 5 different nodes.

3.2 Software Architecture

In the following section, we cover the design decisions and trade-offs specific to each of the SenSearch implementations:

SenSearch-A MANTIS is a multithreaded embedded OS for wireless sensor networks. The MANTIS implementation is based around actions executed in response to a periodic timer and radio interrupts. When there are no other motes in the radio range and the timer fires, the application layer is triggered. The application layer will then decide if it is time to send a beacon or to take a GPS reading. While sending a radio beacon, the program accesses the network layer. Periodically, we update our GPS position by recording a new reading in our database. Upon getting an interrupt from the radio, the network layer forwards the packet to the application layer which takes actions based on the type of packet (HELLO (beacon), REPLY and DATA. The received database entries replace older entries for each node in the local nodes database. We save a subset of the total received entries in the RAM to avoid the latency of accessing the data from the flash storage module.

SenSearch-B In *SenSearch-B*, we used the event-based TinyOS operating system for wireless sensor network platforms. The TinyOS (TOS) implementation is based around actions being executed according to a state machine shown in Fig. 3.2. In the background, the node records GPS information in its database according to the GPS period. At the beginning, the node is in SEND BEACONS state. When a node receives a beacon from another node in its vicinity, it goes into SEND TABLE state. In SEND TABLE state, the node sends its database in response to the beacon. In the last packet, the node transmits an END OF TRANSMISSION flag to signal the switch to DATAMSG TIMEOUT state. The node switches to the RECVING TABLE state when it starts receiving GPS

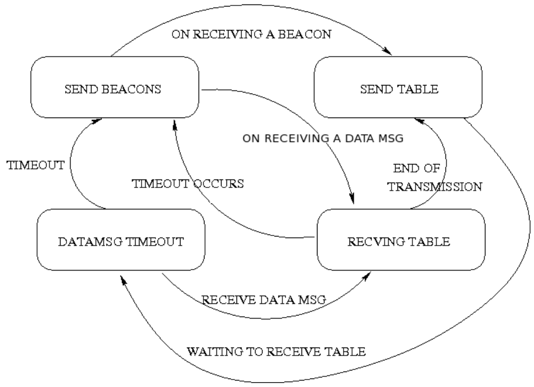
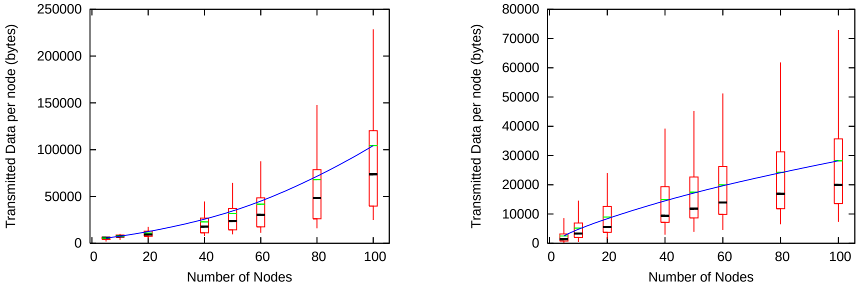


Fig. 2. State Machine for SenSearch-B implementation.



(a)

(b)

Fig. 3. Amount of data transferred when (a) the system is affected by "tagging along" problem in simulation; (b) "tagging along" problem is averted in simulation by separating nodes by distance greater than radio range.

data from the other node. If the other node has moved out of radio range, then the node will wait for a specific period in the DATAMSG TIMEOUT before going into SEND BEACONS state again. In the RECVING TABLE state, the received database entries replace older entries for each node in the local node's database. A node can switch from SENDING TABLE or RECVING TABLE state to SEND BEACONS state in case of errors (like packet loss) when its times-out is triggered in the DATAMSG TIMEOUT state.

In order to examine the behavior of different parameters for our implementation, we created a discrete-event simulation environment for SenSearch (See Section 4.2). In the preliminary simulation results, we realized that the nodes transmitted extremely high amounts of data. Nodes in close proximity kept transferring their GPS information databases in response to beacons from one another even though no new information had been recorded since the prior data exchange, leading to high power consumption. This was attributed to the simple encounter-based data transmission which occurs when two or more nodes are in close proximity of each other. This behavior was termed "tagging along of nodes". Figure 3(a) shows the amount of data transmitted as a function of number of nodes in the simulation when nodes tag along with each other. Figure 3(b) shows the reduction in data transmitted when the tagging along is avoided artificially by separating the nodes by distances greater than radio range. Compared to the linear growth in Fig 3(b), Fig 3(a) shows exponential growth of data being exchanged if “tagging along” scenario is not avoided. The “tagging along” effect impairs the system in two ways. First, constantly exchanging beacons and data with the same nodes will reduce the chances of hearing a beacon from a node passing-by. Second, the additional transmission will cost energy and reduce the system life time. The amount of data exchanged via radio can be reduced by comparing times of most recent GPS entry which still results in significant energy consumption. However, this does not eliminate the sending of beacon messages. In SenSearch-B, we take advantage of the occurrence of the “tagging along” scenario by implementing a coordination scheme between nodes which we term as *group layer*. The group layer aims to increase energy savings by adapting radio and GPS usage when nodes “tag along”.

Memory Management strategy: Generally, in SenSearch, the encounters between nodes are in the order of a few seconds. The beacon message period is also in the order of seconds (see Table 2). If the beacon arrives at the start of the encounter between nodes, then the data exchange between the two nodes has a greater chance of completion than if the beacon arrives when the two nodes are at the end of the radio range and moving away from each other. Due to this, the size of transmitted GPS database needs to be minimized while not affecting the tracking performance (localization error) of the system. Since, the goal of our system is to narrow the location of a missing node to a relatively small area, and not pin-point its current position, it suffices if we are able to transmit a small number of GPS entries per node. So, we store the latest $MAX\_ENTRIES$ number of entries for every node.

Group Layer Design: From an operational point of view, the group layer is comprised of two node states, leader and member. Initially every node has leader status. The nodes broadcast their leader status and battery voltage periodically at a rate determined by the group leader synchronization period. Other leader nodes. When other leader nodes receive a pre-specified number of group leader synchronization messages and if they have lower battery voltage, they will will respond to the potential new group leader with a request to join its group. When the potential leader node receives a join message, it sends an acknowledgment confirmation message to the group member. When the confirmation from the new leader is received, the leader and the new group member synchronize their GPS entry databases. The group member powers off its GPS module, stops sending beacon messages and listens for group leader update and synchronization messages. Since, the group leader is the only node communicating with other “new” nodes, it sends periodic update messages synchronizing the leader’s GPS entry database among its member nodes. The group members respond with a short ACK message with their current battery voltage. In the event that the member node misses a certain number of group leader synchronization messages, the node breaks away from its group leader, reestablishing itself as a group leader.

Also, if the group leader finds a member in its group having energy reserves above a certain threshold than its own, it sends out a resign message which will make all the group members independent leaders again. If the nodes still tag along with each other, they will join up as a whole group eventually as described above. The design of group layer follows two important rules: First, all the nodes in a group always have the same GPS entry information, ensuring that there is no loss of the GPS information acquired from encounters. Second, the node with the highest energy reserve serves as the group leader improving the lifetime of the nodes in a group and the system as a whole. Additionally, the *group layer counter* is a parameter which affects the performance of the group layer design. In our implementation, if the group layer counter is set to two, a node needs to receive to at least six group leader synchronization messages i.e., group layer counter times 3, in order to associate with a group leader. A node breaks away from a group, if it misses *group layer counter* number of successive group leader synchronization messages. Thus, the *group layer counter* parameter controls the formation and breakup times for a group. If it is set high, the energy savings from the group layer functionality will be reduced because of the difficulty in forming a group. If it is set low, the group will be constantly breaking up and reforming again.

Adaptive GPS Design: In order to improve the lifetime of the nodes, the power consumption has to be reduced. As seen from Table 1, the GPS module consumes the most power. In the system, if a node does not encounter any other nodes for a long period of time, its old GPS entries are overwritten due to the memory constraints. This indicates that the GPS module is wasting energy by acquiring coordinates too often. In this section, we describe the strategy for GPS period adaptation.

Assuming a Poisson distribution of encounters for a specific node and given a sample of n measured values k_i, we can estimate the value of λ using the maximum likelihood:

$$L(\lambda) = \log \prod_{i=1}^{n} f(k_i|\lambda) = \sum_{i=1}^{n} \log\left(\frac{e^{-\lambda}\lambda^{k_i}}{k_i!}\right) = -n\lambda + \left(\sum_{i=1}^{n} k_i\right)\log\lambda - \sum_{i=1}^{n}\log(k_i!) \tag{1}$$

Solving the equation for the maximum-likelihood estimate of λ,

$$\frac{d}{d\lambda}L(\lambda) = 0 \Longrightarrow \lambda_{MLE} = \frac{1}{n}\sum_{i=1}^{n} k_i \tag{2}$$

In our case, $k_i = \frac{1}{t_i - t_{i-1}}$ or the inter-arrival frequency. We keep a running average of the encounter frequency, i.e.

$$\lambda_{MLEi} = \frac{n \times \lambda_{MLEi-1} + k_i}{n+1}$$

Because we want at most $MAX\_ENTRIES$ GPS acquisition between two encounters, we dynamically adapt the GPS period to be

$$min\{\frac{1}{\lambda_{MLE} \times MAX\_ENTRIES}, GPS_p\} \tag{3}$$

where GPS_p is the GPS period for a node (refer Table 2).

Parameter	Values for Experiments
GPS period (GPS_P)	**20, 45, 60**, *90, 180, 300* (sec)
Beacon period (B_P)	**3, 5, 7** (sec)
Number of nodes	**5, 10, 15**, *20, 40, 60, 80, 100*
Group Layer Counter	**2, 4, 8**
max DB size(DB_{SIZE})	**45, 60, 75** (total # GPS entries)

Table 2. Parameters explored in Experiments for SenSearch.(Note: Values in bold faces are explored in simulation and in experiments whereas values in italics are explored only in simulation.)

4 Performance Evaluation

4.1 Goals and Objective Functions

Our goals while evaluating the performance of SenSearch are two-fold. First, we want to characterize the behavior of the system based on a specific set of objective functions. Second, we want to understand the relationships between different parameters and their impact on system performance. The parameter space explored is given in Table 2. The objective functions evaluated are:

1. Localization Error: How well can we track the missing entities? What is the localization error based on the information provided by other witnesses?
2. Power Consumption: What is the lifetime of the system? What are the effects of tuning different system parameters?

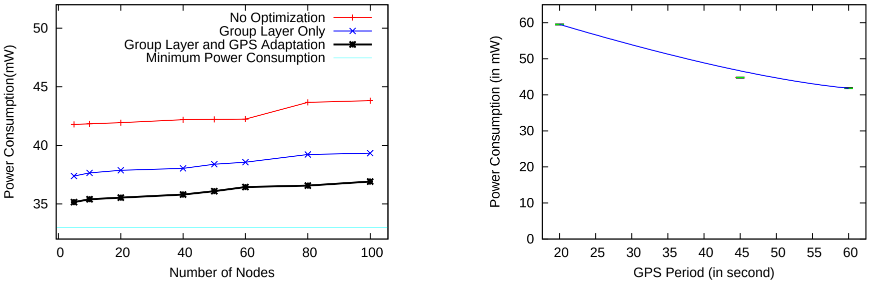
4.2 Simulation

We built a discrete event simulator in C to understand the effect of different parameters on system performance. For each simulation run, all nodes have the same set of system parameters including GPS period, beacon rate and memory limitation. Nodes have different speeds while traveling along a specified path. We assume a constant radio range of 50 meters in all our simulations.

Through simulations, we investigated the effect of group layer and GPS period adaptation on the power consumption of the system. Figure 4(a) shows power consumption as a function of the number of nodes. In these simulations, the group layer counter is set to 4 and GPS period is set to 45 seconds. Notice that with only group layer enabled, the power consumption reduces from 42mW to 38mW, and with both the group layer and GPS period adaptation enabled, we can reduce the power consumption to 36mW. The minimum power consumption is calculated with $P_{CPU} + P_{LPL}$, i.e. 32.99 mW. As the number of nodes increases, the node encounters increase. As a result, the group leaders exchange more beacons with each other and more group messages with the group members, which increases the overall power consumption.

4.3 Experiments

We conducted 12 experiments using the SenSearch-A implementation at the University of Colorado-Boulder (Oct'2007-Mar'2008) and 41 experiments using the SenSearch-B implementation at the University of California-Merced (UCM)



(a) Simulation (b) Experimental

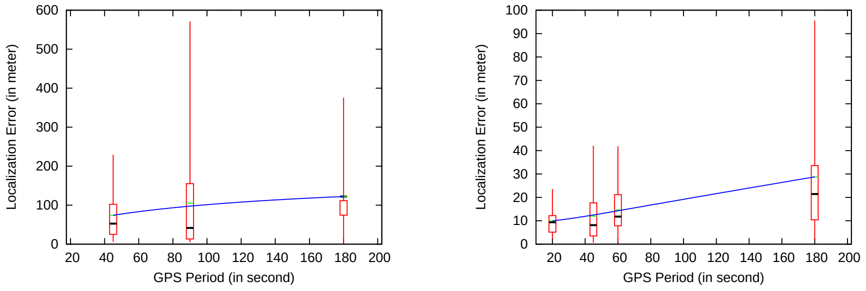
Fig. 4. (a) Power consumption as a function of number of nodes. (b) Variation of power consumption as a function of the GPS period.

(Oct'2007- Mar'2008). We conducted a further 12 experiments after adding the group layer protocol. We performed a number of experiments with SenSearch in real-world environments in Mt. Sanitas near Boulder, and in Boulder Reservoir city park. A typical run involves a hike on a mountain ridge and averages 2 hours round trip time. Different hikers started their hikes at different times with a separation of about twenty minutes. When hikers reached a summit in the middle of their hike, they tended to stay for a while there before returning to the base station. In controlled experiments conducted at the UCM campus, the nodes walk individually or in groups along a path (roughly 1000m) towards each other, starting from opposite ends of the path. From the data dumped at the base station, the localization error, power consumption and memory usage is computed. In both the implementations, we record the ground truth using hand held GPS units at way-points along the path.

4.4 Results

To present our experimental results, we use box-plots [9] instead of displaying the means with error bars. This method improves data visualization by displaying the minimum, first quartile (25%), second quartile (or median, 50%), third quartile (75%) and maximum value of the distribution of data in addition to the mean. In our graphs, we connected the means with a smoothing line to visualize trends in the data.

Localization Error: The localization error is the difference in the real location of a way-point (ground truth) and the estimated position of the same using the data from the SenSearch system relayed back using witnesses. Since, the GPS information recorded by the node is collected periodically, there is uncertainty regarding the location of a node in between two recorded positions. This uncertainty is introduced by the random mobility patterns which includes changes in direction and speed. The GPS hardware error could account for part of the localization error but this was not the case in our experiments. Also, in this paper we are not focusing on the specific algorithms that can be used to calculate the estimated position of the missing node;the goal of our system is to narrow the



(a) SenSearch-A

(b) SenSearch-B

Fig. 5. Localization Error (LE) as a function of GPS period.

location of a lost entity to a relatively small area, and not pin-point its current position. For that purpose, we used simple linear extrapolation to estimate an expected position for a node as a function of all the previous GPS records in the database. In order for this simple method to work, we need at the very least two known positions and the corresponding times so we can infer direction and speed of the entity being tracked. Nodes cannot be localized if no data points are received for a specific node, or if only one data point is obtained.

Figs. 5(a) and 5(b) shows the Localization Error (LE) as a function of the GPS period (GPS_P) for SenSearch-A and SenSearch-B, respectively. We see that the LE increases as the GPS period increases. This is attributed to the increase in the number of GPS entries for a node transported to the base station by all other nodes in the system as a function of GPS period. Here, as the GPS period increases, the number of GPS entries at the base station per node decreases. As location information for a node decreases, the localization error increases. Shorter GPS periods results in more location information. With more data points available, we can reduce localization error, as long as past movement behavior of the entity is correlated with future locations. LE values are much greater for the SenSearch-A (Fig. 5(a)) as compared to the SenSearch-B (Fig. 5(b)). This fact can be attributed to the differences in GPS hardware and the path followed by the nodes during the controlled experiment. For SenSearch-A, the nodes always walk in a straight line and turn back at the end of the path whereas for SenSearch-B, the nodes walk in a straight line towards each other. For experiments in SenSearch-A, if the nodes do not exchange data on their way back, our linear extrapolation for the expected position does not work and we get inflated errors. This is because the information received at the base station indicates that the node kept moving in a straight line, even though it turned around. In such cases, having contextual knowledge of the terrain and possible paths could significantly improve the quality of the estimation.

Power Consumption: The GPS module is the dominant factor in the power consumption of the system. In addition, in a system with high levels of radio activity, transmissions/receptions/idle listening times account for a majority of the power consumption outside the GPS module. Power consumption would be affected by the amount of transmitted data per node (beacon + database) and the total time the radio and the GPS modules are active. The total power consumption can be expressed as follows:

$$\begin{aligned} PC = P_{CPU} + \frac{T_{GPS_{HS}} \times P_{GPS}}{GPS_P} + \Big(1 - \frac{(T_B + T_{RX} + TTX)}{T_{EXP}}\Big) \times P_{LPL} \\ + \frac{T_B}{T_{EXP}} \times P_{TX} + \frac{1}{T_{EXP}} \times (T_{TX} \times P_{TX} + T_{RX} \times P_{RX}) \end{aligned} \tag{4}$$

where T_B is the total time spent in sending beacons, T_{TX} and T_{RX} are the total time spent in sending and receiving the GPS entry table, T_{EXP} is the time duration of the experiment. The remaining terms used in this equation are explained in Tables 1 and 2. Each term in Eq. 4 represents power consumed by a different part of the system. The GPS module is the dominant factor in the power consumption of the system. In addition, in a system with high levels of radio activity, transmissions/receptions/idle listening times account for a majority of the power consumption outside the GPS module. Power consumption would be affected by the amount of transmitted data per node (beacon + database) and the total time the radio and the GPS modules are active.

From Figure 4(b) and Eq. 4, we can see that the GPS period is the major factor in the power consumption of the system. As the GPS period increases, the power consumption decreases because the GPS module is active for a lower fraction of time. In the current scheme, and under the range of dynamic conditions tested in our experiments, the GPS period is the dominant factor in power consumption. However, for a system with a longer GPS period, the power consumed in the low power listening mode by the radio becomes dominant. In our experiments, the power consumption remains unaffected by the beacon period and number of nodes in the current scenario.

From Figures 3(a) and 3(b), the difference in data transmission is obvious highlighting the effect of the group layer coordination scheme in improving the lifetime of the system. The group layer counter parameter can lead to significant changes in the way the GPS is utilized, hence affecting the power consumed by each node. As explained earlier, the group layer counter affects the time required for a node to join a group. In Figure 6(a), we see that as we decrease the value of the group layer counter, the group member nodes power off their GPS for greater durations of time. Since the GPS module consumes significant energy (see Table 1), powering it off results in significant energy savings. This is demonstrated in Figure 6(b) which shows that power consumption reduces to 36mW when the group layer counter is set to 2. This is a significant saving of 20% compared to power consumption shown in Figure 4(b) (at GPS_p=45s) where no group layer mechanism is implemented. From these results, we conclude that the group layer coordination scheme significantly reduces the energy consumption of the node.

5 Summary and Future Work

It has amply been noted that there is a large gap between building a sensor network system in theoretical realm and a successful, wide-spread deployment of

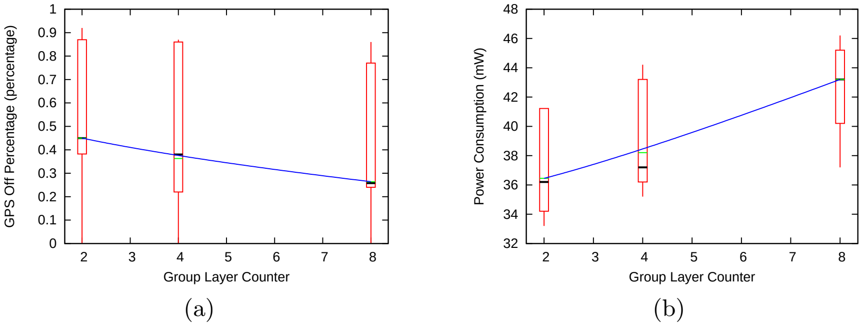


Fig. 6. Effects of Group Layer Counter tuning on GPS off percentage and Power Consumption respectively.

such a system for real-world, every-day use. After creating a working prototype in lab, it has taken us more than seven months, more than 65 outdoor experiments, and a large number of simulation studies to sort out the problems caused by various environmental factors and tune various parameters. The most important contribution of this paper is the lessons learned in a successful deployment of SenSearch in controlled and uncontrolled environmental conditions.

The first lesson is that duty cycling the GPS module is critical for this class of systems. A shorter GPS period enables more data for the missing node to be acquired at the base station. However, increased localization accuracy comes at the cost of higher power consumption and reduction in the system lifetime. It is possible to increase the lifetime of the systems without incurring significant penalties for the localization and tracking of nodes by changing the GPS period adaptively. The balancing point in this trade-off depends on the specific usage of the system and the pre-acquired knowledge of the environment.

The second lesson is that such system must plan for groups of nodes moving together. These groups are typically dynamic in nature with nodes leaving a group and new nodes joining a group at various times. One of the first problems we encountered in our initial experiments was that a significantly large amount of mostly old information got exchanged when nodes moved together, first among the nodes in the group, and then by the rest of the nodes in the system once other witness nodes were encountered. After experimenting with different scenarios, we have implemented a simple optimization that consisted of limiting each pair of node data exchanges to one during a particular GPS period. This simple optimization helped us to significantly reduce the redundant exchange of old information.

The third lesson is that the group structure can be utilized to reduce power consumption. We introduced group layer design along with a tunable parameter, group layer counter. Group layer, along with an adaptive GPS period results in significant power savings. The use of adaptive algorithms similar to the one introduced in this paper can be used to adapt parameters such as GPS entries per node, beacon period and group layer counter.

In the future, we plan to explore more complex methods for localization and tracking than the simple linear fitting methods used in this paper. Furthermore, we would like to continue the exploration of adaptive approaches to tune the other parameters of the system such as those mentioned above. We plan to get further experimental data with a larger number of nodes to verify some of the simulation findings as well as to experiment with the dynamic nature of groups.

References

1. http://www.xbow.com/.
2. S. B. Eisenman, E. Miluzzo, N. D. Lane, R. A. Peterson, G.-S. Ahn, and A. T. Campbell. The bikenet mobile sensing system for cyclist experience mapping. In *SenSys '07: Proceedings of the 5th international conference on Embedded networked sensor systems*, pages 87–101, New York, NY, USA, 2007. ACM.
3. L. Gu, D. Jia, P. Vicaire, T. Yan, L. Luo, A. Tirumala, Q. Cao, T. He, J. A. Stankovic, T. Abdelzaher, and B. H. Krogh. Lightweight detection and classification for wireless sensor networks in realistic environments. In *SenSys '05: Proceedings of the 3rd international conference on embedded networked sensor systems*, pages 205–217, New York, NY, USA, 2005. ACM Press.
4. R. Gupta and S. R. Das. Tracking moving targets in a smart sensor network. In *VTC '03: Proceedings of 57th IEEE Vehicular Technology Conference*, pages 3035–3039, 2003.
5. T. He, S. Krishnamurthy, J. A. Stankovic, T. Abdelzaher, L. Luo, R. Stoleru, T. Yan, L. Gu, J. Hui, and B. Krogh. Energy-efficient surveillance system using wireless sensor networks. In *MobiSys '04: Proceedings of the 2nd International Conference on Mobile Systems, Applications, and Services*, pages 270–283, New York, NY, USA, 2004. ACM Press.
6. J.-H. Huang, S. Amjad, and S. Mishra. Cenwits: a sensor-based loosely coupled search and rescue system using witnesses. In *SenSys '05: Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems*, pages 180–191, New York, NY, USA, 2005. ACM Press.
7. P. Juang, H. Oki, Y. Wang, M. Martonosi, L. S. Peh, and D. Rubenstein. Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with zebranet. In *ASPLOS-X: Proceedings of the 10th international conference on Architectural support for programming languages and operating systems*, pages 96–107, New York, NY, USA, 2002. ACM Press.
8. B. Thorstensen, T. Syversen, T.-A. Bjornvold, and T. Walseth. Electronic shepherd: a low-cost, low-bandwidth, wireless network system. In *MobiSys '04: Proceedings of the 2nd International Conference on Mobile Systems, Applications, and Services*, pages 245–255, New York, NY, USA, 2004. ACM Press.
9. J. W. Tukey. Box-and-whisker plots. In *Exploratory Data Analysis*, pages 39–43, 1977.
10. P. Zhang, C. M. Sadler, S. A. Lyon, and M. Martonosi. Hardware design experiences in zebranet. In *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 227–238, New York, NY, USA, 2004. ACM Press.