Implementation of the DELTA object tracking algorithm on the ESB sensor nodes

Piotr Skoczylas[1] and Markus Wälchli[2] and Torsten Braun[2]

[1] EPFL, Lausanne
[2] Institute of Computer Science and Applied Mathematics
University of Bern, Neubrückstrasse 12, CH-3012 Bern
waelchli@iam.unibe.ch

Abstract. Event tracking is one of the basic tasks of sensor networks. An intensity-based event localization algorithm was developed at the University of Bern [1]. The inaccuracy of the measurements and resource limitations in terms of energy consumption, storage capacity, processing and transmitting time make a real world implementation of this algorithm a challenging and exciting task. In this internship we investigated the real world feasibility and performance of this distributed object localization and tracking algorithm DELTA that depends on the light sensor readings. Some improvements have been made to the original DELTA algorithm to tailor it to the ESB hardware from Scatterweb [2]. Finally, evaluations and comparisons of different versions of DELTA as well as of a reference algorithm have been done.

Categories and Subject Descriptors

C.4 [**Computer Systems Organization**]: Performance of systems—*Measurement techniques*; C.4 [**Computer Systems Organization**]: Computer - communication networks—*Distributed systems*; C.2.1 [**Computer-communication networks**]: Network Architecture and Design—*Wireless communication*

General Terms

Algorithms, Experimentation

Additional Key Words

Sensor network, localization, tracking, measurement

1 Introduction

Wireless sensor networks consist of hundreds small battery-powered devices able to collect data from their surrounding and to communicate over the radio with their neighbors. A challenging task of sensor networks is that of localizing and tracking infrequent events such as fire, leakages of poisonous substances, or intruders entering buildings. In this internship we implemented and investigated the performance of the localization and tracking algorithm DELTA on real hardware using the light measurement.

From a conceptual point of view devices have very limited capacities what make sensor collaboration necessary to perform a tracking task. The ESB sensor nodes run on small batteries, have little computation power, storage capacity, and inaccurate sensor measurements. To enable long network lifetime a tracking algorithm is expected to use as little wireless communication as possible, as communication is very energy consuming. DELTA[1] involves both, object localization and tracking. Hardware limitations implied some adaptations of the original algorithm in order to tailor it to the ESB sensor nodes. Two versions of DELTA have been implemented: A basic version (see section 4.2) which may be used in less powerful sensor networks and a full featured version (see section 4.3) which requires more computational power, but is more accurate and guarantees efficient tracking in a higher percentage of time. DELTA's real world performance is compared to EnviroTrack[3] which provides similar basic features.

In the next section the related work relevant to the project is presented. In section 3 different restrictions of the ESB hardware are expalined. Taking this into consideration, in section 4 both versions of algorithm are presented. The real world evaluation and comparison is described in section 5. The paper ends with the conclusions and an outlook on future work in section 6

2 Related work

DELTA is a fully distributed object tracking framework that does not require a sink node with more computational power and memory. Therefore it does not have the disadvantage of increased data traffic between the observed area and the sink. In DELTA, groups of nodes are dynamically established around objects as soon as these objects are sensed. One of the nodes is elected as a group leader and it's first task is to inform its vicinity about its election to avoid concurrent leaders/groups. Furthermore, it gathers the sensor readings of its one-hop neighbors and uses this information to estimate the position of the moving object. The leader is also responsible to send object reports to the sink and to handover its leadership once the object leaves its sensing range.

The simulation based implementation and analysis of DELTA has been done in [4] using the Omnet++ environment[5]. The implementation of the algorithm is done in the C++ language with addition packages including Mobility Framework and matrix libraries[6]. A simulation based comparison of DELTA and EnviroTrack has been provided. The results showed DELTA to be superior to

EnviroTrack in terms of small numbers of concurrent groups, smart leader handover, etc. A real world implementation and comparison has however not yet been done.

EnviroTrack is a distributed object tracking algorithm proposed by [3], [7]. EnviroTrack supports object detection and tracking, but no object localization. A moving object is tracked by dynamically established groups of nodes. The group leader is randomly determined among all sensing nodes and broadcasts a periodic heartbeat to claim its leadership and to inform the group members about the object. The range of this broadcast is twice the sensing range. This is the main restriction of the algorithm and simplifies the leader election, group establishment and maintenance considerably. In case the assumption does not hold EnviroTrack leads to many concurrent groups. The basic operations of EnviroTrack are similar to DELTA. However, DELTA supplies a number of enhancements: DELTA determines the leaders according to their relevance what cannot be achieved with the random leader election of EnviroTrack. The restriction on the sensing range is overcomed with DELTA. DELTA provides a number of leader election factors that make DELTA more adaptive to application requirements. Finally, DELTA supports object localization and a simple broadcast protocol for efficient leader election dissemination in multi-hop network areas.

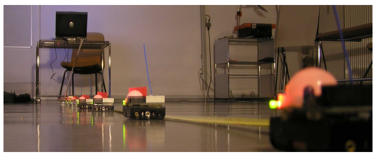
The real world experiments are done in a dark room with a multihop sensor network and a moving object emitting light from a torch. DELTA shows to be superior to EnviroTrack as it creates less groups and leaders.

3 The Hardware Issues

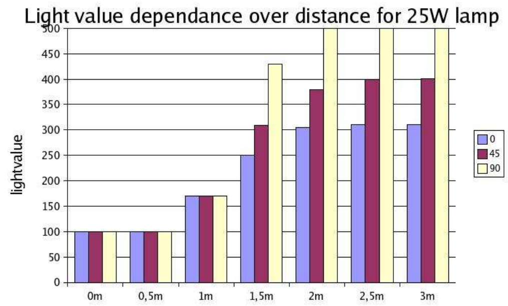
The sensor hardware used is that of the ESB sensor boards [2]. They consist of a chip with a TI MSP430 microcontroller, 2kB of RAM, 60kB flash memory, and a low power consuming radio transceiver (868MHz) operating at a transmission rate of 19.2kb/s. From a variety of implemented sensors we use the luminosity sensors. The ESB sensor boards have two main restrictions that affect the design of DELTA: The low transmission rate and the inaccuracy of sensor readings. This is mainly due to the miniaturization of the implemented hardware. The sensors have to work with at most 3V DC and should consume as little energy as possible. Furthermore, the accuracy of the sensor readings depends on the input voltage, which decreases and varies over time. Below a detailed analysis of and solutions for these restrictions are presented.

3.1 Light measurements

The luminosity sensors implemented on the ESB sensor boards provide very high variations of the light measurements of about 35%. Therefore, the original Scatterweb firmware supports only binary values returned to the application (light ON/OFF). As we need scalar values for our algorithm, this part of the firmware had to be refined. The firmware has been rewritten to combine the use of two sample values (frequency of measurements 102Hz) and a sliding window mechanism to measure the light and keep the measurement constant in highly changing environments. The average of the sliding window is send back to the application. The light value distribution achieved with that approach is depicted



(a) Test setup



(b) Light distribution

Fig. 1: Light distribution test setup and results.

in Fig. 1. The luminosity range of the sensors starts from 100 (very bright) to 500 (dark) with a granularity of 10.

3.2 Transmission speed and sending power

By default the ESB platform uses a bitrate of 19.2kbps using OOK modulation. As DELTA needs to transmit about 25 messages with 15B of data every half a second, the available bandwidth becomes scarce. This is worsened by the additional control bytes needed like the preamble, clock synchronization data, postamble, etc. For our evaluations the sensors are placed in a grid, where only nodes that are adjacent to each other should be able to communicate with each other. We thus obtain a multihop network. The transmission power of the nodes has to be adjusted to support such a mutlihop network where the nodes are placed approximately 1.5 meters away of each other. The grid is built of hexagons (see Fig. 6). Compared to a square arrangement the number of neighbors is thus reduced form 9 to 6, decreasing the number of messages to send.

In the original DELTA implementation each node starts sending at any time (ALOHA). This way of communication works however only efficiently if the sending time is less than 40% of the whole time. To deal with the scarce bandwidth we investigated the ASK modulation implemented on the ESB nodes that allows bandwidths up to 115.2kbps. The best transmission power to support multihop communication with approximately 1.5 meters distance between two nodes had been evaluated. To support the mutlihop network suggested above, the percentage of received messages has to be high for one hop neighbours and null or at least small for all others. This is not the case for the ASK modulation, where even at 58.6kbps the variation of the percentage of received messages is mucht too high. The results with a the transmission power set to 64 are shown in Fig. 2. For this reason we were forced to use a bandwidth of 19.2kbps for our

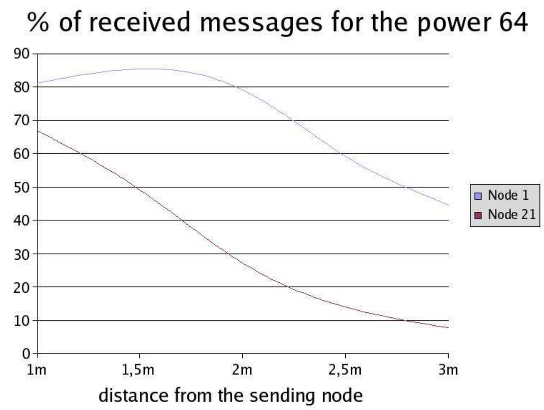
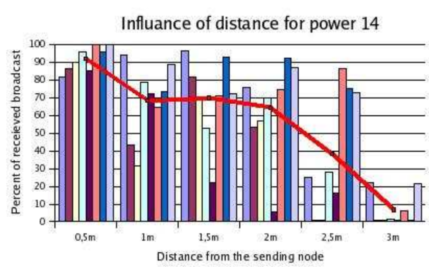


Fig. 2: Analysis of ESB at 58kbps

DELTA implementation. To fulfill less bandwidth requirement the coding and redundant control information had been reduced. The result is about 50% cut-off on the additional bytes with negligible drop of performance. Additionally, a time-slotted medium access algorithm was used (S-ALOHA) with a slot time of 23.5ms during heartbeat and 24ms during leader re-election phase (Please refer to Section4 for more details). With the OOK modulation the transmission power has been evaluated to be 14 to support our network setup (see Fig. 3).

(a) Test setup



(b) Transmission power

Fig. 3: Transmission power evaluation.

3.3 Available memory and sensor variation

The Intensity-Based Event Localization Algorithm[1] used in DELTA need the usage of advanced matrix computation and exact sensor reading values. The values we received from the light sensors had to high variations and the matrix implementation is too resource consuming to make the original algorithm run on the ESB sensors. We therefore implemented a simple and less power consuming approximation algorithm based on the interpolation of coordinates. The details are available in section 4.

4 Implementation of DELTA

The basic communication scheme of DELTA is shown in Fig. 4. During the

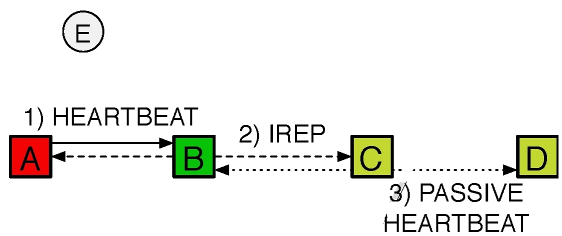


Fig. 4: Message flow of DELTA from leader(red) to member(green) and passive member(yellow)

implementation it became obvious that some features need to be adapted to apply to the ESB hardware.

4.1 Changes made to the original DELTA algorithm

Node internal states The states were enhanced with a "passive member" state which occurs when a node is in idle status, but is receiving passive heartbeat. In [4] this information is kept as an internal variable and a node is still seen as idle, but is expected to behave differently when it starts sensing light or receives a leader re-election message. To clarify this ambiguity the "passive member" state was introduced. The expanded state diagram of DELTA is presented in Figure 5.

Leader election delay As described in section 3.2 we use S-ALOHA to access the medium. Thus, the leader delay has to be an integer pointing to the slot number in which the leader re-election message is sent. 30 slots of 24ms were chosen. The delay is calculated as a sum of 4 parts. First is a value from 0 to 18 slots, which is based on the value of the light. Second - a value from 0 to 18

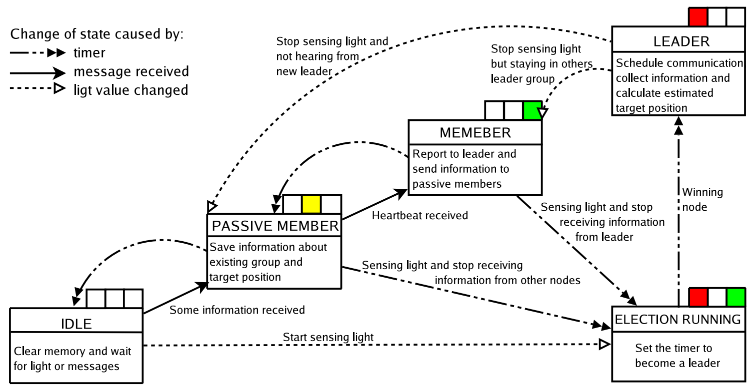


Fig. 5: State diagram for ESB implementation of the DELTA node's roles.

slots based on the node position (for DELTA 1 this is always 0, as there is no target estimation). Third - 0 to 3 slots depending on how many times the node has been leader in the past. Finally, a random value of 0, 1, or 2 slots is added. If the sum is bigger than the number of slots, it it truncated.

Sending the requested information to the leader A reply delay based on the sensor readings is not feasible for two reasons: First, only 20 slots of 23.5ms are available. Second, the light value sensed on different nodes might be the same due to the high granularity needed to classify the sensor readings. An equal delay for two different nodes would lead to collisions and data loss. As in the previous subsection the delay is calculated as a sum of partial delays. It is up to 7 slots depending on the value of the light, up to 6 slots depending on the distance to the leader (again only for DELTA 2) and up to 2 random slots. Concerning the hexagonal grid this does not prevent two nodes from sending at the same time. A sending schedule solution is described in next subsection.

Sending schedule Each node saves information about its neighbors. Each time a node receives information from a not yet registered node it updates its list until the list contains ten entries. Each time a node enters idle state the list is emptied. Theoretically, all nodes in the hexagonal grid should have at most 6 neighbors. As a node sometimes receives also the information from other nodes it dynamically changes the entries in its table according to the usefulness of the received information (in our implementation the value of the light). This schedule is being send by the leader in its heartbeat. All nodes receiving the heartbeat either reply according to schedule sent via the heartbeat, or calculate their delay as described in previous subsection plus the delay needed to send all scheduled messages.

4.2 DELTA 1

This is a basic implementation of DELTA where only heartbeats, passive heartbeats and leader re-election messages are sent. With the passive heartbeat messages DELTA is able to work when the ratio of the sensing range and wireless communication range $\frac{\mathrm{SR}}{\mathrm{CR}}$ of a node gets closer to one or even above it. Thus, DELTA overcomes the requirement of $\frac{\mathrm{SR}}{\mathrm{CR}} < 1$ or even $\frac{\mathrm{SR}}{\mathrm{CR}} < \frac{1}{2}$ other approaches like EnviroTrack suffer from.

Another enhancement compared to EnviroTrack is the mechanism of calculating the delays based on the intensity of the sensor readings. Moreover, the usage of the history how many times a node has already been leader is also new. The assumption behind this is that a node that already was a leader works correctly and is therefore to favor. The computation based on the sensor readings and the history increase the probability that the node best qualified to track the object as long as possible is elected. This reduces the number of leader handovers and minimizes the time when a not-correctly working node elects itself as leader.

4.3 DELTA 2

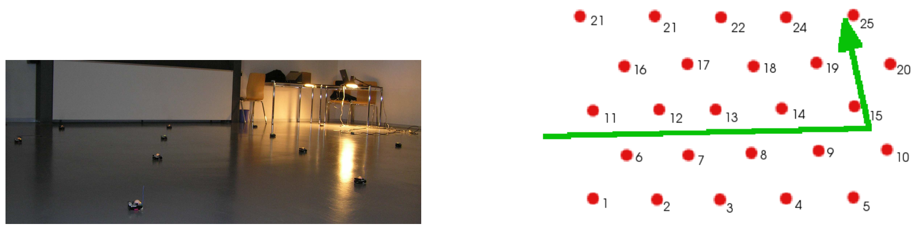
In this version of DELTA the passive heartbeats send by the neighbor nodes of the leader are used to carry their light information. This information is used by the leader to estimate the position of the event and to enable smooth leader handover.

In order to work, nodes running DELTA 2 need to know their position and the positions of their neighbors to localize an event. Additionally, the position information is used to calculate the delay to send messages as explained in subsection 4.1. The event location is computed using the light values sensed by the nodes around the leader. The target position is estimated using the neighbor positions with weights assigned to these positions and deriving the event location as an interpolation of that data. The smooth handover is an application of the estimated target position.

As explained in section 4.1 each node maintains a table of ten surrounding nodes. When the target position is estimated the leader node is able to determine the closest node basing on the position information he collects while receiving the messages from sourrounding nodes and may handover the leadership to that node. Therefore the position of the next leader is included in the leader re-election message when the event leaves the sensing range of the leader. All nodes not being marked as new leaders stay or enter in the member state. Only if the prospective leader does not send out heartbeat for a given time a new leader election is started.

5 Evaluation

This section describes the analysis of the data gathered in the tests performed from 25.09.2006 to 06.10.2006 at University of Bern. The real world experiments have been done in a dark room with a multihop sensor network and a moving object emitting light from a torch. DELTA shows to be superior to EnviroTrack as it creates less groups and leaders. The network setup is depicted in Fig. 6



(a) Tracking experiment

(b) Node placement and movement pattern

Fig. 6: Tracking experiment setup.

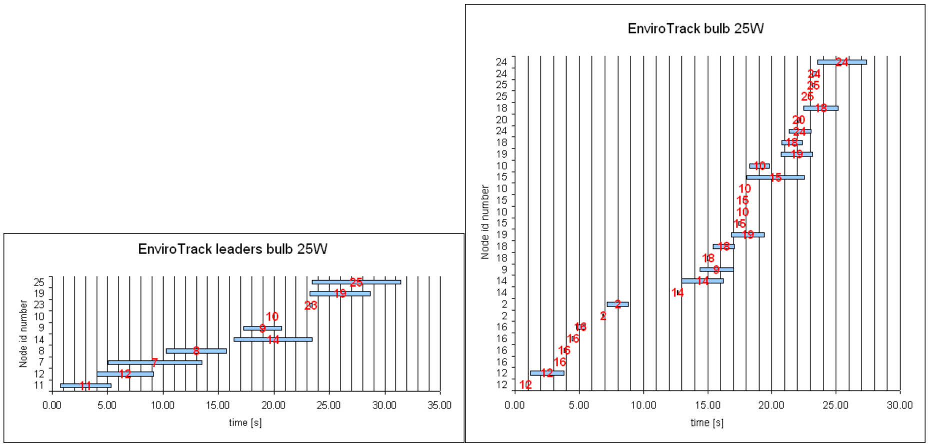
The room was a seminar room darkened by blinds. All the desks and chairs were removed in order to make a free space of 8x6 meters. The lamp used was a desktop type lamp with two kinds of bulbs with a power of 25W and 40W, respectively. Thus some diversification of light sources was included in our tests. We moved the lamp as presented in Fig. 6 (green line) with different speeds. The path was finished within 15 to 35s. The grid used is hexagonal built from 25 nodes with a distance of 1,5 meters between any two nodes. The hexagonal grid is depicted in Fig. 6.

As the ESB nodes have only 4KB of EEPROM memory available to the user only the most important data could have been saved. The solution was to save only the changing states, time stamps of the event as well as the number of sent and received messages.

5.1 EnviroTrack Results

The tests done with the 25W bulb already show many problems of EnviroTrack. First of all, from Fig. 7 we can see that there are multiply leaders (parallel bards). This is due to the fact that the restriction of the ratio of the sensing range and the communication range $\frac{\mathrm{SR}}{\mathrm{CR}}$ being smaller than $\frac{1}{2}$ does not always hold.

Heartbeat collisions become more and more frequent the more leaders there are, as all leaders compute their delay randomly. This leads to unnecessary resources consumptions and data loss. Comparison of the missed messages due to communication error or messages send at the same time is presented in Figure 8



(a) Leaders (b) Members

Fig. 7: Sample EnviroTrack test results with 25W bulb

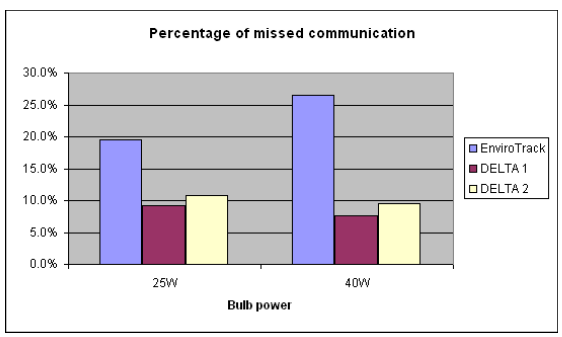
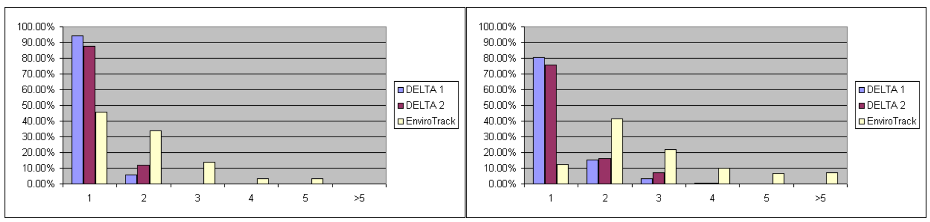


Fig. 8: Statistical overview values of how many messages were not heard.

To keep group coherence is rather difficult with EnviroTrack for such environments as members go to idle state and later elect themselves as leader if they did not hear a heartbeat message from the leader for a certain time. The fluctuations are shown in Figure 7 for example for the nodes 15 or 16. The results gained for the 25 W bulb remain nearly the same for the 40 W bulb. Results concerning the 40 W bulb can be found in the Appendix 8.

5.2 DELTA 1 Results

Test results show that even for the 40W bulb (higher sensing areal) DELTA 1 prevents multiply leaders from being present for more than a fraction of a second (see Fig. 9). In Fig. 8 it is shown that the number of missed messages is much



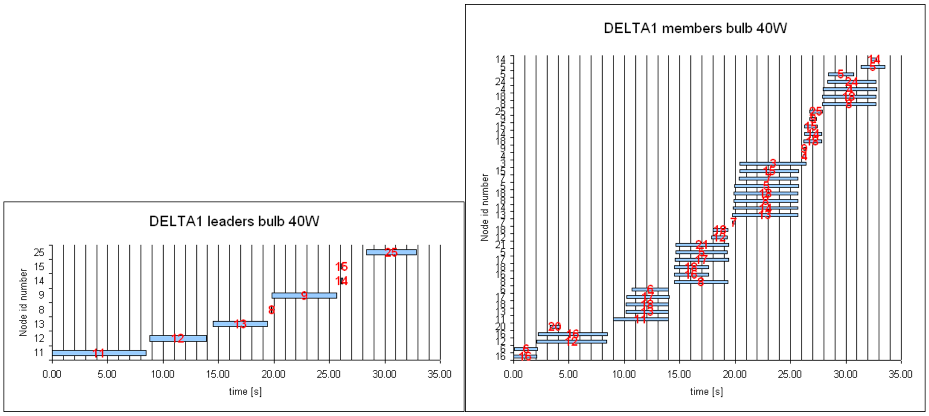
(a) 25W bulb (b) 40W bulb

Fig. 9: Number of leaders on average depending on the algorithm and bulb used in the test

smaller compared to EnvrioTrack. The results are reflected in Fig. 10. It shows that for most of the time there is only one leader. Groups are built around it and the group members tend to remain in their group, instead of swaping state all the time.

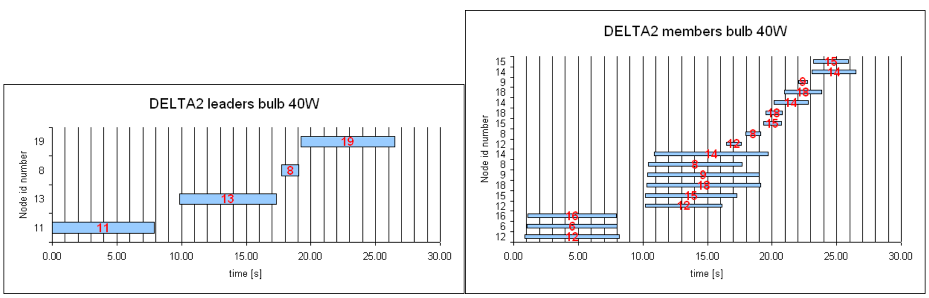
5.3 DELTA 2 Results

Delta 2 has additional features to DELTA 1 such as the leader handover described in Section 4.3(Examples are the transitions from 13 to 8 and than to 19 in Fig. 11). Due to imprecise sensor readings the position cannot be calculated optimally (for example the optimal transitions in the example from Fig. 11 are from 13 to 14 and than to 19). Sometimes the readings are so imprecise that a node that will not sense the light by the time of handover is selected as the next leader. In such cases (e.g. transition from 11 to 13 in Fig. 11) a new leader election has to start. Currently, the target position information is too bad to improve the DELTA algorithm. Accordingly, nodes can become leader at the same time, because one node is supposed to be closest to the target and another



(a) Leaders (b) Members

Fig. 10: Sample DELTA1 test results with 40W bulb



(a) Leaders (b) Members

Fig. 11: Sample DELTA2 test results with 40W bulb

node is really closest to the target. That is why DELTA 2 seems to have even less performance than DELTA 1.

6 Conclusions and future work

The main advantages of the DELTA algorithm are the prevention of multiply leaders, and a more efficient object tracking. The theoretical results proved to be true in a real-world testbed. However, with the limited hardware of the ESB sensors it is difficult to explore all the benefits of the algorithm. Some work that might be done in the future includes the implementation of the improved passive heartbeat dissemination scheme proposed in [1]. Furthermore, hardware with more accurate sensors and more memory to implement the multilateration could be looked for. An improved version of DELTA 2 might be tested and compared to DELTA 1.

7 Acknowledgments

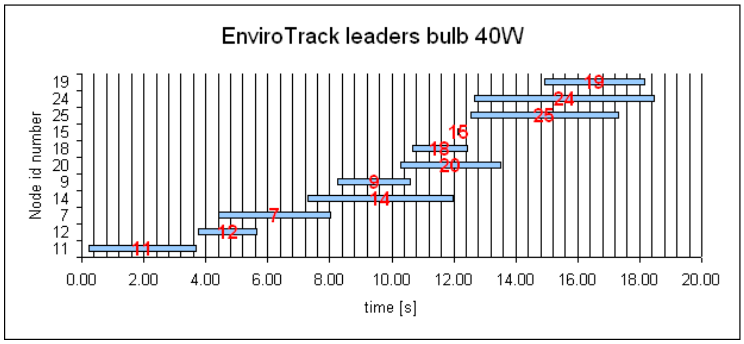
I would like to thank Professor Torsten Braun for his contribution to the final report, Markus Waelchli for his help and advices during all parts of the project, Thomas Bernoulli for participation during the tests and data analysis and Markus Anwander for explaining the details of transmission speed adjustment on the ESB platform. I am also thankful to the National Competence Center in Research on Mobile Information and Communication Systems (NCCR-MICS), a center supported by the Swiss National Science Foundation under grant number 5005-67322 that supported this exciting and challenging summer internship.

References

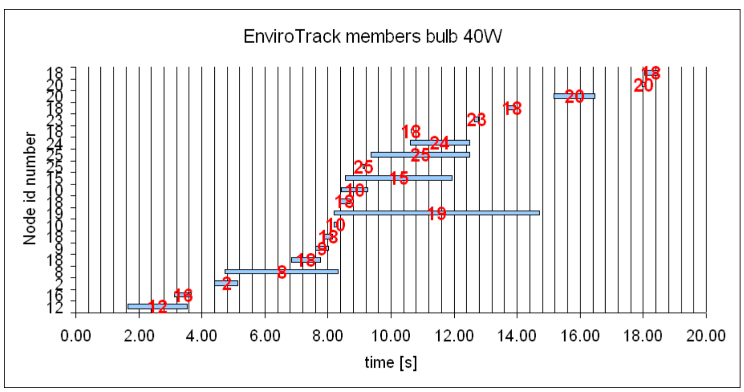
1. Wälchli, M., Scheidegger, M., Braun, T.: Intensity-based event localization in wireless sensor networks. In: Proceedings of IFIP Third Annual Conference on Wireless On Demand Network Systems and Services (WONS'06), Les ménuires, France (2006)
2. Scatterweb: Platform for self-configuring wireless sensor networks (2006) http://www.scatterweb.net.
3. Abdelzaher, T., Blum, B., Evans, D., George, J., George, S., Gu, L., He, T., Huang, C., Nagaraddi, P., Son, S., Sorokin, P., Stankovic, J., Wood, A.: Envirotrack: Towards an environmental computing paradigm for distributed sensor networks. In: Proc. of 24th International Conference on Distributed Computing Systems (ICDCS), Tokyo, Japan (2004)
4. Meer, M.: The delta object tracking and localization algorithm for sensor networks. Master's thesis, University of Bern (2006) pending.
5. Varga, A.: Omnet++ discrete event simulation system (2006) http://www.omnetpp.org.
6. Andreaskoepke Hkarl, L.P.: Mobility framework for omnet++ and matrix template library (2006) http://mobility-fw.sourceforge.net/, http://osl.iu.edu/research/mtl.

7. Luo, L., Abdelzaher, T.F., He, T., Stankovic, J.A.: Envirosuite: An environmentally immersive programming framework for sensor networks. ACM Transaction on Embedded Computing System (TECS) **V** (2006) 1–31

8 Appendix



(a) Leaders



(b) Members

Fig. 12: Sample EnviroTrack test results with 40W bulb.