

Scheduling data transmissions in wireless sensor networks used for position tracking

Péter Györke / Béla Pataki

Received 2013-04-03, revised 2013-11-28, accepted 2014-03-26

Abstract

In wireless sensor networks developed for ambient assisted living (AAL) applications, power supply is one of the most challenging problems. To extend the duration of the operation of a wireless sensor node, efficient utilization of the available energy can be achieved with low duty-cycle operation, where the sensor node is in sleep mode most of the time. In the case when measurements have low cost, a method is proposed for decreasing the usage of the most energy consuming mode (communication) by handling the measured data locally. In AAL applications the position tracking of a person is an essential task, and it is a good demonstrative example for showing the solution principles. Position tracking with motion sensors requires a high number of messages and most of them are caused by local movements. Our suggestion is to eliminate these messages. The method is based on a Hidden Markov Model of the motions of an observed person. The model provides information based on the estimated global state of the system, which is the position of the person in the space of interest. This state can be forwarded to the nodes so they locally perform the filtering to save valuable energy by not transmitting messages which are not relevant.

Keywords

Wireless Sensor Networks · Energy Harvesting · Hidden Markov Models · Ambient Assisted Living

Péter Györke

Department of Measurement and Information Systems, Budapest University of Technology and Economics, Magyar Tudósok krt. 2., H-1117 Budapest, Hungary
e-mail: gyorke@mit.bme.hu

Béla Pataki

Department of Measurement and Information Systems, Budapest University of Technology and Economics, Magyar Tudósok krt. 2., H-1117 Budapest, Hungary

1 Introduction

Scheduling the measurements and data transmissions is required in energy constrained wireless sensor networks (WSN's). The wireless sensor nodes have typically limited energy; they are operating from batteries, because the wiring required to power the sensor network from the local power lines is usually difficult in such applications. Extending the life time of a power source with energy harvesting power supply (e.g. solar-cells) is possible [2, 4], but using energy harvesting in indoor environments is still questionable.

The energy consumption can be divided into two parts:

- 1 the energy consumption of the hardware components;
- 2 the energy consumption resulting from the operation modes, which depends on the software control of the nodes.

The operation modes are the following:

- 1 measurement;
- 2 communication;
- 3 sleeping.

Each mode has different energy consumption. In some cases the measurement has similar or less consumption compared to the consumption of the sleep mode, but in nearly all the situations the communication is the critical phase.

According to a current survey paper [1] many previous works deal with routing and clustering problems and a couple of researches are aimed at ensuring maximal data throughput over the wireless network considering the energy limitations. Routing means that some of the sensor nodes are out of the range of the central unit, so their messages are transmitted through a router unit. Clustering means the sensor nodes are in groups, and each group has a router which communicates with the central unit. In ambient assisted living (AAL) applications the most common topology is the star-shape (see Fig. 1), and in most cases routing is not necessary because all of the nodes are in one small area (e.g. in an apartment) so the nodes can send their measurements directly to the central unit. Therefore our approach is different; star topology networks are investigated for the optimization of

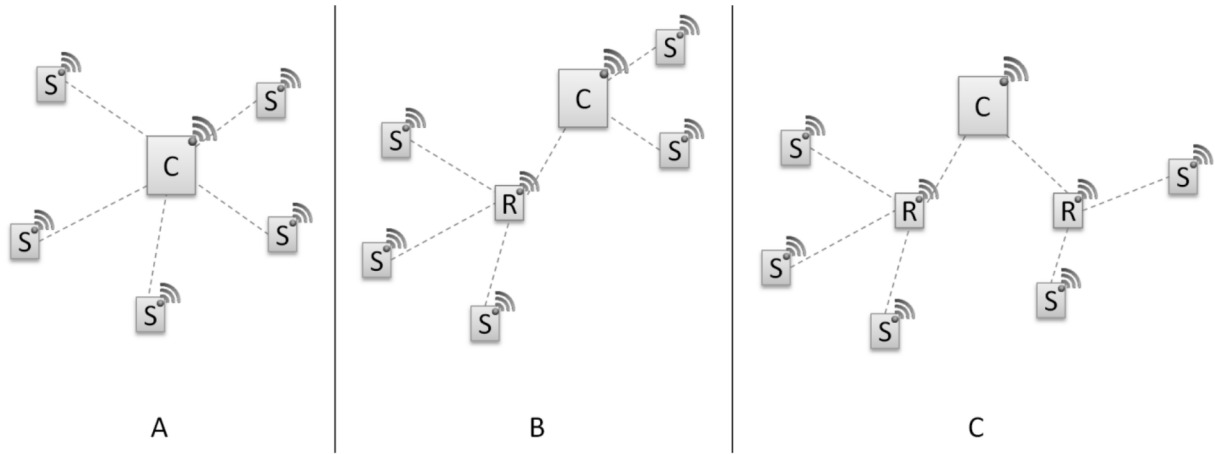


Fig. 1. Typical WSN topologies: “A” – Star topology; “B” – Tree; “C” – Cluster tree. (C – Central unit; S – Sensor node; R – Router)

energy consumption by applying measurement and communication scheduling.

The typical operation of this kind of wireless network is the following: the central unit is in receiving mode initially and the sensors are in sleep mode. If a sensor wakes up and transmits a message, the central unit will receive it and answer in an acknowledgement (ACK) message. Meanwhile the sensor switches to receiving mode to be able to receive the ACK message (see Fig. 2).

In [6] local scheduling methods are presented where the sampling rates of the sensors were adjusted according to the measured signal’s estimated frequency, and an algorithm is shown how to adjust the sampling rate. In many cases collaboration of the sensors and global scheduling is required; e.g. [7] shows a data-driven sensor selection method for power optimization based on the actual measurements and the capabilities of a particular sensor. Sensor selection is addressed in [8]: a quantitative measure is proposed to represent the actual utility of a sensor based on a Bayesian approach, and the selection can be performed according to this measure.

Dynamically configuring every sensor (e.g. changing a particular sensor’s sampling interval based on the actual (measured) state of the observed system) gives a possibility to save energy. In this case sensors or the central controlling unit can take the global state of the system into account; to estimate this global state, modeling of the observed system is necessary. The basic idea of modeling with Hidden Markov Models was suggested in [9, 10]. The HMM of the system can provide probabilistic information about the system’s state which cannot be observed directly. Papers [11, 12] have shown adaptive sampling methods for tracking, based on the system state. Scheduling of data transmissions can positively affect the wireless network’s performance. In [13] a method is shown for transmission scheduling based on the wireless channel condition. The channel condition can be measured e.g. by using Received Signal Strength Indication (RSSI), or measuring the packet loss ratio. A similar approach is suggested in [14] especially for small-scale WSNs.

In this paper an adaptation of these methods will be presented. Motion sensors are widely used and cheap devices, and although the original purpose of them are not position tracking, if the necessary spatial accuracy is not very high (as it is the case in several AAL applications), they could be used for that purpose as well. This is investigated as an example application showing most of the problems mentioned. The main goal is still to reduce energy consumption, therefore extending the operation time (battery-cycle) of the sensors in the network.

In the presented application it is assumed that only one person is in the observed space. The tracking of multiple persons with PIR sensors needs further research (only approximations are possible), nevertheless the suggested energy saving scheduling could be demonstrated in the one person case as well.

In the following the case is investigated when the consumption of the measurements is irrelevant compared to the whole energy consumption of the sensor node during the time. Using a different PIR (Passive Infra-Red) motion sensor with very low operation consumption in the sensor node gives the possibility of continuous measuring. In this case, sampling the measurements is not necessary, but it is important to filter the messages because they may contain information which is irrelevant for the position tracking. In the sequel, basic hardware related principles will be presented. In section 3 motion detection and measurement filtering strategies will be presented. In section 4 a novel method is presented to extract position information from the motion measurements. After that in section 5, a new method is proposed to reduce the number of data transmissions by maintaining the error of position tracking. Finally the results of the new method will be analyzed. The results are simulated, but the initial data for the simulation is real-life measured data.

2 Hardware context

The sensor node comprises 3 main components: 1) microcontroller (in our demonstrative cases ATMEGA88PA was used); 2) RF module (RFM12B); 3) PIR (Passive Infra-Red) motion sensor (EKMC1601111 (case 1) or the low power version:

EKMB1201111 (case 2). The software of the microcontroller enables and disables these components, which determines the power consumption. The values in Table 1 are typical, being based on measurements of this particular sensor node.

Tab. 1. Energy consumptions of a sensor node in different modes. Case 1 is when the measurement has a significant consumption; in case 2 this is negligible.

Operating mode	Case 1	Case 2
Measure	0.2 mA	0.002 mA
Sleep	0.01 mA	0.01 mA
Communicate	12 mA	12 mA

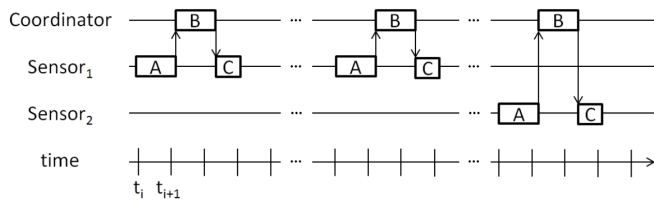


Fig. 2. Sequence diagram of the communication between the coordinator and two sensors. The durations are around these values: $T_A = 10$ ms, $T_B = 20$ ms, $T_C = 1$ ms. Between these bursts usually many seconds elapse.

According to Table 1, in case 1 the measurements (and the communication) are to be scheduled, because in both modes relatively high consumption will occur. In case 2 only the communication is to be scheduled in order to achieve lower energy consumption. In case 1 and 2 only the motion sensing element is different, the price of the sensor used in case 1 is significantly smaller than the other's price. In other ways the two sensors are identical.

In our previous work [3] we investigated case 1 (see below). Scheduling of the measurements highly depends on the measurement type. For example a temperature monitoring sensor network does not have to be so reactive as a typical motion sensor network because rapid temperature changes are rare, and delayed transmissions are not an issue in particular cases. It should be emphasized that the consumption of communication is roughly the same in receive and in transmit mode, so receiving data also has a high cost. For this reason, when the sensors are in sleep mode, communication towards the sensors is not possible.

Therefore the reconfiguration of the sensors (setting of sampling rates) is only reasonable in the acknowledgement message after a successful transmission from the sensor. See Fig 2. where "A" means: a motion event happened or a scheduled transmission is done by the sensor. "B" means: coordinator processes the new data and calculates the next wake-up time (and/or measuring time) for this sensor and sends acknowledgement. "C" means: the sensor processes the acknowledgement and goes into sleep mode for the given time. This operating principle reduces the system's reactivity when an unforeseen new event happens, which is usual in motion sensor networks.

In case 2, where the energy consumption of the measurement is low, and therefore the measurement can be continuous, the system can react to motion events. If the data is analyzed off-line (e.g. in every hour, because a single data point cannot be evaluated) the nodes can aggregate messages and the transmission of the aggregated messages can be done periodically. In this case the energy usage can be calculated in advance. If the data is volatile (e.g. security system, lighting control based on motion) the transmission cannot be delayed, so the scheduling of the transmissions has to be done in real time. The number of transmissions will depend on the measured data, so it is possible that the different sensors will consume different amounts of energy.

In our previous work [3], case 1 was examined when the measurements have considerable costs. A capability of taking sampled measurements by switching the sensor on and off was assumed. When a sensor is turned off, only the internal timer can wake it up, so if an external event happens (e.g. motion) the sensor will miss that event. This operation principle can cause considerable degradation in the precision of the motion tracking. A method for controlling the sampling intervals by maintaining the measurement precision and also considering the energy conditions was proposed. The method was based on a Hidden Markov Model which modeled the system to be measured. The area where the system is operated was divided into several subsections. The model calculates the probability of a new event in every subsection based on the previous measurements and the system model. Based on this probability vector a cost is assigned to every sensor:

$$C_{i,j,k} = \frac{1}{R(i,j)} + \frac{\alpha}{P_0 + P_S(i,k)} \quad (1)$$

where $i \in \{1, 2, \dots, m\}$ (sensor index), $j \in \{1, 2, \dots, n\}$ (index of the subsection with the highest probability calculated by the HMM), $R(i, j)$ is a relevance function, which gives how relevant a sensor in a particular subsection of the area. The reader should note that $R(i, j)$ is independent of the k time. The relevance function can be substituted with a relevance look-up table which is used in the following. In this look-up table (matrix) all the relevance values are higher than zero; therefore the reciprocal values are always finite. $P_S(i, k)$ is the actual power stored in the i th sensor node at time k . P_0 is used to avoid the possible zero value of the denominator. In (1) α is a parameter which sets the ratio between the sensor's relevance and the power condition. If α is low the relevance has a higher influence in the cost. This means higher precision, but ignoring the sensor's power conditions may lead to premature depletion of the batteries. If α is high, the power condition will define the cost, therefore the precision will decrease dramatically.

The sampling intervals for each sensor are assigned based on the cost. For the sake of simplicity only two values of sampling interval were assigned: a short interval for the l sensors with the lowest cost, and a long interval for every other sensor. This re-

sulted in the following behavior: a location in the observed area is selected where the possibility of changes in the state is high. The sensors around this area are set to a short sampling interval, so the system remains reactive to new events. Meanwhile the other sensors can save energy by using a longer sampling interval. The proposed method was tested with simulation of a motion sensor network. The results have shown that the battery-cycle can be considerably extended owing to the reduced power usage. However, a degradation in precision is inevitable because of the under sampling of the motions.

3 Position tracking with motion sensors

In the introduction the idea of position tracking or estimation with motion sensors is mentioned. This kind of position tracking differs from the traditional position tracking where the goal is the estimate the position coordinates of the observed object. With motion sensors exact coordinates cannot be determined, however the region where the motion of a person has happened is known from the sensor data. The spatial resolution of the position depends on the number of sensors (e.g. one sensor in a room can detect that the person is in the room, 4 sensors in a room can detect the person in several subsections of the room, in [3] we investigated the 9 subsections/room case). The advantage of this method is that there is no additional hardware which has to be worn by the observed person.

In the actual case the motion sensor is continuously measuring and the sensor node should decide whether to transmit a message or not. This decision is based on the motion sensor's output. A simple timeout (t_{end}) is used to filter the raw motion data (see Fig 3): the "motion finished" message is sent only after not having new motion signals after the last one in t_{end} time. The selection of the parameter t_{end} is essential. If t_{end} is short, there will more transmissions (more energy is used). If t_{end} is longer, the detection when the motion ends will be more inaccurate. So in this case t_{end} acts similar to sampling time in the former case (when the measurements are rare and scheduled). It should be optimized for two contradictory requirements: energy usage and measurement precision. In a real world scenario t_{end} cannot be set to the best value, because the best value depends on the motion of the person under observation; only approximate values can be used and adaptively set to enhance the performance of the system. The t_{end} parameter should be longer for a sensor which is related to a zone where the observed person is staying for a longer time. To determine the actual position of the observed person a method is proposed in the following section. If the sensor's output is disturbed by random impulse noise, filtering should be applied also before sending the motion start message. This filtering is out of the scope of this paper.

4 Modelling the motions of the observed person

Extracting the position information from the sensor data is not straightforward. Here a probabilistic model is proposed – a Hidden Markov Model (HMM) was chosen. The time series of

a person's movement fulfills the Markov criteria, so the process can be considered as a Markov Process [9]. The proposed model can be described with the following notations:

- k is the (discrete) time,
- N is the number of states (possible spatial positions of the person),
- M is the number of sensors,
- L is the number of symbols provided by the sensors,
- $\mathbf{x}_k = [x_k^1, x_k^2, \dots, x_k^N]^T$ is the state probability vector at time k ,
- $\mathbf{X}_{1:k} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_k\}$ is a matrix built from all state probabilities from the beginning to the current time instant,
- $\mathbf{s} = \{s_1, s_2, \dots, s_M\}$ is the set of sensors,
- $\mathbf{o}_i = \{o_i^1, o_i^2, \dots, o_i^L\}$ is the finite set of symbols the i th sensor can provide (e.g. o_i^1 means motion, o_i^2 means no motion, o_i^3 means no measurement, $i = 1 \dots M$),
- $\mathbf{y}_k = [y_k^1, y_k^2, \dots, y_k^M]^T$ is the observation vector of the sensors' output values at time k ,
- $\mathbf{Y}_{1:k} = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_k]$ is a matrix containing all observation vectors from the beginning to the current time,
- $\mathbf{e}_k = [e_k^1, e_k^2, \dots, e_k^N]$ is the output of the sensor model, which connects the observations with the states of the system (see equation (4)).

The elements of a state vector will represent the zones in the observed area. The value of a single element from the vector describes the probability that the person is in the zone:

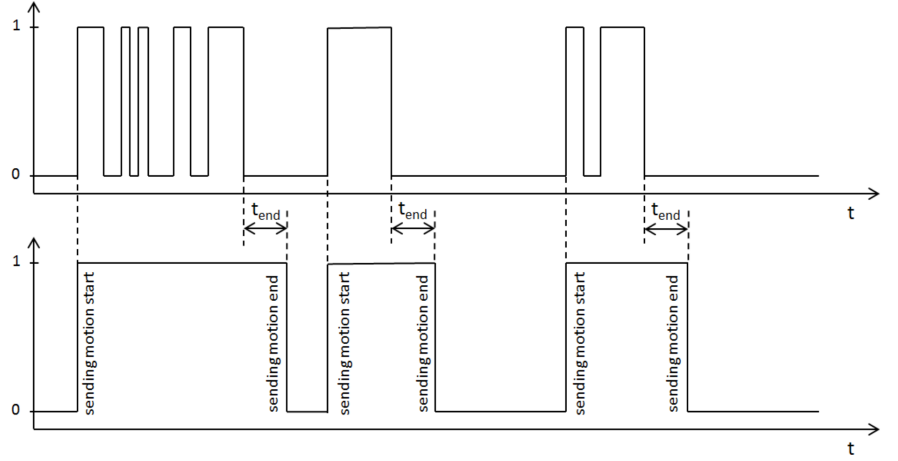
$$\mathbf{x}_k = \begin{bmatrix} x_k^1 = P(\text{pos} = 1 | \mathbf{Y}_{1:k}) \\ x_k^2 = P(\text{pos} = 2 | \mathbf{Y}_{1:k}) \\ x_k^3 = P(\text{pos} = 3 | \mathbf{Y}_{1:k}) \\ \vdots \\ x_k^N = P(\text{pos} = N | \mathbf{Y}_{1:k}) \end{bmatrix} \quad (2)$$

$$\sum_{n=1}^N x_k^n = 1, \text{ for every } k \quad (3)$$

where k is the time index, $P(\text{pos} = i | \mathbf{Y}_{1:k})$ means the probability of the person is in the "i" coded position (see Fig. 4) if the measurements were $\mathbf{Y}_{1:k}$. With the assumption of (3) a constraint is introduced: in zones $[1 \dots N]$ only one person can be observed.

\mathbf{x}_k is the probability vector, the elements are given by the probabilities of the person being in the different subsections at time k . The HMM model is described by the state transition probabilities $p(\mathbf{x}(k+1) | \mathbf{x}(k))$, and the measurement characteristics (which depends on the topology of the sensors and the measurement noise). The $p(\mathbf{x}(k+1) | \mathbf{x}(k))$ probabilities could be learned during a training phase. The sensor model should be defined for the HMM. The model stores information about

Fig. 3. Filtering the motion sensor's output locally. Only if there is no motion for t_{end} time the "motion end" message will be sent. The data is generated for this demonstrative example. In the upper chart 1 on the vertical axis means motion is present, 0 means no motion. In the lower chart 1 and 0 means respectively that the last message was motion start or motion end. In both charts the horizontal axis (t) means the time.



Tab. 2. Conditional probabilities for the sensor model as a look-up table.

s	$y_k^1 = o_1^1$	$y_k^1 = o_1^2$	$y_k^1 = o_1^3$...	$y_k^M = o_M^1$	$y_k^M = o_M^2$	$y_k^M = o_M^3$
1	$P(y_k^1 = o_1^1 pos = 1)$	$P(y_k^1 = o_1^2 pos = 1)$	$P(y_k^1 = o_1^3 pos = 1)$...	$P(y_k^M = o_M^1 pos = 1)$	$P(y_k^M = o_M^2 pos = 1)$	$P(y_k^M = o_M^3 pos = 1)$
2	$P(y_k^1 = o_1^1 pos = 2)$	$P(y_k^1 = o_1^2 pos = 2)$	$P(y_k^1 = o_1^3 pos = 2)$...	$P(y_k^M = o_M^1 pos = 2)$	$P(y_k^M = o_M^2 pos = 2)$	$P(y_k^M = o_M^3 pos = 2)$
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
N	$P(y_k^1 = o_1^1 pos = N)$	$P(y_k^1 = o_1^2 pos = N)$	$P(y_k^1 = o_1^3 pos = N)$...	$P(y_k^M = o_M^1 pos = N)$	$P(y_k^M = o_M^2 pos = N)$	$P(y_k^M = o_M^3 pos = N)$

how sensors are related to different zones, and the sensors' actual output modifies the related zone (or zones) probability. In our case three possible symbols are provided by the sensors: o_i^1 means motion, o_i^2 means no motion, and o_i^3 means no measurement ($L = 3$).

In Table 2 the conditional probabilities are given : $P(y_k^m o_m^j | pos = n)$ is the probability that m th sensor transmits symbol o_m^j when the person is in position (subsection) n . These conditional probability values should be measured with statistical methods a priori.

The input of the sensor model is observation (\mathbf{y}_k) vector, and the output $\mathbf{e}_k = \{e_k^1, e_k^2, \dots, e_k^N\}$ gives the probabilities of the zones (states) based on the prior probability values and the current measurements only. It is calculated using the following formula, where o_m^j is the actual output symbol given by the m th sensor in time k :

$$e_k^n = \frac{\sum_{m=1}^M \sum_{j=1}^L \delta_{mj} P(y_k^m = o_m^j | pos = n)}{\alpha} \quad (4)$$

$$\text{where } \delta_{mj} = \begin{cases} 1, & \text{if } y_k^m = o_m^j \\ 0, & \text{else} \end{cases} \quad (5)$$

and α in the denominator is a normalization factor. The nominator of equation (4) sums the probabilities of all sensors' actual symbols supposed that the person is in subsection n . E.g. if we have 3 sensors and the first one gives "motion" signal (o_1^1), the second "no measurement" (o_2^3), the third gives "no motion" (o_3^2) at time k ; and the person in position 5 case is investigated, then the nominator gives

$$e_k^5 = P(y_k^1 = o_1^1 | pos = 5) + P(y_k^2 = o_2^3 | pos = 5) + P(y_k^3 = o_3^2 | pos = 5).$$

The general approach known as POMDP (Partially Observable Markov Decision Process) [10] is used for estimating the probability of the actual state of the system. The estimation of the unobservable variables can be obtained using the recursive formula:

$$p(\mathbf{x}_{k+1} | \mathbf{Y}_{1:k+1}) = \alpha \cdot p(\mathbf{Y}_{k+1} | \mathbf{x}_{k+1}) \cdot \sum_{\mathbf{x}_k} p(\mathbf{x}_{k+1} | \mathbf{x}_k) \cdot p(\mathbf{x}_k | \mathbf{Y}_{1:k}) \quad (6)$$

where α is a normalization factor that ensures that the sum of the probabilities of the states will be 1. (Note that \mathbf{x}_k here is the state probability vector which was defined at the beginning of that section. It serves as the state vector of the HMM.) We have to calculate the probability of the next state in two cases. If there are new measurements:

$$\mathbf{x}_{k+1} = \alpha \cdot \text{diag}(\mathbf{e}_k) \cdot \mathbf{A}^T \mathbf{x}_k \quad (7)$$

where the probability of the new measured information is represented by \mathbf{e}_k ; this is calculated from \mathbf{Y}_k and the sensor model (equations (4) and (5)). If there are no new measurements, we can predict one step according to the state transition matrix:

$$\mathbf{x}_{k+1} = \alpha \cdot \mathbf{A}^T \mathbf{x}_k \quad (8)$$

The state transition matrix can be defined with a statistical approach based on the collected data. The initial \mathbf{A} matrix will contain information about the layout of the zones (e.g. the transition probability between neighboring zones is higher than the transition probability between the zones far away from each

other). During the learning stage of the \mathbf{A} matrix some transition probabilities will be increased, and others will be decreased based on actual measurements of the sensors. After the learning phase the \mathbf{A} matrix will show the most probable routes of the person in the observed area.

5 Distinction of position changes from other movements

Most of the data transmissions from sensors are initiated because of local movements (e.g. leaning back in a chair, typing on keyboard). These movements do not cause position changes, but the motion sensors still detects them. The sensor node should make a distinction between these motions. For this decision locally available information is not enough; the sensor node should know about other sensor states (previous output values), or the coordinator can send the believed position of the observed person to the sensor nodes, based on the output of the HMM.

As shown in Figure 4. there are many transmissions which are not necessary for position tracking, so valuable energy can be saved without the degradation of the precision. The result of the estimation of the actual position of the observed person will be the zone with the highest probability in the state vector of the previously introduced HMM. If the probability reaches a threshold (or the zone becomes the most probable location for the observed person) the sensor in the zone should use a longer t_{end} (see Section 2.) time setting. Every sensor will get the actual t_{end} in every ACK message from the coordinator. If the probability that the person is in the zone is decreased, the t_{end} should be set back to smaller value. Increasing t_{end} can reduce the precision of the position tracking (see Fig. 5.) especially when fast zone changes happen. In this case the system cannot track the changes fast enough. The performance of this method will be presented in the next section.

6 Evaluation of the method using measured data in a test environment

To analyze the efficiency of the proposed method shown in section 4 a simulation was performed. The input of the simulation was real-life measured data series. A test apartment has been equipped with 6 identical motion sensors ($m = 6$) (see Fig. 6); these sensors divide the space into 6 zones ($n = 6$). The zones are not overlapping, except zone 5 and 6, but this was neglected when the states were defined. The observation vector may contain three symbols: o_i^1 means motion, o_i^2 means no motion, and o_i^3 means no measurement. The o_i^3 symbol is generated when there is no message from sensor i in an actual time step. The parameters of the sensor model are defined empirically based on the experimental results of the initial simulations. The state transition (\mathbf{A}) matrix is defined to reflect the relationship between the different locations. The state transition probability was effectively zero between states which do not represent not neighboring zones. The \mathbf{A} matrix may also defined by learning if enough data points are recorded, which in this case were not

available.

Only one observed person was in the apartment during the recording of data for the evaluation. The length of the measurement was 6 hours while the observed person was active. These measurements were recorded with small t_{end} , used as the possible best precision for measurements investigation purposes. The global states (spatial positions) based on these short t_{end} data were taken as true states; all the degradations were compared to these ones.

The simulator implements the HMM and processes the measurements. Based on the output of the HMM, longer t_{end} times were simulated ($t_{\text{end},\text{sim}} = [5..30]\text{s}$), and the power consumption and degradation in precision was calculated and compared to the original (short t_{end}) values.

The method described previously was applied: sensors in the zone with the highest probability were set to a longer t_{end} , the other ones were set to the original value. The error was calculated from the output of two HMMs for every k time-step: HMM_A ran on the original measurements; HMM_B ran on the simulated measurements. The time-step was 1 second. The error was calculated with the following formula:

$$D_k = \begin{cases} 0, & \text{if } A_k = B_k \\ 1, & \text{if } A_k \neq B_k \end{cases} \quad (9)$$

$$D_{\text{rel}} = \frac{\sum_{k=1}^K D_k}{K} \cdot 100 \quad (10)$$

where A_k and B_k represent the zone index with the maximal probability from the state vector of HMM_A and HMM_B, K is the time index of the end of the processes measurement series. D_{rel} is given as percentage value.

The energy consumption can be estimated from the number of transmissions. To visualize the saved energy a ratio was calculated:

$$\text{Eliminated messages}[\%] = \left(1 - \frac{\text{Simulation message count}}{\text{Original message count}}\right) \cdot 100 \quad (11)$$

Figure 7. shows the result of the simulation (e.g. if the longer t_{end} is 10 seconds, 54% of the original messages can be eliminated and the relative error is still below 3%).

This result can also be interpreted as an increase in the particular sensor's lifetime (battery life). The number of transmissions is almost directly proportional to the energy consumption (a small amount of energy is needed for the measurement and sleep modes), e.g. if half of the transmissions are performed, half the energy is consumed, therefore the battery life is doubled. This operation is especially beneficial in AAL applications, where battery-powered long-term operation is needed for position tracking to infer the current activity of the observed person. Small degradation of the tracking performance will not affect the activity detection. Notice that the method does not affect the start-of-the-motion messages if the person enters a different zone. And despite this small degradation in precision, the lifetime of the sensors can be extended by even 4-5 times.

Fig. 4. Measured data from two motion sensors in two different zones (every transition means a data transmission). Below the real position changes are marked with arrows. The other transmissions are generated by movements within the zone; these messages contain no valuable information for position tracking.

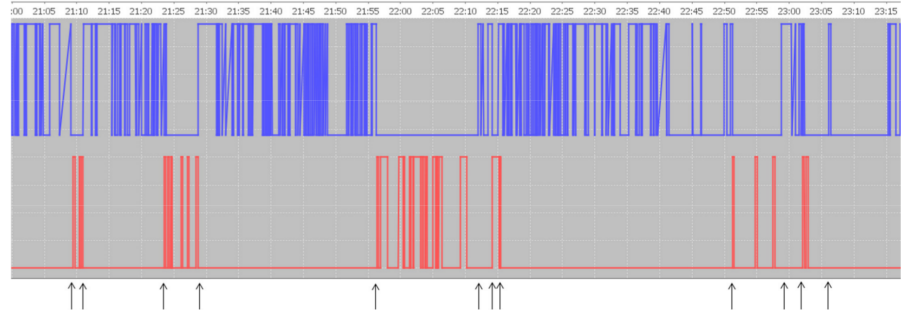


Fig. 5. The effect of increasing t_{end} : notice the eliminated messages and the delayed motion end messages

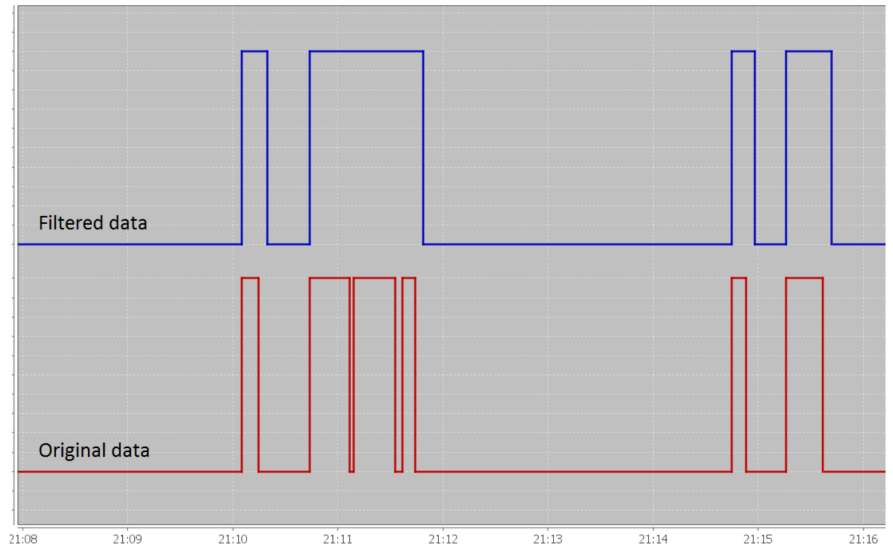
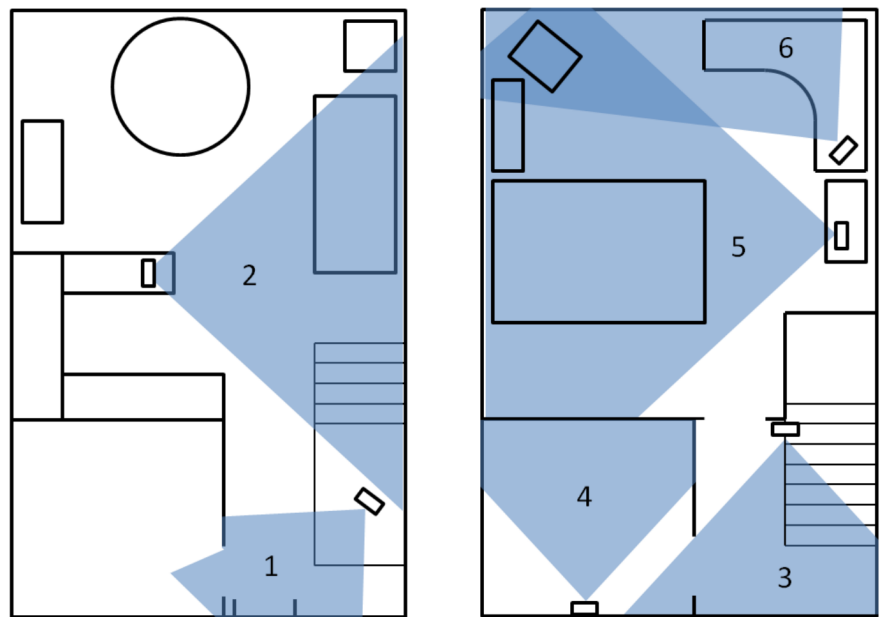


Fig. 6. The arrangement of the test apartment with 6 motion sensors

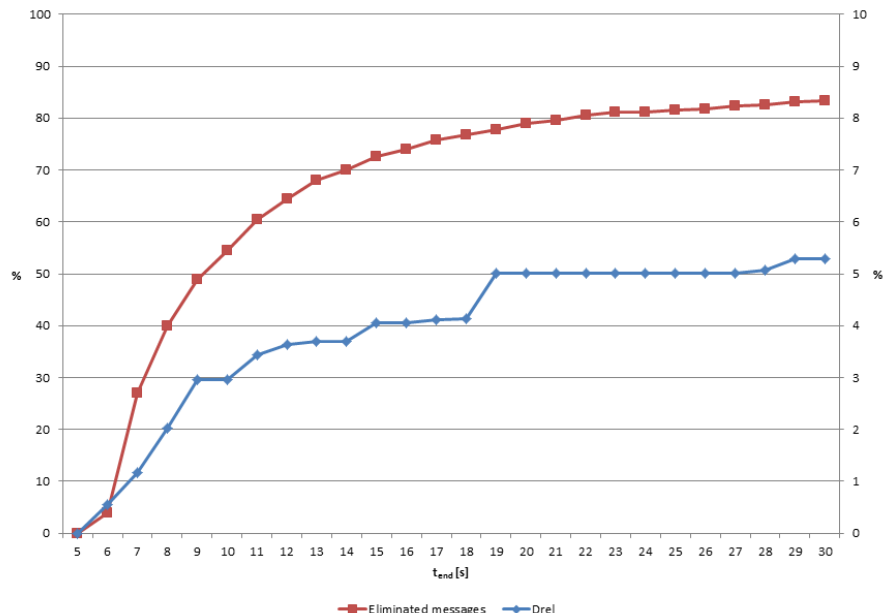


7 Conclusion

In this paper a new method is presented for filtering the transmissions of motion sensors used in AAL applications. In former research, modeling of the motion of a person with HMM was used to schedule the measurements of the different sensors. In the current case the sensors have very low power consumption, therefore they are measuring continuously and only the commu-

nication is affected. The HMM's output is used to filter the messages from the sensor to the central coordinator node. It is useful because most of the messages are caused by local movements which are not relevant to the position tracking. The results have shown that at least the half of the messages can be eliminated with minimal degradation of the precision of position tracking.

Fig. 7. The results of the simulation. Notice that the scale for Drel is on the right side and for the eliminated messages is on the left side.



Acknowledgement

The results discussed in this paper were supported by the grant TÁMOP - 4.2.2.B-10/1–2010-0009.

References

- 1 Kulkarni RV, Förster A, Venayagamoorthy GK. *Computational Intelligence in Wireless Sensor Networks: A Survey*, Communications Surveys & Tutorials, IEEE, **13**(1), (First Quarter 2011), 68–96.
- 2 Györke P, Pataki B. *Application of energy-harvesting in wireless sensor networks using predictive scheduling*, IEEE International Instrumentation and Measurement Technology Conference (I2MTC 2012) Proceedings, (May 2012), 582-587.
- 3 Györke P, Pataki B. *Energy-Aware Measurement Scheduling in WSNs Used in AAL Applications*, IEEE Transactions on Instrumentation and Measurement, (2013). Accepted.
- 4 Kansal A, Hsu J, Srivastava M, Raghunathan V. *Harvesting aware power management for sensor networks*, Proceedings of the 43rd annual Design Automation Conference, (July 24-28, 2006).
- 5 Alippi C, Anastasi G, Di Francesco M, Roveri M. *An Adaptive Sampling Algorithm for Effective Energy Management in Wireless Sensor Networks With Energy-Hungry Sensors*, IEEE Transactions on Instrumentation and Measurement, **59**(2), (Feb. 2010), 335-344.
- 6 Kurp T, Gao RX, Sah S. *An adaptive sampling scheme for improved energy utilization in wireless sensor networks*, IEEE Instrumentation and Measurement Technology Conference (I2MTC), (May 2010), 93-98.
- 7 Ghasemzadeh H, Guenterberg E, Jafari R. *Energy-Efficient Information-Driven Coverage for Physical Movement Monitoring in Body Sensor Networks*, IEEE Journal on Selected Areas in Communications, **27**(1), (January 2009), 58-69.
- 8 Zhang Y, Ji Q. *Efficient Sensor Selection for Active Information Fusion*, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, **40**(3), (June 2010), 719-728.
- 9 Krishnamurthy V. *Algorithms for optimal scheduling and management of hidden Markov model sensors*, IEEE Transactions on Signal Processing, **50**(6), (Jun 2002), 1382-1397.
- 10 Evans J, Krishnamurthy V. *Optimal sensor scheduling for Hidden Markov models*, Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, **4**, (1998), 2161-2164.
- 11 He Y, Chong KP. *Sensor scheduling for target tracking in sensor networks*, 43rd IEEE Conference on Decision and Control, **1**, (2004), 743-748.
- 12 Lin J, Xiao W, Lewis FL, Xie L. *Energy-Efficient Distributed Adaptive Multisensor Scheduling for Target Tracking in Wireless Sensor Networks*, IEEE Transactions on Instrumentation and Measurement, **58**(6), (2009), 1886-1896.
- 13 Liu X, Chong EK, Shroff NB. *Transmission scheduling for efficient wireless utilization*, Proceedings Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies, **2**, (2001).
- 14 Srivastava R, Koksal CE. *Energy optimal transmission scheduling in wireless sensor networks*, IEEE Transactions on Wireless Communications, **9**(5), (2010), 1550-1560.