A Method for Self-Organizing Communication in WSN Based Localization Systems: HashSlot

Marcel Baunach, Reiner Kolla, Clemens Mhlberger
University of Wuerzburg, Germany

A Method for Self-Organizing Communication in WSN Based Localization Systems: HashSlot

Marcel Baunach, Reiner Kolla, Clemens Mühlberger
Department of Computer Engineering, University of Würzburg, Am Hubland, 97074 Würzbrug, Germany
Email: {baunach, kolla, muehlberger}@informatik.uni-wuerzburg.de
Telephone: +49 (0)931-888-6715, Fax: +49 (0)931-888-6702

***Abstract*—Localization of objects within space is a common problem in WSN research. Besides the location estimation itself, wireless communication is a central aspect within such systems.**

We present the novel HashSlot method for assured and collision-free transmission of radio packets from multiple sources to a common destination within a constant and predictable time. Due to self-organization, our approach needs no prior active coordination between sensor nodes and offers various techniques like selectable quality of service levels to dynamically limit the number of returned information during runtime. Thereby, precision and speed of the localization process can be adjusted, fault tolerance can be achieved and energy consumption will be reduced. This paper describes theory and application of the HashSlot method within a real WSN based localization system.

***Index Terms*—localization, self organization, service deployment, wireless communication, dynamic TDMA**

I. INTRODUCTION

Many applications depend on precise knowledge about the spatial location of various (mobile) objects in space. For example, the tracking of mobile vehicles within an industrial environment is a common scenario. Where large scale systems like GPS or Galileo [1] are designed for coarse outdoor navigation, WSN based localization systems are optimal for indoor application as they allow easy deployment and achieve a much higher precision. Indeed several indoor localization systems optimized for different requirements are available, e.g. Active Bat [2], AHLoS [3], Cricket [4], [5], Dolphin [6] and SNOW BAT [7]. Some of them are partially presented by Tseng at al. in [8] and Bulusu in [9]. These systems require a pre-installed infrastructure of static sensor nodes (*anchors*) for localizing mobile sensor nodes (*clients*) mounted on the objects to be observed. A common approach is to measure distances or angles between a client and several anchors and to apply location estimation algorithms on the obtained information.

This paper addresses the problem of efficient communication between mobile and anchor nodes within such systems. The HashSlot method is optimized for application within localization systems as it adopts to environmental factors like the geometry of the observed room, the anchors' positions and the usage of prior localization results for further optimizations. Additionally, it allows anchors to detect very early if they are not demanded for localization. In this case, the distance measurement can be omitted entirely to save energy and to be available for other mobile nodes.

To support the considerations within this paper, we'll refer to the SNOW BAT system [7] based on the SNOW^5 sensor nodes [10] with TI's MSP430 MCU [11] as a fully implemented testbed was available for research.

II. BASIC CONCEPTS OF THE SNOW BAT SYSTEM

The SNOW BAT system is designed for reliable, fast and precise 2D/3D localization and tracking of mobile objects. It uses ultrasonic (US) ranging for distance measurement and an elaborated heuristic for location estimation with a precision of $4\ \mathrm{mm}$ per dimension.

The basic idea is to deploy static sensor nodes within the monitored environment and to mount mobile sensor nodes on the objects under surveillance. The clients carry US transmitters and send US signals to the anchors which carry US receivers. Since SNOW BAT is a completely wireless (loosely coupled) system, radio is used for all communication tasks, even for synchronization. The localization process as shown in Fig. 1 designates SNOW BAT as a decentralized system of four stages: (1.) combined initiation/synchronization via radio (CAV, Chirp Allocation Vector), (2.) distance computation via measurement of TDoA (Time Difference of Arrival) between radio and ultrasound chirp, (3.) return of measured distances via radio (DV, Distance Vector) from the anchors to the client and (4.) location estimation.

Localization always relies on measuring the distances between a client and some anchors. This is done simultaneously for all anchors in the client's US range. For most position estimation algorithms in dim-dimensional space, at least $m \geq 1 + dim$ distances respectively anchors are required. Overestimating this system commonly yields increased precision and fault tolerance with each additionally measured distance and is also supported by SNOW BAT. HashSlot addresses this special fact adequately by allowing an adjustment of the amount of returned information dynamically at runtime to tune speed and precision. Another feature of SNOW BAT is its ability to track several mobile objects simultaneously. Therefore, the localization process can be initiated individually by each mobile node just when required. This grants additional autonomy to the clients but raises difficulties in maintaining a high localization frequency. The applied wireless protocol in combination with HashSlot significantly enhances the scalability of the overall system regarding the number of simultaneously supported mobile nodes, localization frequency and power consumption.

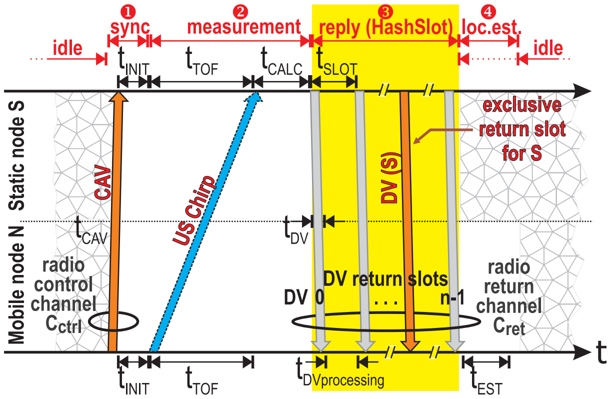

Fig. 1. The SNoW Bat localization process

A. Scalability and communication protocol

To initiate localization, a client N broadcasts a single hop radio message called *chirp allocation vector* (CAV). The CAV is required for wireless (time) synchronization of sender and receivers, which is indispensable within TDoA systems. Furthermore, it contains some parameters about N and its US signal. For now, relevant are the network address of N, the radio channel to be used for returning information, the desired QoS level and optionally an estimation for the current distance h of N to the anchor plane in which all anchors are mounted when using HashSlot (→ section III and Fig. 2a,c, 8).

Returning the obtained distance, the anchor's position and some other information required for location estimation back to the client N is the last step an anchor node S performs during a localization process. This is done by sending a one hop radio packet called *distance vector* (DV). Of course, the DV will only be transmitted by those anchors that could determine a distance to N. Just to get an idea about SNoW Bat: CAV and DV are equal in size (42 B) and require 1.3 ms to transmit (receive) at 159 mW (60 mW).

To track a mobile object precisely, a sufficient spatial and temporal resolution is indispensable. As SNoW Bat already achieves a high spatial accuracy we will address optimizations for the temporal resolution which heavily depends on the number of static and mobile nodes within the system. The basic difficulty in providing a high localization frequency is the wireless transmission of information. If radio or ultrasonic signals interfere, the contained information is lost or corrupted. However, Fast and assured transmission of synchronizing CAVs and US chirps with the first attempt is mandatory for reliable distance measurement. Furthermore, it is likely for anchors which measured the distance to the same mobile node that they will try to return their DVs at about the same time. This evokes the problem just mentioned and may defer the localization of other mobile nodes due to clear channel assessments or radio retransmissions.

Hence, the HashSlot method mainly addresses the optimized transmission of distance vectors from several anchors to one client (many to one).

III. Deployment of the SNoW Bat system

The deployment of a technical system into an existing environment should exert as little influence as possible to it. For SNoW Bat this implies a deliberate number and placement of static nodes. On the one hand, few anchors mean low costs, fast deployment, low energy consumption, minor maintenance effort, reduced environmental pollution due to radio transmissions, etc. On the other hand, a certain amount of anchors is required to guarantee an area-wide coverage with the localization service at sufficient precision and fault-tolerance.

In fact, the original SNoW Bat system [7] needs no special alignment of the anchor nodes as long as each one knows its exact position in world coordinates and it is assured, that a mobile node can always measure distances to a sufficient number of static nodes.

However, as we will see later, the HashSlot method requires the anchors to be roughly aligned along a grid pattern to unfold its potential. Thus, we initially present an approach to find an optimal grid with grid constant L (→ Fig. 2b) to guarantee a sufficient number of distance measurements. All static nodes will be aligned to this grid and mounted on the same level, called the *anchor plane*, within the observed space (e.g. ceiling or floor). The US receivers/senders should listen/transmit into the room ideally orthogonal to the anchor plane.

To find an appropriate L, we'll do some considerations about the room geometry first (→ Fig. 2a,b). The maximum supportable room height h_{sup} depends on the US range u and its beam angle φ which are constant with the applied hardware: $h_{sup} = u \cdot \cos(\varphi)$. The least possible distance h_{min} from a client to the anchor plane determines the maximum allowed value of the grid constant L. This is of special importance as soon as clients might not only move parallel to the anchor plane (2D) but also orthogonal (3D). The minimal coverage zone Z of the US signal has a radius $r_{min} = h_{min} \cdot \tan(\varphi)$. To guarantee that at least four grid points are within this (moving) zone, three nodes aligned like A, B and C in Fig. 2b must be located inside Z. In other words: the coverage zone Z with radius r_{min} must be at least as large as the circumcircle of the triangle ABC. Some simple trigonometric calculations reveal equation (1) for L:

$$a = \sqrt{2 \cdot L^2} = \sqrt{2} \cdot L$$

$$\left.\begin{aligned} \sin\left(\frac{\beta}{2}\right) &= \frac{\frac{a}{2}}{\sqrt{L^2+4 \cdot L^2}} = \frac{1}{2} \cdot \sqrt{\frac{2}{5}} \\ \sin\left(\frac{\beta}{2}\right) &= \sqrt{\frac{1}{2}(1-\cos(\beta))} \end{aligned}\right\} \Rightarrow \cos(\beta) = \frac{4}{5}$$

$$\sin(\beta) = \sqrt{1-\cos^2\beta} = \frac{3}{5} = 0.6$$

$$(\text{Z}) \quad r_{min} \geq \frac{a}{2 \cdot \sin(\beta)} = \frac{\sqrt{2} \cdot L}{1.2} \quad \text{(circumcircle)}$$

$$\Rightarrow L \leq \frac{1.2 \cdot r_{min}}{\sqrt{2}} = \frac{1.2 \cdot h_{min} \cdot \tan(\varphi)}{\sqrt{2}} \qquad (1)$$

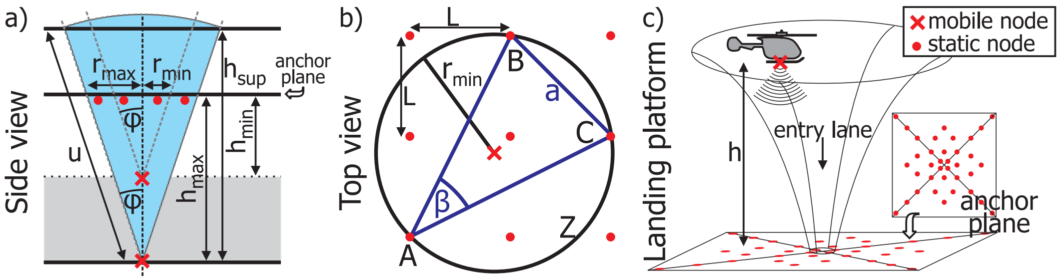

Fig. 2. SNoW Bat room geometry and deployment analysis: (a) side view, (b) top view, (c) landing platform

By arranging the anchors as described, it is possible to localize any mobile object within a distance between h_{min} and h_{max} from the anchor plane. Notice, that fault tolerance improves with increasing distance from the ceiling as more anchors will receive a mobile node's US signal to compute a distance. Another way to achieve this is placing the anchors even more densely just where necessary. In case of a landing platform as illustrated in Fig. 2c, the grid becomes finer towards the central landing point as a function of the helicopter's minimal height according to its entry lane.

Now, that L is computable from the application specific h_{min} and the hardware specific φ, it is also possible to calculate the whole system's maximum coverage area depending on the number of anchors. With a given network address width of a_{width} bits, $2^{a_{width}}$ anchors may be available and w.l.o.g. a quadratic area with an edge length $E = (\sqrt{2^{a_{width}}} - 1) \cdot L$ and a volume $V = E^2 \cdot (h_{max} - h_{min})$ can be covered.

As reference, the SNoW Bat hardware specifications are $\varphi = 30^\circ$ and $u = 8\text{ m}$, thus $h_{sup} = 6.93\text{ m}$. For tracking a vehicle on the plain floor of a hall with $h_{min} = h_{max} = 3\text{ m}$ a grid with $L \leq 147\text{ cm}$ would be used. Using $a_{width} = 8\ (16)$ bit a floor area of $486\ (140512)\text{ m}^2$ might be covered with the localization service.

IV. DISTANCE VECTOR TRANSMISSION.

The transmission of quite a number of distance vectors (DV) from anchors to the mobile node after distance measurement is elementary for a precise and fault tolerant location estimation. Yet, the required time must be kept short to save energy, achieve a high localization frequency and to cause little jamming on the radio channel. There are two main problems to deal with:

1) Anchors $S_1 \ldots S_m$ within the US coverage zone Z of a mobile node will receive its chirp(s) only slightly deferred in time. Thus m nodes will try to return their DVs approximately at the same time and have to coordinate with each other. This might not only consume a lot of energy and time, but
2) it might also delay or disturb the transmission of other CAVs, leading to a reduced localization frequency in a system with several mobile nodes.

We solve problem 2 by reserving a dedicated radio channel for the DVs of each mobile node. This way, SNoW Bat requires one (control) channel C_{ctrl} for CAV transmission and supports an arbitrary number $ch \geq 1$ of channels C_{ret} for returning DVs. These *return channels* will be assigned statically if the total number of mobile nodes $w \leq ch$. If $w > ch$, the return channel must be assigned dynamically for each localization. Yet, the underlying technique will not be discussed in this paper. This way, idle anchors will listen for CAVs on C_{ctrl} and switch to C_{ret} defined in the CAV for returning the DV. With the radio transceiver [12] on the SNoW^5 sensor board we have at least 32 fast and software switchable channels available and modified the original SNoW Bat to support up to 31 mobile nodes simultaneously with static channel assignment.

Solving problem 1 is much more challenging and can be handled by SNoW Bat in two different ways depending on the alignment of the anchors:

Originally, SNoW Bat used CSMA/CA, which is quite common in wireless communication and may be used independent of the anchor's alignment. Here, each anchor performs a time and energy consuming clear channel assessment (CCA) before sending a DV. In case the channel is occupied, it deferres the transmission attempt by applying a backoff method. Anyhow, collisions can't be ruled out completely and would corrupt at least two DVs each. The novel approach is HashSlot and will be addressed in the next section.

V. HASHSLOT.

HashSlot is a TDMA (time division multiple access) method that extends the idea of deploying anchors along a well defined grid (→ section III). It takes advantage of the locality principle which is found in various WSN applications: in SNoW Bat, only anchors within a common US coverage zone will measure and return distance information to the same client.

The basic idea is to dynamically compute exclusive transmission slots. In contrast to time slot protocols like TRAMA [13] that use variable schedule negotiation, HashSlot relies on the anchors' positions within the grid. Slots are computed by each anchor autonomously and without interaction or communication with other anchors. As the slot numbers ($\in \mathbb{N}_0$) specify the sequence in which the transmission will take place, it is important to pack these numbers very tightly (→ Fig. 1). Yet, CCA and ACKs can be avoided entirely, resulting in an extraordinary low time and energy consumption: anchors only require radio TX (no RX), clients only require radio RX (no TX) for DV transmission.

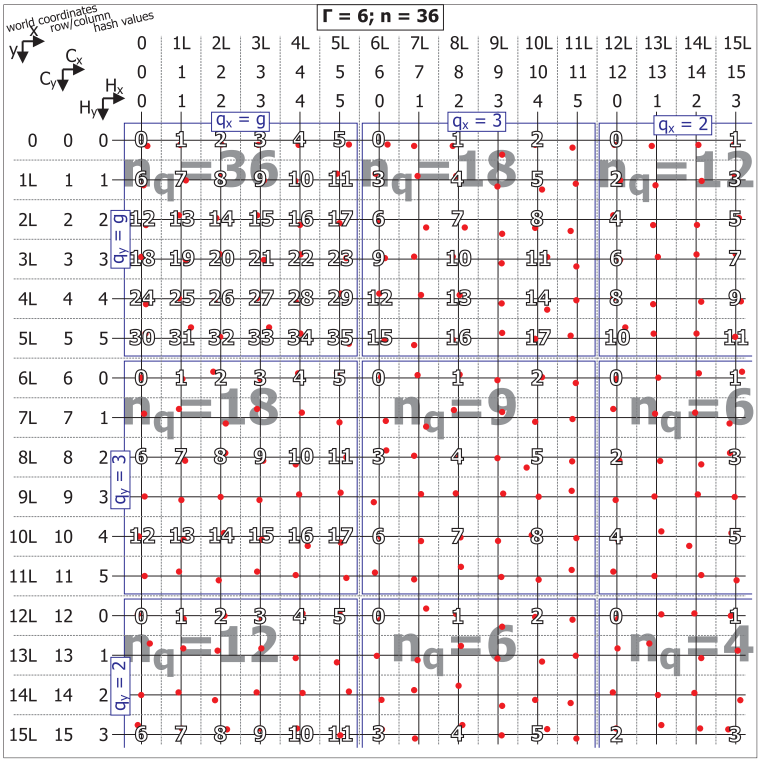

Fig. 3. Calculation of various values for return slot generation

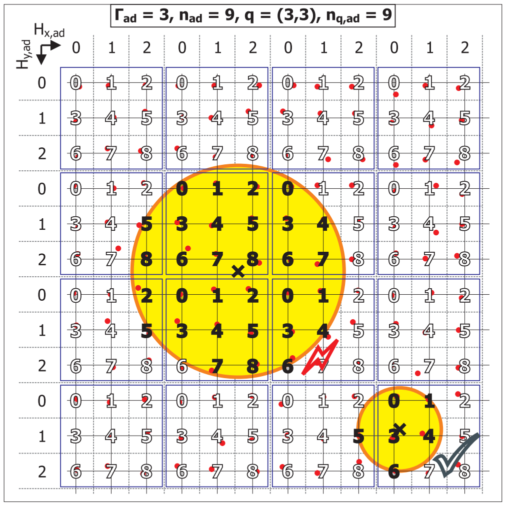

Fig. 4. Usage of the adaptive grid module Γ_{ad}

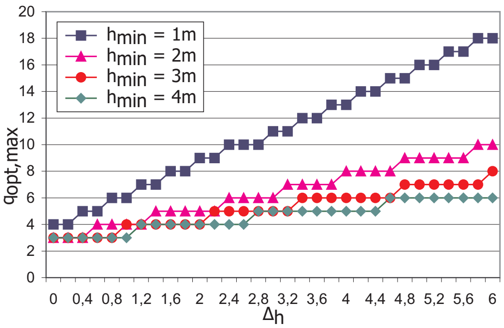

Fig. 5. $q_{opt,max}$ with respect to Δ_h and h_{min}

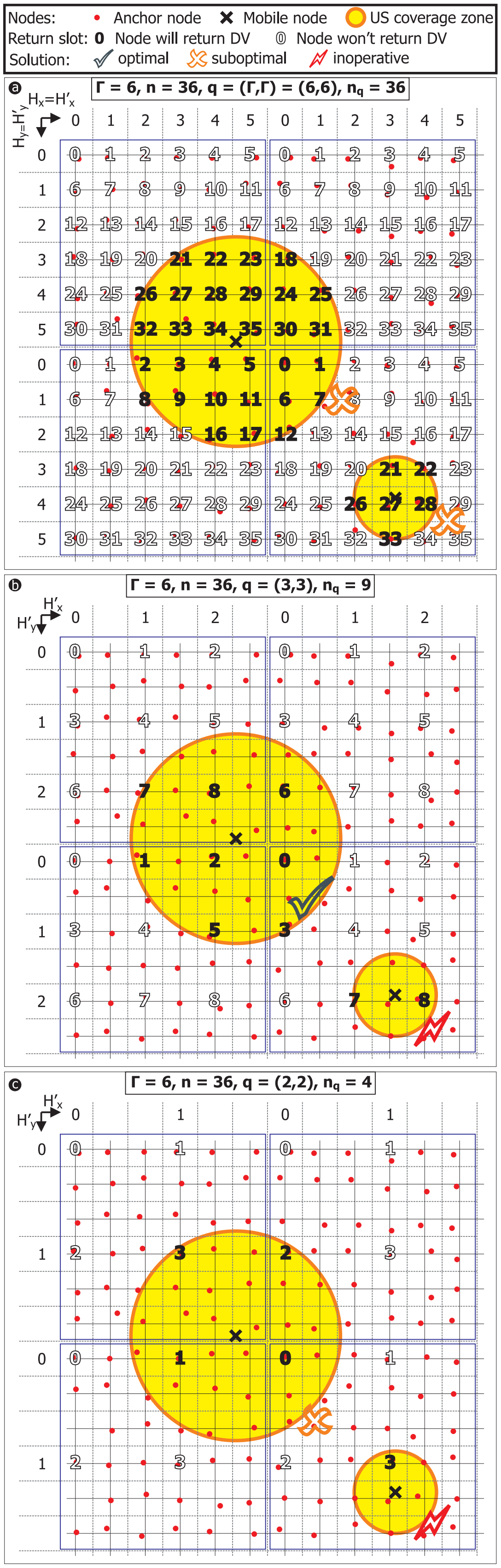

Fig. 6. Return slot assignment within the smallest / largest supported coverage zone at QoS levels a:(Γ, Γ), b:(3,3), c:(2,2)

We will show how to compute unique and tightly packed slot numbers within a US coverage zone Z and how QoS becomes an additional advantage with respect to environmental conditions. Further considerations will allow the calculation of network load, time and energy requirements. We will refer to Fig. 3 to 6 for visualization.

In (1) we calculated the grid constant L depending on h_{min} for the optimum alignment of the anchor nodes. This way, the circular US coverage zone Z with radius r_{min} always includes a minimum of four anchors. It is obvious, that a minimum square Q around Z always includes $4 \leq G \leq \left(\left\lfloor \frac{2 \cdot r_{max}}{L} \right\rfloor + 1\right)^2$ nodes aligned in $2 \leq g \leq \left(\left\lfloor \frac{2 \cdot r_{max}}{L} \right\rfloor + 1\right) = \Gamma$ rows and just as many columns (→ Fig. 6a). We will call $\Gamma = g_{max}$ the *standard grid module* and initially reserve n return slots for anchors/DVs per localization:

$$n = n_{max} = g_{max}^2 = \Gamma^2 = \left(\left\lfloor \frac{2 \cdot r_{max}}{L} \right\rfloor + 1\right)^2 \tag{2}$$

Thus, $\Gamma \overset{(1)}{\geq} 3$ and $\Gamma \cdot L > 2 \cdot r_{max}$ always holds and even the largest Z possible will always fit into a square of $n = \Gamma \times \Gamma$ anchors. See the example in Fig. 6a for $\Gamma = 6, n = 36$. The assignment of the n return slots to the anchors depends on their (geometric) grid position and additional factors like the QoS level which will be addressed later. An anchor S at absolute world coordinates $P_S(x|y|z)$ on the anchor plane (z is common for all anchors) resides within the grid in column C_x and row C_y (→ Fig. 3):

$$C_x = \left\lfloor \frac{x + \frac{L}{2}}{L} \right\rfloor \qquad C_y = \left\lfloor \frac{y + \frac{L}{2}}{L} \right\rfloor \tag{3}$$

Thus, each node may be placed almost $\pm\frac{L}{2}$ in x and y direction around its exact grid point to ease deployment. As we have seen above, anchors from at most Γ columns and rows will return their DVs to a common mobile node. This way, we can compute hash values from C_x and C_y by using the standard grid module:

$$H_x = C_x \bmod \Gamma \qquad H_y = C_y \bmod \Gamma \tag{4}$$

The last step is to assign an exclusive return slot B_S to the anchor S (→ Fig. 3 and 6a for examples on $H_{x/y}$ and B_S):

$$B_S = H_y \cdot \Gamma + H_x \qquad (\in [0 \dots n_{max} - 1]) \tag{5}$$

This guarantees, that any two different anchors S, S' with $B_S = B_{S'}$ may never be located within the same Z independent from the zone's overlay position within the anchor grid. Thus, wherever a mobile node localizes itself, no two or more anchors will use the same return slot for their DVs. According to Fig. 1, the maximum time required for transferring all DVs to the mobile node will be reserved and is

$$t_{n_{max}} = n_{max} \cdot t_{SLOT}. \tag{6}$$

Yet, there remain some problems to address: as long as the mobile nodes move only parallel to the anchor plane ($h_{max} = h_{min}$, $r_{max} = r_{min}$), L and thus n would be chosen perfectly by (2). But as soon as $h_{max} > h_{min}$ (3D operation) we have to distinguish two cases:

1) A mobile node far away from the anchor plane may indeed receive almost n_{max} DVs but in some cases this might be much more than required even for fault tolerant location estimation (→ Fig. 6a, large circle).
2) A mobile node close to the anchor plane still receives a sufficient number of DVs (L was chosen accordingly) but only $n_{min} = \left(\left\lfloor \frac{2 \cdot r_{min}}{L} \right\rfloor + 1\right)^2 \leq n_{max}$ return slots may finally be used (→ Fig. 6a, small circle).

Consequently this means a waste of time and radio RX power in both cases, in 1 even radio TX power is wasted. This is where QoS may be used to find a trade off between time/power consumption and the number of returned DVs for case 1. Case 2 can under certain circumstances be solved by the temporary calculation of the *adaptive grid module* Γ_{ad} at runtime. Remind, that optimizations must always deal with L, which was fixed during deployment and can't be changed anymore.

1) Quality of Service (QoS): This technique allows a client to dynamically select a subset of anchors within its US coverage zone for localization. Therefore, it broadcasts the desired QoS level q along with the CAV. q is a tuple (q_x, q_y) of natural numbers with $q_x, q_y \in [1 \dots \Gamma]$ and defines the number of desired DVs per dimension x, y of the anchor plane. The total number of requested DVs for q is n_q with

$$1 \leq n_q = q_x \cdot q_y \leq n = n_{max} \tag{7}$$

and equals the number of reserved return slots (→ Fig. 3, 6b,c). Thus, 1 (Γ) is the minimum (maximum) QoS level per dimension. Remind that, according to section III, selecting $q_x, q_y < 3$ might result in less than 4 DVs for position estimation depending on the overlay position of Z. The required time to transmit the DVs computes analog to (6) as

$$t_{n_q} = n_q \cdot t_{SLOT} \leq t_{n_{max}}.$$

Since QoS limits the number of used anchors and return slots to n_q, these slots can be rearranged more tightly over time by recalculating new hash values depending on q:

$$H'_x = \left\lfloor \frac{H_x \cdot q_x}{\Gamma} \right\rfloor \qquad H'_y = \left\lfloor \frac{H_y \cdot q_y}{\Gamma} \right\rfloor$$

Again, the final step is to reassign the n_q return slots to the anchors. Therefore, it is required to properly select the nodes that may return their DVs to avoid radio collisions. A node may only return a DV if

$$w := \forall_{v \in \{x,y\}} \; ((H_v \cdot q_v) \bmod \Gamma < q_v)$$

holds. This selects a uniform subgrid and the corresponding slot B'_S for an anchor S will be computed as follows:

$$B'_S = \begin{cases} H'_y \cdot q_x + H'_x & \text{if } w = true \\ none & \text{otherwise} \end{cases} \qquad (\in [0 \dots n_q - 1])$$

An example for various QoS levels is given in Fig. 3. Remains the question, which QoS level q_{opt} is optimal. We'll give a solution for $q_{opt,x} = q_{opt,y}$ if the distance h_l from the mobile object to the anchor plane (or at least a lower bound for it) is known (→ Fig. 7, 8). This might be true for objects that move only parallel to the anchor plane or can estimate h_l from prior localizations. With

$$\begin{aligned} r_{h,l} &= h_l \cdot \tan(\varphi) && (\geq r_{min}) \\ L_{opt} &\overset{(1)}{=} \frac{1.2 \cdot r_{h,l}}{\sqrt{2}} && (\geq L, \text{ fixed!}) \\ \Gamma_{opt} &\overset{(2)}{=} \left\lfloor \frac{2 \cdot r_{h,l}}{L_{opt}} \right\rfloor + 1 = 3 && (\leq \Gamma) \end{aligned}$$

we can easily calculate

$$q_{opt,x} = q_{opt,y} = \left\lceil \frac{L \cdot \Gamma}{L_{opt}} \right\rceil = \left\lceil \frac{r_{min}}{r_{h,l}} \cdot \Gamma \right\rceil \quad (\leq \Gamma). \tag{8}$$

In case $r_{h,l} = r_{min}$ this selects the maximum QoS level $q_{opt} = (\Gamma, \Gamma)$. In case $r_{h,l} = r_{max}$

$$\begin{aligned} q_{opt,x/y} &\overset{(8)}{=} \left\lceil \frac{r_{min}}{r_{max}} \cdot \Gamma \right\rceil \overset{(2)}{=} \left\lceil \frac{r_{min}}{r_{max}} \cdot \left(\left\lfloor \frac{2 \cdot r_{max}}{L} \right\rfloor + 1 \right) \right\rceil \\ &\geq \left\lceil \frac{r_{min}}{r_{max}} \cdot \frac{2 \cdot r_{max}}{L} \right\rceil \overset{(1)}{=} \left\lceil \frac{2 \cdot \sqrt{2}}{1.2} \right\rceil = 3. \end{aligned}$$

Therefore it is proven, that $q_{opt,x/y} \in [3 \dots \Gamma]$ and q_{opt} is a valid and useful QoS level as $9 \leq q_{opt,x} \cdot q_{opt,y} = n_q \leq n_{max}$ and at least 4 DVs are always available ($t_{q_{opt}} = n_q \cdot t_{SLOT}$). The large US coverage zone Z in Fig. 6b gives an example.

2) Adaptive grid module (AGM): Fig. 6b/c show, that low QoS levels are not practical for small US coverage zones as too few DVs might be returned compared to the reserved slots. We now show how to obtain sufficient DVs in a short time despite of a "small" Z and a "large" Γ (→ Fig. 4). Again, it is required that the mobile node knows its distance h_u from the anchor plane or at least an upper bound for it (→ Fig. 8).

In this case, the mobile node broadcasts h_u along with the CAV and allows the anchors to compute an adaptive grid module Γ_{ad} for this single localization. Obtaining Γ_{ad}, $B_{S,ad}$ and t_{ad} works analogous to before:

$$\begin{aligned} r_{h,u} &= h_u \cdot \tan(\varphi) && (\leq r_{max}) \\ n_{h,u} &\overset{(2)}{=} \Gamma_{ad}^2 = \left(\left\lfloor \frac{2 \cdot r_{h,u}}{L} \right\rfloor + 1 \right)^2 && (\leq n_{max}) \quad (9) \\ H_{x/y,ad} &\overset{(4)}{=} C_{x/y} \bmod \Gamma_{ad} && (\leq H_{x/y}) \\ B_{S,ad} &\overset{(5)}{=} H_{y,ad} \cdot \Gamma_{ad} + H_{x,ad} && (\leq B_S) \quad (10) \\ t_{ad} &\overset{(6)}{=} n_{h,u} \cdot t_{SLOT} && (\leq t_{n_{max}}) \end{aligned}$$

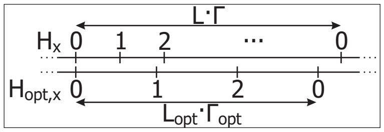

Fig. 7. Calculation of an optimal QoS level

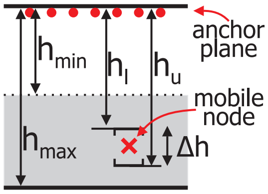

Fig. 8. Estimating $h \in [h_l, h_u]$

The small Z in Fig. 4 shows, that an adaptive grid module grants extremely tight packing of retun slots. In contrast, the large Z shows, that overestimating h_u (the mobile node is farer away from the anchor plane than expected) will result in a Γ_{ad} which is too small for the US coverage zone. In consequence, return slots will collide and valuable information is lost.

3) Combining QoS with AGM: This mixture reveals the most effective method for adapting to a mobile nodes's current distance h. If an upper and lower bound for $h = [h_l, h_u]$ with $h_{min} \leq h_l \leq h_u \leq h_{max}$ can be estimated (→ Fig. 8), it is possible to compute Γ_{ad} from h_u via (9) and q_{opt} from h_l and Γ_{ad} via (8). We will see, that the result becomes better the smaller $\Delta_h = (h_u - h_l)$ is. The idea is similar to modifying h_{min} and h_{max} for obtaining a new L and Γ. As these values are fixed since deployment of the anchors, the combination of Γ_{ad} and q_{opt} finds the optimal solution for the number of return slots n_h with respect to L and the uncertainity Δ_h of h:

$$n_h \overset{(7)}{=} q_{opt,x} \cdot q_{opt,y} \qquad t_h \overset{(6)}{=} n_h \cdot t_{SLOT}$$

It is interesting to notice, that computing q_{opt} from Γ_{ad} reaches its maximum value $q_{opt,max}$ at $h = [h_{min}, h_{min} + \Delta_h]$ for a given uncertainity Δ_h. Unlike Γ and n in (2), the value of $q_{opt,max}$ is independent from h_{max} and r_{max} respectively:

$$\begin{aligned} q_{opt,x/y} &\overset{(8)}{=} \left\lceil \frac{h_{min}}{h_l} \cdot \underbrace{\left(\left\lfloor \frac{2 \cdot \sqrt{2}}{1.2} \cdot \frac{h_u}{h_{min}} \right\rfloor + 1 \right)}_{\Gamma_{ad}} \right\rceil \\ \leq \quad q_{opt,max,x/y} &= \left\lfloor \frac{2 \cdot \sqrt{2}}{1.2} \cdot \frac{h_{min} + \Delta_h}{h_{min}} \right\rfloor + 1 \end{aligned} \tag{11}$$

This way, QoS plus AGM always delivers a computable maximum number of required return slots with $n_{h,max} \overset{(11)}{\propto} (h_{min} + \Delta_h)^2$ (→ Fig. 5). Without optimizations, $n \overset{(2)}{\propto} h_{max}^2$. Fig. 9 shows a concrete example for the number of return slots with respect to different optimizations and ranges of h.

4) Application of the HashSlot method: On power up of the localization system, each anchor S computes its individual values for C_x, C_y, H_x, H_y and B_S from its known absolute world position (→ (3) et seqq.). L and Γ may either be recalculated from room geometry or programmed during deployment.

If a mobile node wants to localize itself, it estimates its current distcance h or simply uses $h_l = h_{min}$ and $h_u = h_{max}$.

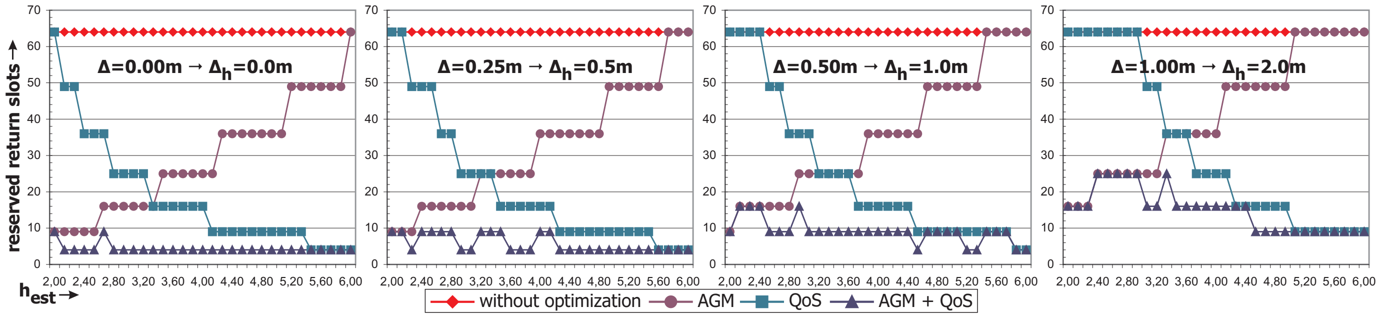

Fig. 9. Number of DV slots for different optimizations and ranges of $h = [h_{est} - \Delta, h_{est} + \Delta]$ (h_{min} = 2 m, h_{max} = 6 m $\Rightarrow \Gamma = 8, n = 64$)

If QoS or AGM is desired, the client respectively the anchors calculate the corresponding values for Γ_{ad} via (9) and q_{opt} via (8). Finally, the static nodes compute their individual collision free return slots as described above to transmit their DVs.

A closer look to the equations given within this section shows, that basic integer arithmetic operations are sufficient for all computations (L is given rounded off to mm). This allows a very efficient implementation even on small MCUs without floating point unit like the MSP430 on the SNoW^5 sensor node.

The HashSlot method is not limited to circular zones Z of replying nodes – indeed any shape for Z is supported. Yet, Γ and L must be calculated to guarantee that a square Q with edge length $\Gamma \cdot L$ always covers Z independent from Z's rotation in the anchor plane (→ Fig. 10). Only this allows the computation of collision free return slots. The optimal Q is the smallest square around the smallest circumcircle C of Z. Thus, the calculations mentioned above must be applied to r_{max} and r_{min} of C instead of Z. If the orientation of Z in the anchor plane is fixed, even the smallest rectangle R around Z is sufficient. However, the HashSlot method must be modified to support individual Γ_x and Γ_y for each dimension of the anchor plane.

Since the HashSlot method always reserves $q_x \cdot q_y$ return slots within the square Q it operates the more efficient the better Z covers Q as nodes within the area $Q \backslash Z$ won't return radio packets and their reserved time is wasted. Thus, for a non-rotating Z, any rectangular shape is optimal. For rotating shapes of Z, circles are optimal as the circumcircle of Z is Z itself and Q is the minimal square around Z. Obviously, the SNoW BAT localization system uses the HashSlot method in the most effective way possible as the (mobile) US transmitters may rotate parallel to the anchor plane and produce an almost circular US coverage zone Z.

5) Performance of HashSlot: Finally, we'll give a short performance overview of the HashSlot method based on empirical tests. Fig. 11a-c shows the average usage of return slots with respect to several QoS levels q and radii r of a circular Z. As Z moves freely over the anchor plane, it covers a different number of nodes depending on its position. Calculating the maximum / minimum number of lattice points within a circle is a hard problem in mathematics (circle problem [14], [15]). Thus, we obtained these numbers by (brute force) simulation. $G(r)_{min/max}$ shows the minimum / maximum number of nodes within Z. $J(r)_{min/max}$ describes the percentage of unused return slots. $P(G(r)_{min} > 4)$ is the probability that Z covers at least four anchors. It becomes obvious how r and q are correlated and determine precision and speed of the overall system. Fig. 11d shows the percentage of nodes within coverage areas – and thus used return slots – for various properties of Z and values of q.

Regarding the power consumption of the SNoW BAT hardware, the total amount of required energy for transmitting all DVs is: $G(r) \cdot 159\ \mathrm{mW} \cdot 1.3\ \mathrm{ms}$ for all anchors together (TX), and $n \cdot 60\ \mathrm{mW} \cdot 1.3\ \mathrm{ms}$ for the client (RX).

VI. CONCLUSION AND FUTURE WORK

In this paper we presented the HashSlot method for optimizing radio communication within localization systems due to self-organizing anchors. We described how the geometry of the observed space defines an optimal anchor grid and extended this approach by developing an efficient communication protocol based on this alignment. We showed, that the HashSlot method allows collision free transmission of one hop radio packets from several sources to a common destination within a predictable time without prior active coordination of nodes. Time and energy efficiency was achieved by dynamic calculation of transmission slots (TDMA) which were proven to never interfere with each other. Additional features like QoS and AGM refine the approach and allow an optimal adaption to given environmental conditions.

Future work aims on adding adaptive fault tolerance by automatic compensation of defective nodes using of QoS and AGM. Furthermore we are currently extending our real world testbed to comprise 70 anchors in a hall of $15 \times 10 \times 4$ m. This will allow us to compare theoretical calculations with a larger number of practical results regarding transmission

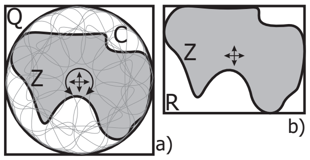

Fig. 10. Support for arbitrary shapes of Z: a) rotating b) non-rotating

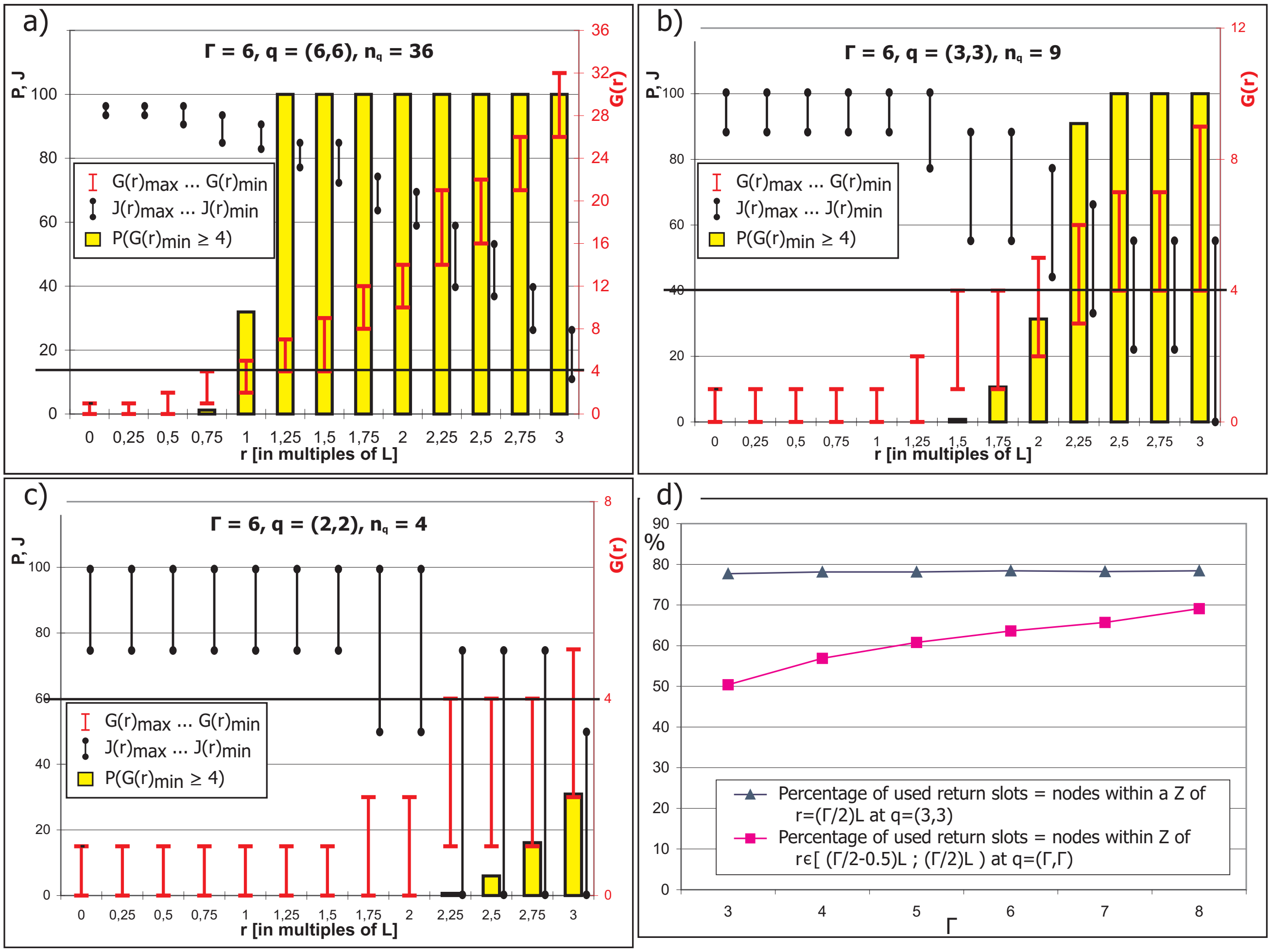

Fig. 11. Usage of return slots with respect to different QoS levels q and overlay position of Z over the anchor grid.

efficiency (compared to e.g. CSMA), multi-node localization, service coverage, energy consumption and fault tolerance.

REFERENCES

[1] R. Prasad and M. Ruggieri, Eds., *Applied Satellite Navigation Using GPS, GALILEO, and Augmentation Systems*. Artech House, 2005.

[2] A. Ward, A. Jones, and A. Hopper, "A New Location Technique for the Active Office," *IEEE Personal Comm.*, vol. 4, no. 5, pp. 42–47, Oct. 1997.

[3] A. Savvides, C.-C. Han, and M. B. Strivastava, "Dynamic fine-grained localization in ad-hoc networks of sensors," in *MobiCom '01: Proceedings of the 7th annual international conference on Mobile computing and networking*. New York, NY, USA: ACM Press, 2001, pp. 166–179.

[4] N. B. Priyantha, "The Cricket Indoor Location System," PhD Thesis, Massachusetts Institute of Technology, Jun. 2005.

[5] N. B. Priyantha, A. K. Miu, H. Balakrishnan, and S. Teller, "The Cricket Compass for Context-Aware Mobile Applications," in *MobiCom '01: Proceedings of the 7th annual international conference on Mobile computing and networking*. New York, NY, USA: ACM Press, Jul. 2001, pp. 1–14.

[6] Y. Fukuju, M. Minami, K. Hirasawa, S. Yokoyama, M. Mizumachi, H. Morikawa, and T. Aoyama, "DOLPHIN: A practical approach for implementing a fully distributed indoor ultrasonic positioning system," in *UbiComp 2004: Ubiquitous Computing (LNCS 3205)*. Springer, 2004, pp. 347–365.

[7] R. Kolla, M. Baunach, and C. Mühlberger, "SNoW Bat: A high precise WSN based location system," Institut für Informatik, Universität Würzburg, Tech. Rep. 424, May 2007.

[8] Y.-C. Tseng, C.-F. Huang, and S.-P. Kuo, "Positioning and location tracking in wireless sensor networks," in *Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems*, M. Ilyas and I. Mahgoub, Eds. CRC Press, 2005, ch. 21, pp. 1–13.

[9] N. Bulusu, "Localization," in *Wireless Sensor Networks - A Systems Perspective*, N. Bulusu and S. Jha, Eds. Artech House, 2005, ch. 4, pp. 45–57.

[10] R. Kolla, M. Baunach, and C. Mühlberger, "Snow^5: A versatile ultra low power modular node for wireless ad hoc sensor networking," in *5. GI/ITG KuVS Fachgespräch "Drahtlose Sensornetze"*, P. J. Marrón, Ed. Stuttgart: Institut für Parallele und Verteilte Systeme, Jul. 2006, pp. 55–59.

[11] *MSP430x1xx Family User's Guide*, Texas Instruments Inc., Dallas (USA), 2006.

[12] *CC1100 Data Sheet*, Chipcon AS, Oslo (Norway), 2005.

[13] V. Rajendran, K. Obraczka, and J. Garcia-Luna-Aceves, "Energy-efficient, collision-free medium access control for wireless sensor networks," *Wireless Networks*, vol. 12, no. 1, pp. 63–78, Feb. 2006, tRAMA. [Online]. Available: http://dx.doi.org/10.1007/s11276-006-6151-z

[14] H. Keller, "Numerical studies of the gauss lattice problem." [Online]. Available: citeseer.ist.psu.edu/keller97numerical.html

[15] M. N. Huxley, "Corrigenda: Exponential Sums and Lattice Points II," *Proc. London Math. Soc.*, vol. s3-68, no. 2, pp. 264–, 1994. [Online]. Available: http://plms.oxfordjournals.org