Locating Zigbee Devices in a Cluster-Tree Wireless Sensor Network: an ESD-based Integrated Solution

Stefano Tennina
WEST Aquila srl,
University of L'Aquila, Italy
tennina@westaquila.com

Marco Di Renzo
Laboratory of Signals and Systems (L2S)
Univ Paris-Sud (Paris), France
marco.direnzo@lss.supelec.fr

***Abstract*—Recent advances in the technology of wireless electronic devices have made possible to build ad-hoc Wireless Sensor Networks (WSNs) using inexpensive nodes consisting of low power processors, a modest amount of memory and simple wireless transceivers. Over the last years, many novel applications have been envisaged for distributed WSNs in the area of monitoring, communication and control. One of the key enabling and indispensable services in WSNs is localization (i.e., positioning), given that the availability of nodes' location may represent the fundamental support for various protocols (e.g., routing) and applications (e.g., habitat monitoring). Furthermore, WSNs are now being increasingly used for real-time applications having stringent Quality-of-Service (QoS) requirements, such as timeliness and reliability. Towards this end, Zigbee/IEEE 802.15.4 and the Cluster-Tree model are considered among the most promising candidates. Building from (i) our proposed Enhanced Steepest Descent (ESD) algorithm to solve positioning of nodes in a fully distributed fashion, (ii) the mechanism to evaluate at run-time the site-specific parameters for the correct operation of the ESD (i.e., RSSI-based ranging) and (iii) the recent availability of Zigbee/IEEE 802.15.4 implementations over TinyOS, the main output of this paper is to outline how a positioning service can be fully integrated into a communication protocol stack.**

***Keywords*-positioning service; communication protocol; ZigBee/IEEE 802.15.4; system integration.**

I. Introduction

Wireless Sensor Networks (WSNs) have been emerging as underlying infrastructures for new classes of large-scale and dense networked embedded systems. While there has been a plethora of scientific publications on WSNs, the vast majority focuses on protocol design (e.g., medium access control, routing, data aggregation) while only a scarce number of papers report real(istic) applications [1]. This might be due to the following facts: (i) WSN technology is extremely expensive for large-scale systems and (ii) is still very limited/unreliable, particularly in what concerns communications; (iii) difficulty on finding "killer" applications with a good cost/benefit trade-off; (iv) unavailability of standard, application-adequate, mature and commercially available technology; (v) lack of complete and ready-to-use system architectures, able to fulfill both functional and non-functional requirements. Despite relevant work on WSN architectures proposed so far (e.g., [2], [3]), none of them fulfills all requirements for large-scale real-time monitoring.

Moreover, WSNs are required to possess self-organizing capabilities, so that little or no human intervention for network deployment and setup is required. A fundamental component of self-organization is the ability of sensor nodes to "sense" their location in space, i.e., determining where a given node is physically located in a network [4].

In this work, we try to overcome the above limitations, showing how our previously presented fully distributed positioning service for WSNs [5] can be integrated into a full network architecture, built upon the Zigbee Cluster-Tree model [6] and IEEE 802.15.4 standard [7].

The remainder of this paper is as follows. Section II outlines the Zigbee Cluster Tree network architecture. Section III summarizes our proposed positioning service and Section IV presents how it can be integrated into the architecture. Finally, Section V provides concluding remarks.

II. Network Architecture

To achieve efficiency, scalability and QoS in WSN-based systems, a network architecture should have the following common features: (i) being multi-tier; (ii) using a core IP-based network for interconnecting heterogeneous elements and (iii) the IEEE 802.15.4 protocol for short range communications among sensor nodes. While the IP-based core network and the IEEE 802.15.4 standard are natural choices, thanks to their maturity, the use of a multi-tiered architecture, although it offers the highest level of flexibility, raises a number of challenges: (i) how many tiers and therefore how many communication technologies must be chosen, and (ii) what kind of nodes are the most appropriate for each tier (in terms of hardware features and power supply type)?

By focusing on the WSN portion, the devised architecture can be detailed as in Fig. 1. Tier–0 consists of simple wireless sensor nodes (SN), performing sensing tasks and delivering data to the devices at the upper tier in the hierarchy using the IEEE 802.15.4 protocol. SNs are cheap enough to be deployed in large quantities, therefore they usually have very limited computational, memory and energy capabilities. Multiple SNs are grouped to form a WSN Cluster at Tier–1

 ISBN: 978-1-61208-171-7

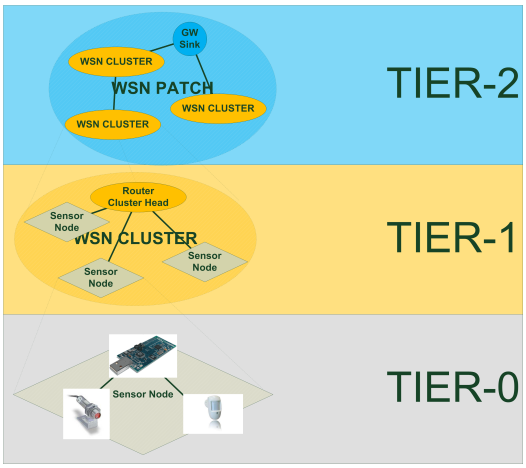

Figure 1. WSN multi–tiered architecture.

in a star topology, where a Cluster Head (CH, or *router*) is responsible for local management (e.g., synchronizing the nodes in the cluster, informing nodes about current duty-cycle, GTS slots, etc.), upstream/downstream routing and some data aggregation. CHs may be slightly more powerful than ordinary sensor nodes, in terms of computational capabilities and energy reserves. Multiple CHs are grouped to form a WSN Patch at Tier–2, where a gateway (GW) is present. GWs have the highest computational capabilities in the WSN and play the role of sinks/roots for their WSN Patch. GWs are equipped with a secondary transceiver, which enables their access to the IP core network.

WSN Patches adopt a *Cluster-Tree* model, with the GW as root and the SNs as leaves, and the synchronous beacon-enabled version of the IEEE 802.15.4 standard, where GW and CHs are the beacon emitter nodes. This has the advantages of (i) easily support time synchronization, (ii) improve the coordination to save energy (reduce retransmissions, put the nodes in sleep and wake them up in a synchronous fashion) and (iii) guarantee a given level of QoS, provided that a mechanism such as the Time Division Cluster Scheduling (TDCS) algorithm [8] is used to preserve the coordination and avoid intra-cluster collisions. TDCS involves the definition of the *StartTime* value (IEEE 802.15.4), such that the active portion of a cluster is scheduled during the inactive period of all the others, that share the same *collision domain*.

Once clarified that intra-clusters collisions within a WSN Patch are avoided using a time-division approach, it is worthwhile to state that a frequency-division approach is exploited to minimize the collision probability among nodes belonging to different WSN Patches. Similarly to [2], neighboring WSN Patches communicate over distinct radio channels, and channel re-use is allowed for any two patches distant enough from each other.

Overall, these two mechanisms are the key factors to improve the scalability of the network architecture.

III. Positioning Service

In this section, the ESD algorithm is briefly introduced as an enhancement of the well-known Steepest Descent (SD) method. Then, we recall the method presented in [5], used to enhance the accuracy of RSSI-based distance estimations.

A. Gradient-based Algorithms

Both SD and ESD are gradient descent methods [9]. This means that the position of a node is computed through the minimization of an appropriately defined error cost function.

The following notation will be used here: (i) bold symbols denote vectors and matrices, (ii) $(\cdot)^T$ denotes transpose operation, (iii) $\nabla(\cdot)$ is the gradient operator, (iv) $\|\cdot\|$ is the Euclidean distance and $|\cdot|$ the absolute value, (v) $\angle(\cdot,\cdot)$ is the phase angle between two vectors, (vi) $(\cdot)^{-1}$ denotes matrix inversion, (vii) $\hat{\mathbf{u}}_j = [\hat{u}_{j,x}, \hat{u}_{j,y}, \hat{u}_{j,z}]^T$ denotes the estimated position of the mobile node $\{u_j\}_{j=1}^{N_U}$, (viii) $\mathbf{u}_j = [u_{j,x}, u_{j,y}, u_{j,z}]^T$ is the trial solution of the positioning algorithm, (ix) $\bar{\mathbf{u}}_i = [x_i, y_i, z_i]^T$ is the position of the reference node $\{a_i\}_{i=1}^{N_A}$, and (x) $\hat{d}_{j,i}$ denotes the estimated (via ranging measurements) distance between reference node $\{a_i\}_{i=1}^{N_A}$ and blind node $\{u_j\}_{j=1}^{N_U}$.

The position of a node u_j is obtained by minimizing the error cost function $F(\cdot)$ defined as follows:

$$F(\mathbf{u}_j) = \sum_{i=1}^{N_A} \left(\hat{d}_{j,i} - \|\mathbf{u}_j - \bar{\mathbf{u}}_i\|\right)^2 \quad (1)$$

such that $\hat{\mathbf{u}}_j = \arg\min_{\mathbf{u}_j} \{F(\mathbf{u}_j)\}$. The minimization of such a function can be done using a variety of numerical optimization techniques, each one having its own advantages and disadvantages in terms of accuracy, robustness, convergence speed, complexity, and storage requirements [9].

1) Classical Steepest Descent: The classical Steepest Descent is an iterative line search method that allows to find the (local) minimum of the cost function in Equation (1) at step $k+1$ as follows [9, pp. 22, sec. 2.2]:

$$\mathbf{u}_j(k+1) = \mathbf{u}_j(k) + \alpha_k \mathbf{p}(k) \quad (2)$$

where α_k is a step length factor, which can be chosen as described in [9, pp. 36, ch. 3] and $\mathbf{p}(k) = -\nabla F(\mathbf{u}_1(k))$ is the search direction of the algorithm.

When the optimization problem is non-linear, small values of α_k are preferred to reduce the oscillatory effect when the algorithm approaches the solution.

2) Enhanced Steepest Descent: The SD method usually provides a good accuracy in estimating the final solution. However, it may require a large number of iterations, which may result in an unacceptably slow convergence speed. Then, the ESD has been proposed in order to improve such speed. The basic idea is to continuously adjust the step length value α_k as a function of the current and previous search directions $\mathbf{p}(k)$ and $\mathbf{p}(k-1)$, respectively:

 ISBN: 978-1-61208-171-7

$$\begin{cases} \alpha_k = \alpha_{k-1} + \gamma & \text{if} \quad \theta_k < \theta_{\min} \\ \alpha_k = \alpha_{k-1}/\delta & \text{if} \quad \theta_k > \theta_{\max} \\ \alpha_k = \alpha_{k-1} & \text{otherwise} \end{cases} \tag{3}$$

where $\theta_k = \angle(\mathbf{p}(k), \mathbf{p}(k-1))$, $0 < \gamma < 1$ is a linear increment factor, $\delta > 1$ is a multiplicative decrement factor, and θ_{min} and θ_{max} are two angular threshold values, that control the step length update.

By using the four degrees of freedom γ, δ, θ_{min} and θ_{max}, both the convergence rate and the oscillatory phenomenon when approaching the final solution can be simultaneously controlled, in a simple way and without appreciably increasing the complexity of the original SD algorithm.

B. Ranging Model

The ESD goal is the minimization of the error cost as defined in Equation (1). This assumes there is a way to estimate the distances $\hat{d}_{j,i}$ between pairs of nodes u_j and a_i, $i = 1, \ldots, N_A, j = 1, \ldots, N_U$.

Usually, for low cost platforms the Received Signal Strength (RSS)-based ranging method is preferred, since it doesn't require any extra hardware. However, this technique assumes a model to convert a RSS measurement into a distance, as e.g.:

$$d = 10^{[\frac{RSS-A}{10n}]} \tag{4}$$

where d denotes the transmitter-to-receiver distance, n is the propagation path-loss exponent, A is the RSS reference value, measured by a receiver located at a distance $d_0 = 1$ m from the transmitter, and RSS is the actual measured value.

In order to use the model, the values of the parameters A and n must be chosen. However, they are strongly environment-dependent, as clearly evidenced in Fig. 2, where A and n are shown as continuously updated during a conference event [5]. The big fluctuations suggest that using any fixed and outdated estimate certainly yields less accurate distances and, thus, final positions.

Hence, a new RSS-based *anchor*[1]-aided ranging method has been proposed in [5]. It foresees that every anchor node deployed in the area performs the following tasks: (i) transmits a packet containing its own position data; (ii) receives similar packets from other anchors in its radio range; (iii) extracts the position data as well as the RSS from the received packets, (iv) computes the Euclidean distance[2]; (v) after having collected enough (RSS, distance) pairs, estimates locally A and n via a linear least-square fitting using Equation (5)[3], and (vi) broadcasts these estimated parameters to the blind node. As far as the blind node is concerned, it receives the (A, n) pairs from each anchor, computes an average and uses them into Equation (4), to estimate the distances. Finally, it runs the ESD algorithm to compute its own position.

[1]An anchor node is a node, which knows, by definition, *a-priori* its position, or is able to estimate it, with high accuracy.

[2]Remember that each anchor knows its own position, hence, this computation gives a distance which is not affected by measurement errors.

[3]$y = RSS$, $x = 10 \cdot \log(d)$ and m = number of available measurements.

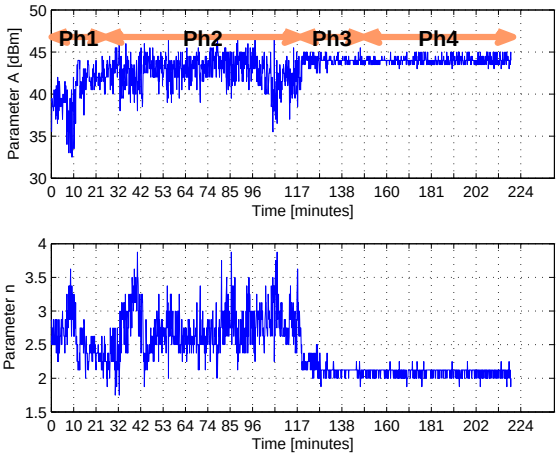

Figure 2. Estimated propagation parameters in a dynamic environment during a half-day conference [5].

IV. Integration

In order to optimize the connectivity for Cluster-Tree-based network models, it is often assumed to control the deplyment of the CHs and the GWs. As a consequence, it is straightforward to assume the local coordinators (i.e., CHs and GWs) as anchors and SNs as blind nodes.

In the light of above, we are implementing on the CHs and the GWs the described *anchor-aided* ranging mechanism. Then, the network formation procedure is as follows. At network setup, each GW starts by emitting its beacons using a predefined IEEE 802.15.4 channel, and all other nodes are scanning the medium, searching for such beacons. As soon as some CHs receive GW's beacons, they start the association process, in accordance with the IEEE 802.15.4 protocol and acting as normal nodes. Once associated with the parent, they start a negotiation procedure [8] to get an appropriate *StartTime* value, defining a window where they can transmit their own beacons, without interfering with other CHs. Hence, this mechanism iteratively enables all other nodes (SNs and other CHs) to join the network, upon a successful association phase.

On top of the TinyOS official 802.15.4 MAC [10] a Cluster-Tree model has been already implemented [11] as an extension of [12]. In this approach, the beacon payloads sent by every CH and GW are used to carry the positioning data (`setBeaconPayload`), such as the node's ID and its coordinates, as well as the two locally computed parameters A and n. As a matter of fact, since beacons are needed for networking and communication purposes, using their

$$\begin{bmatrix} A \\ n \end{bmatrix} = \frac{1}{m\sum_{i=1}^{m} x_i^2 - \left(\sum_{i=1}^{m} x_i\right)^2} \begin{bmatrix} \sum_{i=1}^{m} y_i \sum_{i=1}^{m} x_i^2 - \sum_{i=1}^{m} x_i \sum_{i=1}^{m} x_i y_i \\ m\sum_{i=1}^{m} x_i y_i - \sum_{i=1}^{m} x_i \sum_{i=1}^{m} y_i \end{bmatrix} \quad (5)$$

payload as a conveyor of data greatly helps lowering the energy costs: the overhead, that would be generated if the same data were sent using specific IEEE 802.15.4 Data frames, is simply avoided.

Finally, during the channel scan phase (`MLME-Scan`), every SN is able to extract (i) from the *beacon header* (`parsePANDescriptor`) the information needed to accomplish the IEEE 802.15.4 association with a parent, and (ii) from the *beacon payload* (`getBeaconPayload`) the positioning data needed to run the ESD algorithm, which has already demonstrated good performance in terms of accuracy, robustness, convergence speed, complexity, and storage requirements [13].

V. Conclusion

Wireless Sensor Networks are now being increasingly used for real-time embedded applications having stringent Quality-of-Service requirements, in terms of timeliness and reliability. Towards this end, Zigbee/IEEE 802.15.4 and the Cluster-Tree WSN model are considered among the most promising candidates. Moreover, one of the key enabling and indispensable services in WSNs is localization, since the availability of nodes' location may represent the fundamental support for various protocols (e.g., routing) and applications (e.g., habitat monitoring).

In this paper, building from (i) our proposed Enhanced Steepest Descent algorithm to solve positioning of nodes in a fully distributed fashion, (ii) the mechanism to evaluate at run-time the site-specific parameters for the correct operation of the ESD and (iii) the recent availability of Zigbee/IEEE 802.15.4 implementation over TinyOS, we outlined how a fully distributed positioning service can be implemented into a communication protocol stack, based on the Cluster-Tree WSN model. In particular, we stressed the fact that the peculiarities of the Cluster-Tree model (i.e., the presence of the beacons and of their scheduling to avoid intra-clusters collisions) can be exploited to implement an efficient localization system, with a very limited protocol overhead.

References

[1] B. Raman and K. Chebrolu, "Censor Networks: a Critique of "Sensor Networks" From a Systems Perspective," *SIGCOMM Comput. Commun. Rev.*, vol. 38, pp. 75–78, July 2008. [Online]. Available:

[2] C.-J. M. Liang, J. Liu, L. Luo, A. Terzis, and F. Zhao, "Racnet: a High-Fidelity Data Center Sensing Network," in *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, ser. SenSys '09. New York, NY, USA: ACM, 2009, pp. 15–28.

[3] Wirelessly Accessible Sensor Populations (wasp) project. FP6-IST-2005-2.5.3 Embedded Systems, Contract Number IST-034963. Duration: Sep 2006 to Oct 2010. [Online]. Available: http://www.wasp-project.org/

[4] C. Wang and L. Xiao, "Sensor Localization Under Limited Measurement Capabilities," *Network, IEEE*, vol. 21, no. 3, pp. 16 –23, may-june 2007.

[5] S. Tennina, M. Di Renzo, F. Graziosi, and F. Santucci, "Locating Zigbee Nodes Using the TI's CC2431 Location Engine: a Testbed Platform and New Solutions for Positioning Estimation of WSNs in Dynamic Indoor Environments," in *Proceedings of the first ACM international workshop on Mobile entity localization and tracking in GPS-less environments*, ser. MELT '08. New York, NY, USA: ACM, 2008, pp. 37–42.

[6] *Zigbee Specification*, ZigBee Standards Organization Std. 053 474, Rev. 17, January 2008.

[7] *IEEE Standard for Information Technology Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (LR-WPANs)*, LAN/MAN Standards Committee of the IEEE Computer Society Std., September 2006.

[8] P. Jurcik, R. Severino, A. Koubaa, M. Alves, and E. Tovar, "Dimensioning and Worst-Case Analysis of Cluster-Tree Sensor Networks," *ACM Transactions on Sensor Networks*, vol. 7, no. 2, August 2010.

[9] J. Nocedal and S. Wright, *Numerical Optimization*, 2nd ed., Springer, Ed. Springer, 2006.

[10] J.-H. Hauer, R. Daidone, R. Severino, J. Busch, M. Tiloca, and S. Tennina. (2011, February) An Open-Source IEEE 802.15.4 MAC Implementation for TinyOS 2.1. Poster Session at 8th European Conference on Wireless Sensor Networks.

[11] S. Tennina, M. Bouroche, P. Braga, M. Alves, R. Gomes, M. Santos, F. Mirza, A. Garg, V. Cahill, G. Carrozza, and V. Ciriello, "EMMON: a System Architecture for Large-Scale, Dense and Real-Time WSNs," in *Poster Session of the 8th European Conference on Wireless Sensor Networks (EWSN 2011)*, Bonn, Germany, February 2011, pp. 59–60, (invited poster).

[12] A. Koubaa. (2010, September) TinyOS Zigbee Working Group. [Online]. Available: http://www.hurray.isep.ipp.pt/activities/ZigBee_WG

[13] S. Tennina, M. D. Renzo, F. Graziosi, and F. Santucci, *Distributed Localization Algorithms for Wireless Sensor Networks: From Design Methodology to Experimental Validation*. InTech, June 2011, ch. Wireless Sensor Networks, iSBN: 978-953-307-325-5

 ISBN: 978-1-61208-171-7