2nd Joint ERCIM eMobility and MobiSense Workshop

Desislava Dimitrova, Marc Brogle,
Torsten Braun, Geert Heijenk (Eds.)

Palace of Grand Duke Vladimir
St Petersburg, Russia, June 4, 2013

Published: June 2013
University of Bern, Bern, Switzerland

ISBN 978-3-9522719-4-0

Preface

The second joint edition of the ERCIM eMobility workshop and the MobiSense workshop was held in St Petersburg (Russia) on June 4, 2013. The joint workshop was hosted in co-location with the 11th International Conference on Wired/ Wireless Internet Communications.

ERCIM, the European Research Consortium for Informatics and Mathematics, aims to foster collaborative work within the European research community and to increase co-operation with European industry. The ERCIM eMobility working group dedicates its research to mobile applications and services that require technical solutions on various levels. In the ERCIM eMobility workshop current progress and future developments in the area of eMobility are discussed and the existing gap between theory and application closed.

The MobiSense workshop (Opportunistic Sensing and Processing in Mobile Wireless Sensor and Cellular Networks) is dedicated to the collaboration and interoperability among wireless sensor networks and other wireless networks, especially cellular ones. These border research topics are of interests due to their potential to enhance the performance of the currently deployed wireless technologies.

This year's edition of the workshops welcomed scientific research in the areas of indoor localisation and location management, intelligent transportation systems, wireless sensor networks and machine-to-machine communications. All papers were either carefully selected in a peer review process by the joint workshop technical program committee or were invited on the ground of excellent experience of the speakers in the corresponding areas. In addition, an invited talk on Quality of Experience and its impact on current networks was given by prof. Peter Reichl from Telecom Bretagne (France).

We hope that all workshop delegates enjoyed the scientific program and discussion opportunities with colleagues. At this point, we wish to thank all authors of submitted and invited papers and the members of the program committee for their contribution to the success of the event. The proceedings include work in progress papers as well as more in-depth full papers on the presented topics. We hope that the workshop will continue to interest previous participants but will also attract new ones in the future as an event for the exchange of ideas and experiences.

Best wishes,
Mark Brogle and Desislava Dimitrova
TPC chairs

Torsten Braun and Geert Heijenk
General chairs

General chairs

Torsten Braun, University of Bern, Switzerland
Geert Heijenk, University of Twente, The Netherlands

TPC chairs

Marc Brogle, Hewlett-Packard, Switzerland
Desislava Dimitrova, University of Bern, Switzerland

Technical program committee

Mari Carmen Aguayo-Torres, Universidad Malaga, Spain
Francisco Barcelo-Arroyo, UPC, Spain
Boris Bellalta, UPF, Spain
Robert Bestak, TU Prague, Czeck Republic
Fernando Boavida, University of Coimbra, Portugal
Raffaele Bruno, Institute of Informatics and Telematics, Italy
Cristina Cano, Universidad Pompeu Fabra, Spain
Do van Thanh, NTNU, Norway
Rossitza Goleva, Technical University Sofia, Bulgaria
Edmundo Monteiro, University of Coimbra, Portugal
Gregory O'Hare, University College Dublin, Ireland
Vasilios Siris, ICS-FORTH / AUEB, Greece
Anna Sperotto, University Twente, The Netherlands
Alexey Vinel, SPIIRAS, Russian Federation

Table of Contents

Part I

Invited Talks

What You Pay Is What You Get - Economics of Quality of Experience in Fixed and Mobile Telecommunications

Peter Reichl

Telecom Bretagne, France

Overview

Advancing far beyond its engineering roots, research in telecommunications has become a holistic and interdisciplinary endeavour, addressing more and more the entire communication ecosystem and including the perspectives of business stakeholders and end user likewise. In this talk we discuss the recent paradigm change from Quality of Service (QoS) to Quality of Experience (QoE) as a key step into this direction. We review actual advances in the quest for fundamental laws of QoE which serve as starting point for a new fixed point model of QoE-based charging. Finally, we report on current user trials validating the model and allowing further insight into this exciting field where communication technology meets microeconomics and the human user.

Active vs Passive localisation strategies

D. C. Dimitrova

dimitrova@iam.unibe.ch
University of Bern, Switzerland

Although localization (and tracking) inside buildings is a well-investigated topic, there is not yet a single system that can meet the combined demands for low cost, deployment ease and ubiquitous coverage. Even the Global Positioning System (GPS), which performs well in most outdoor areas, faces problems, i.e., for most indoor or dense urban environments GPS coverage is weak or non existent due to the strong signal absorption [1]. Most indoor location systems currently in deployment rely on a radio technology such as WiFi, Bluetooth or cellular networks [2] to support positioning or tracking applications and related location-based services. The wide availability of Global System for Mobile Communications (GSM) networks motivates research on the use of GSM as a common radio technology for localization systems. In addition, GSM signals seem more stable over time in comparison with WiFi or Bluetooth signals [3].

In general, a localization solution, including a GSM-based one, can be implemented as an active or passive system. In active systems one of the participants in the radio communication is also responsible for performing the positioning process. The process can be implemented at the network side [5, 4], the terminal side [8, 7] or at both [6]. An active system has the advantage of having access to location-relevant parameters, which can significantly help the system design but at the same time requires the participation of mobile devices or network entities, i.e., network operators. A passive system on the contrary assumes no participation of the communicating parties but relies on overhearing radio (GSM) signals and their subsequent processing for the purpose of localization. Note that the system is interested in signals originating at the mobile terminals (uplink). Such system is invisible to the GSM network and the users and hence attractive for third parties, which wish to avoid dependency on network operators to provide location-based services. A passive system poses several specific challenges, not found in active systems, towards signal recovery and interpretation.

The number of proposed localisation techniques is equally great, the most often used being angulation, lateration and fingerprinting, see [10]. In angulation the location is a derivative of measured angles to fixed reference points. Lateration is based on the same concept but uses distances, which can be determined by various methods among which Time of Arrival (ToA), Time Difference of Arrival (TDoA), received signal strength (RSS) and hop count. Various modifications of each method have been proposed, e.g., [9], as well as combinations thereof. Finally, a fingerprinting technique compares on-line measurements to an off-line database in order to determine location. Currently there are so many localisation proposals based on the fingerprinting technique that a separate taxonomy such as [11] is appropriate.

Out of the above techniques working with time shows the biggest promise for accurate location estimation down to a meter or on even finer scale. Nevertheless, time-based localisation techniques such as TDoA are very sensitive to synchronisation offset between the nodes and their ability to determine with accuracy of nano-seconds the exact moments at which the signal reaches the receiver. These two requirements are already not easily met in active systems and are even more difficult to achieve in a passive system. Tackling them requires complex signal progressing and the use of specialised hardware and software. Among the other challenges a passive system for localisation that uses cellular (GSM) signals needs to deal with are synchronisation with the end user in time and frequency and recovery of user traffic (decoding, demodulation, etc) in order to retrieve data (messages) that carry meaningful for localisation information. Encryption can also pose a difficulty and limit the number of messages that can be used in the localisation process. Hence, careful analysis on the requirements towards cellular-based passive localisation system is necessary before one approaches the topic in detail. A comparative study against the requirements with an active system is moreover beneficial.

References

1. Xiangdong Lin. Enhanced accuracy gps navigation using the interacting multiple model estimator. Aerospace Conference, 4:19111923, 2001.
2. Jin-Shyan Lee et. al. A comparative study of wireless protocols: Bluetooth, uwb, zigbee, and wi-fi. Industrial Electronics Society (IECON), pages 4651, 2007.
3. Veljo Otsason et. al. Accurate gsm indoor localization. In The Proc. of UBICOMP 2005, 2005.
4. Stefan Zorn et. al. A novel technique for mobile phone localization for search and rescue applications. Indoor Positioning and Indoor Navigation, 1-4, 2010.
5. Richard Rose et. al. A gsm-network for mobile phone localization in disaster scenarios. Microwave Conference (GeMIC), pages 14, 2011.
6. Dimitri Tassetto et.al. A novel hybrid algorithm for passive localization of victims in emergency situations. Advanced Satellite Mobile Systems, pages 320327, 2008.
7. Nisarg Kothari et. al. Robust indoor localization on a commercial smart phone. Procedia Computer Science, 10:11141120, 2012.
8. Ioan Lita et. al. A new approach of automobile localization system using gps and gsm/gprs transmission. Electronics Technology, pages 115119, 2006.
9. M. Ciurana, F. Barcelo-Arroyo, and S. Cugno. A robust to multi-path ranging technique over IEEE 802.11 networks. Wireless Networks, 16:943:953, 2010.
10. C. Fuchs, N. Aschenbruck, P. Martini, and M. Wieneke. Indoor tracking for mission critical scenarios: A survey. Pervasive Mobile Computing, 7:1:15, 2011.
11. M.B. Kjaergaard. A taxonomy for radio location Fingerprinting. In Proc. of 3rd international conference on Location-and context-awareness, LoCA'07, pages 139-156. Springer-Verlag, 2007.

Mobility patterns: implications on network parameters and handover

Enrica Zola – Universitat Politècnica de Catalunya (Barcelona, Spain)

The purpose of a wireless network is to provide coverage to moving users. Knowledge about the pattern followed by mobile users in a given scenario may help network planning to guarantee service along the pathway followed by each user. In a comprehensive survey of existing mobility patterns [1], the authors prove that nodes' movements have a noticeable impact on end-to-end delay, data packet delivery ratio, and other metrics related to the Quality of Service (QoS). Thus, it is really important that the mobility pattern mimics the movements of real users so to appropriately plan the network layout and allocate resources at each cell. Both trace-based models and synthetic models can be used. Traces are those mobility patterns that are observed in real-life systems. They can be derived from the observation of real movements; to achieve this, user's log traces should be collected during a period long enough to capture periodical behaviours. Despite the ability of trace-based models to reflect real movements, they may be too specific for the environment from which they have been extracted. Moreover, sometimes traces are not available, as for example in new network environments, and it is necessary to use synthetic models. Synthetic models attempt to realistically represent the behaviours of mobile users over time while providing a simplified algorithm that describes their movements. Speed, direction and pause times are the main parameters needed to define how users move inside the simulation area. Despite the simplified and less realistic movement pattern generated, they capture enough of the key characteristics of human mobility to make protocol evaluation meaningful and easier.

In their publication in 2005 [2], the authors analysed MANET simulation studies published in a premiere conference for the MANET community between 2000 and 2005. They found that 66% of the studies involving mobility used the Random Waypoint (RWP) mobility model. Despite it has been criticized for not being representative of how humans actually move, nowadays it is still largely used in many studies [3-5]. Rojas et al. [6] validate the RWP against real mobility data. With small changes to the distributions used in the RWP (e.g., non-uniform distribution of the waypoints), the authors show that it can be used as a good model for mobility in large geographic areas such as a city. Considering that extracting a mobility pattern from real traces is complex and, anyway, it would be specific to the environment and conditions from which it has been extracted, the use of the RWP in simulation studies is widely accepted. The RWP mobility model has been deeply studied in the past. Much effort has been made to gain understanding of the implications that the use of the RWP model may cause. A formal description of the RWP model has been given [7] from which some key parameters of the model can be easily derived (i.e., the length and time of the movement between waypoints, the spatial distribution of nodes, the direction angle at a new waypoint, and the cell change rate).

In the last years, our research aimed at better understanding user mobility and its implications on network parameters. The handover procedure is essential in cellular networks. The thorough understanding of the parameters related with the handover handling as, for example, statistics on the time spent by a mobile user in the same cell, are crucial for a better planning of the network. With this aim, several issues related with the handover have been investigated. Simulations have been performed in different scenarios and under different network layouts (i.e., minimum number of antennas for coverage vs. higher density of antennas for capacity constraints) and key statistics have been extracted [8]. Trace data from the WLAN in our Campus have been also analysed in order to gain understanding on the user behaviour in real environments [9]. Among other user behaviour trends, results on the cell

residence time in an academic environment are shown which can be compared to simulation results in [8]. Also, mobility trends inside classrooms and the library, together with the frequency of users' connections have been analysed.

Taking advantage of the knowledge of the statistical properties of the RWP, the predictability of the handover in a given scenario has been recently investigated [10]. An analytical framework has been proposed which can predict the next cell to which a user can move in the near future, provided that this user is moving according to the RWP mobility model. Interest in forecasting the cell to which a device may be handed off depending on the movement pattern is twofold. First, it gives insight into properties and statistics of the mobility model. Second, and from a more practical perspective, it is useful to manage resource allocation and reservation strategies in order to smooth the HO process, which then turns in an improvement in the QoS.

References

[1] T. Camp, J. Boleng, V. Davies, "A survey of mobility models for ad hoc network research," *Wireless Communication and Mobile Computing (WCMC): Special Issue on Mobile Ad Hoc Networking*, vol. 2, no. 5, pp. 483-502, 2002

[2] S Kurkowski, T Camp, M Colagrosso, MANET simulation studies: the incredibles. ACM SIGMOBILE Mobile Computing and Communications Review 9(4), 50–61 (2005)

[3] T Ali, M Saquib, C Sengupta, Vertical handover analysis for voice over WLAN/cellular network, in Proceedings of the IEEE International Conference on Communications, ed. by, 2010, pp. 1–5

[4] C Tong, JW Niu, GZ Qu, X Long, XP Gao, Complex networks properties analysis for mobile ad hoc networks. IET Communications, 6(4), 370–380 (2012)

[5]. Q Min, R Zimmermann, An adaptive strategy for mobile ad hoc media streaming. IEEE Transactions on Multimedia, 12(4), 317–329 (2010)

[6] A Rojas, P Branch, G Armitage, Validation of the random waypoint mobility model through a real world mobility trace, in Proceedings of the IEEE Region 10 TENCON, ed. by, 2005, pp. 1–6

[7] C Bettstetter, H Hartenstein, X Pérez-Costa, Stochastic properties of the random waypoint mobility model. ACM/Kluwer Wireless Networks (Special Issue on Modeling and Analysis of Mobile Networks) 10(5), 555–567 (2004)

[8] E. Zola, F. Barcelo-Arroyo, "Impact of mobility models on the cell residence time in WLAN networks", *Proc. IEEE Sarnoff Symposium*, ISBN: 978-1-4244-3382-7, pp. 1-5, Princeton (USA), 30 March 2009

[9] E. Zola, F. Barcelo-Arroyo, "Characterizing User Behavior in a European Academic WiFi Network", to appear in *IGI Global International Journal of Handheld Computing Research (IJHCR)*, due date July 2013

[10] E. Zola, F. Barcelo-Arroyo, I. Martín-Escalona, "Forecasting the Next Handoff for Users Moving with the Random Waypoint Mobility Model", in *EURASIP Journal on Wireless Communications and Networking*, 2013:16, doi:10.1186/1687-1499-2013-16, ISSN: 1687-1499, January 2013

Mobility2.0: Co-operative ITS Systems for Enhanced Electric Vehicle Mobility

G. Heijenk

geert.heijenk@utwente.nl
University of Twente, The Nehterlands

Mobility2.0 will develop and test an in-vehicle commuting assistant for FEV mobility, resulting in more reliable and energy-efficient electro-mobility. In order to achieve a maximum impact, Mobility2.0 takes an integrated approach of addressing the main bottlenecks of urban FEV mobility: range anxiety related to the limited FEV range, scarcity of parking spaces with public recharging spots, and the congestion of urban roads. Our integrated approach means the application developed by Mobility2.0 will utilise co-operative systems to simultaneously consider these bottlenecks, so that such an optimisation can be achieved which still guarantees reliable transportation for each FEV owner. Mobility2.0 will focus on assisting the daily urban commute, which represents the bulk of urban mobility. Mobility2.0 outcomes will be the following:

- an FEV-specific multi-modal urban guidance application implemented for prolific smart-phone platforms; this application will include the integrated reservation of a suitable FEV recharging spot, while also prioritising FEVs with low battery levels for the reservation, and making optimal use of the available public transportation along the journey;
- the above application will include the capability to allow municipal/utility control over the temporal and spatial aspects of recharging; the corresponding tools will be dynamic electricity pricing and a map analysis framework;
- the project will specify the scalable broadcasting of FEV recharging spot notification over 5.9 GHz networks;
- the project will specify and contribute to standardisation the technology which enables plugged-in FEVs to act as 5.9 GHz road-side units, maintaining infrastructure connectivity via the V2G interface.

Besides FEV manufacturing, FEVs may also be produced by the conversion of traditional vehicles into FEVs. Mobility2.0 shall ensure that its results are applicable to both FEV types. The Mobility2.0 proposal name is meant to express that the co-operative electromobility technology targeted by this project is a next level concept for personal mobility. One of the partners in Mobility2.0 is the University of Twente (UT). UT's focus in the project is on Vehicular Networks. These networks allow cars to communicate between themselves and with roadside infrastructure to make road traffic safer, more efficient, and greener. Within the Mobility 2.0 project, the University of Twente will research, design, and implement protocols for wireless communications between fully electrical vehicles, nomadic devices, and roadside infrastructure to exchange information

regarding a.o. vehicle battery status and charging sport availability. Especially, extensions to the ITS-G5 standard, which can be considered as a variant to wireless LAN, optimized for vehicular networking, need to be defined. The results of the work will be integrated in a prototype system that will be tested at test sites in Barcelona and Reggio Emilia.

Part II

Regular Papers

Using Floor Plan to Calibrate Sensor Drift Error for Indoor Localization

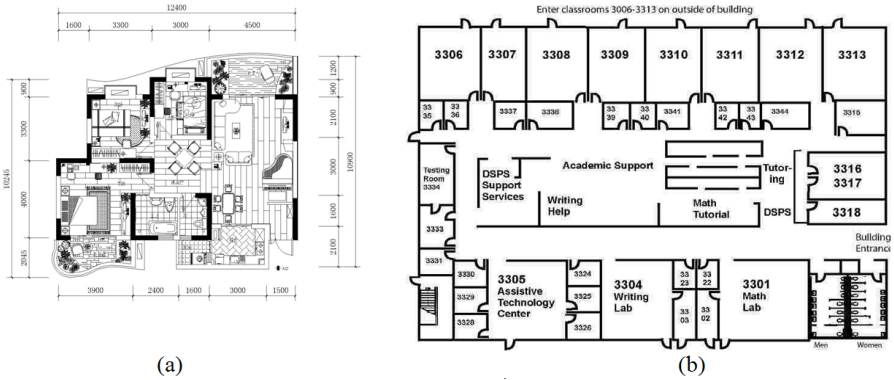
Kun-Chan Lan[1], Wen-Yuah Shih[2]
[1] Department of CSIE, National Cheng Kung University, Tainan, Taiwan (R.O.C.)
klan@csie.ncku.edu.tw
[2] Department of CS, National Chiao Tung University, Hsinchu, Taiwan (R.O.C.)
Todd629.cs01g@nctu.edu.tw

Abstract. Some prior studies proposed the use of Pedestrian Dead Reckoning (PDR) for indoor localization, in which a small number of inertial sensors are put on the pedestrian. These sensors (such as a G-sensor and gyroscope) are used to estimate the distance and direction that a user travels. The effectiveness of a PDR system lies in its success in accurately estimating the user's moving distance and direction. In our prior work, we implemented a waist-mounted based PDR method on a smart-phone that can measure the user's moving distance with a high accuracy. In this paper, we extend our prior work by designing a map matching algorithm to calibrate the direction errors from the gyro using building floor plans. The results of our experiment show that we can achieve an overall location error of about 0.48 meter.

1 Introduction

The accurate localization of objects and people in an environment has long been considered an important component of ubiquitous networking. A personal dead reckoning or pedestrian dead reckoning (PDR) system is a self-contained technique for indoor localization. This technique only requires a couple of inertial sensors to be put on the user, so that it can be used in any building without pre-installing beacon nodes or pre-building RF maps/propagation models based on surveys of the environment. These inertial sensors (such as an accelerometer, gyroscope, or digital compass) are used to measure step length and heading direction. Some PDR systems also use other sensors, such as cameras [1], ultrasound [2], RFID [8], or laser [3], to calibrate their results.

Sensor drift [6] is a well-known problem in PDR systems. Given that the hardware used in such systems is not perfect, the inertial sensors constantly have some small errors when estimating the distance and direction, and signal noise (such as vibrations of the user's body) can further exacerbate this problem [5]. Some PDR systems use a map matching mechanism to calibrate these errors [10, 16], and these can be categorized into two types. One tries to matches the user trajectory to the closest junction and road on the map [16], while the other one utilizes the map to filter out positions where the user is unlikely to move (e.g. walls, obstacles, and so on) [10]. Both techniques require the use of a detailed scaled map of the building, as shown in Fig. 1(a), although in practice this is usually difficult to obtain.



(a) (b)

Fig. 1. (a) A detailed map and (b) a floor plan

In this paper, we consider a scenario in which the user has a smart-phone and can access the floor plan of the building, like the one shown in Fig. 1(b), as these are widely available. The system utilizes the sensors on the smart-phone to compute the user's moving distance and direction, and this can be used with the map to estimate their current position in the building. Our system can be considered as an add-on to a traditional outdoor navigation app (e.g. Google Latitude [15]) so that the starting position of the user in the building can be estimated using the last-recorded GPS position.

In our prior work [20], we implemented a waist-mounted-based PDR method on the smart phone to accurately estimate the user's moving distance. In this paper, based on the geometric similarity between the user trajectory and the floor map, we design a map-matching algorithm to calibrate sensor errors using common building floor plans and show the location error is about 0.48m in our test scenario.

The rest of this paper is structured as follows: In Section 2, we describe the related work. We discuss the details of our map-matching algorithms in Section 3. The results of our experiment are shown in Section 4. Finally, we conclude this paper in Section 5.

2 Related work

2.1 PDR system

Some prior studies use a PDR system to estimate the trajectory of a user by placing some sensors on the user's body. Inertial sensors, such as accelerometers, gyroscopes, and compasses, are commonly used in such systems. Some PDR systems also include GPS sensors [4], and use these to calibrate the PDR drift as long as the GPS signal is available. When the GPS signal is obstructed, the system can then be changed to the PDR mode and continue to record the trajectory. Our study is based on the PDR system, which has the advantage of avoiding the deployment overhead of the signal-based methods. On the other hand, the performance of our system can be enhanced by a signal-based system if available. For example, one limitation of our map-matching

approach is its need to collect "enough" trajectory data before it can uniquely identify the user's position on the map. When there is not enough trajectory information, we make use of the known locations of existing WiFi base stations in the building to help estimate the user's location [9].

2.2 Map-matching algorithm

In some prior PDR systems, a map-matching mechanism is used to match the user trajectory onto the map [10, 16] in order to calibrate the sensor errors. There are two ways this is achieved. The first tries to match the user trajectory to the closest junction and road on the map [16], while the other utilizes the map information to filter out positions where the user is unlikely to walk (e.g. walls and obstacles) [10]. However, both techniques require the use of a detailed scaled map of the building (i.e. with detailed distance information for each route on the map) which, however, is usually not easy to obtain in reality. In our work, we utilize the more commonly-seen building floor plans instead of detailed scaled maps. Based on the geometric similarity between the trajectory data and the map, we propose a new map-matching method that uses the floor plan for locating the user. As shown later in Section IV, our approach has similar performance to those that rely on detailed scaled map information.

3 Sensor error calibration with map matching

The effectiveness of a PDR system lies in its success in accurately estimating the user's moving distance and direction. In the previous section, we discussed how to use zero velocity updating to reduce the sensor drift error when measuring the distance. In a PDR system, the direction in which the user is heading is most commonly obtained from a gyroscope sensor. However, a gyroscope can only produce the relative angular displacement (RAD) of a device with respect to a specific direction, and this is not necessarily the absolute direction. Therefore, while we could track the user's trajectory using the gyroscope, this trajectory might be biased by the error in its initial direction and appear as a rotated version of the true path, as in the example shown in Fig. 2. When the error of the gyroscope is significant, and if left uncorrected, it can make the entire PDR system unusable. Map-matching is the process of comparing the pedestrian's trajectory data with a digital map of the environment to match the trajectory data to the route segment on which the pedestrian is walking, and it can be used to correct the heading error of a PDR system [17]. Unlike previous map-matching approaches that require the use of a detailed scaled map (i.e., with detailed distance information for each corridor on the map) [10], which is normally difficult to obtain in practice, in this study we propose a new map-matching method utilizing the more commonly-seen building floor plans to calibrate the gyroscope errors. We assume that a floor plan is an "approximate" scaled down version of the physical layout of the floor. Our basic idea is to utilize the geometric similarity between the trajectory data and the floor plan to infer the last-visited corner by the user. The flow chart of our algorithm is shown in Fig. 3. Before starting the map matching,

we adopt an approach similar to that in a prior work [7] by first converting the floor plan into a link-node model in which information such as the turning angels of the corners and comparative ratios of the lengths between any two corridors can be estimated, as shown in Fig. 4. The link-node model is used to approximate the layout of corridors and corners. We then compare the geometry of the user trajectory with the link-node model to find the possible routes that the user has travelled. Here we consider the map and the trajectory as two independent graphs, say *M* and *T*. We list out all the sub-graphs of *M* and compare *T* with all these sub-graphs to find the most similar one. We define the "similarity" between two graphs by comparing their shapes, vertex angles and relative edge lengths. Once a unique route is identified, this can then be used to calibrate the trajectory data and identify the most recently visited corner. Since the location of every corner is known within the floor plan, the system can then localize the user while they move between corners using the dead-reckoning data from the accelerometer, as previously discussed in Section III. Note that, given that the link-node model is only an approximation of the physical layout of the building (e.g., the link length might not be an exact scaled down version of the corridor length), the results of this comparison between the map and trajectory could generate multiple candidate routes. Therefore, we also implement a RSSI-based filter by using the existing WiFi-signal-based landmarks (e.g., a corridor-corner may overhear a unique set of WiFi APs, but the set may change at short distances away from that spot; some dead spots inside a building may not overhear any WiFi signals, which by itself is a signature). When a WiFi AP signal is available, we can use this RSSI-filter to select the correct route from multiple candidates. For example, if we know the user has passed a certain landmark, we can remove those candidate routes that do not contain it. This approach is similar to the method used by a recent study [18]. The map-matching process is performed every time the system detects that the user makes a turn. Details of the turn detection process are described below.

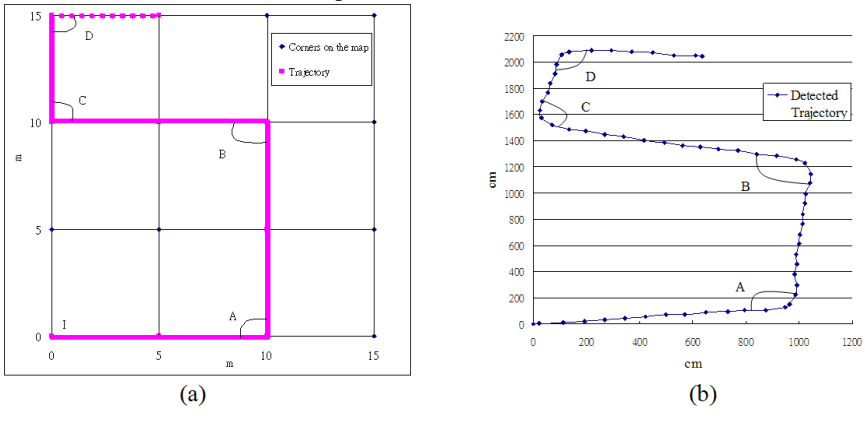


Fig. 2. (a) Link-model of the map and the ground truth (b) the estimated trajectory from the sensor data

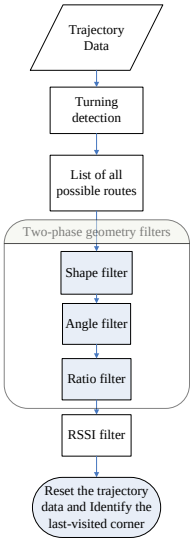


Fig. 3. The flow chart of our map-matching algorithm

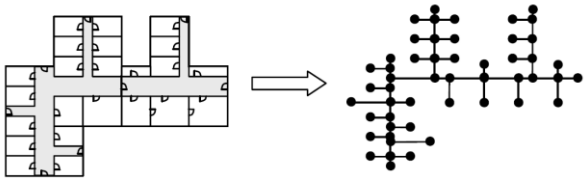


Fig. 4. The link-node model from a floor plan

3.1 Turn detection

The flow chart of our map matching algorithm is shown in Fig. 3. The first step is to find out all the turnings in the trajectory data and use these as a signature to match all possible routes on the map. However, given that the gyroscope can only provide the relative angular displacement (RAD), we use a sliding-window-based algorithm to infer the user's possible turnings from the data. To determine if a person is making a turn, we compare the standard deviation of the window with a threshold which is estimated during the period when the user is walking straight. A turning event is considered to have occurred when the standard deviation of the window exceeds the threshold. Note that, since it could take several steps to turn around a corridor corner, we record all the turning events that are related to a possible corner-turning event in order to compute the angle of making a complete turn of this corner. One example is shown in Fig. 5. The turning angle (*CT*) of a possible corner can be estimated as

$$CT = AT - BT$$

Here *AT* is the heading angle after making a turn around a corner and *BT* is the heading angle before making a turn around a corner. For the example in Fig. 5, $CT = AT - BT = 90 - 0 = 90$. However, in reality, it is not necessary that one only makes a turn when encountering a corner. For example, one might walk back and forth along the same corridor/aisle. In addition, possible gyroscope drift errors can also produce false turning events. Therefore, detection of a turning event is not necessarily an indication that the user is indeed passing a corner. We consider these kinds of turning events, which are detected when the user is not passing a corner, as 'fake' turnings. Nevertheless, it is difficult to distinguish normal corner turnings from fake one based only on accelerometer and gyroscope data. In this study we utilize the floor map information to resolve the issue of fake turnings, as follows. We assume that, once the fake turnings are removed from the trajectory data, the trajectory data should be geometrically similar to a possible route on the map.

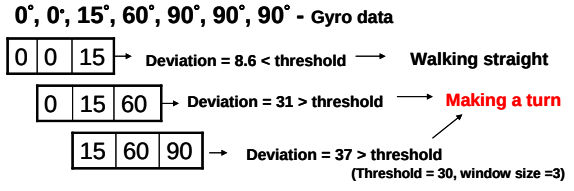


Fig. 5. An example of turning detection

3.2 List of all possible routes

We first try to find all the possible routes that a user might take (say, R_T) based on all the possible combinations of detected turnings from the trajectory data. We then compare these R_T with the all the possible routes on the map (say, R_M). The objective here is to find an ((R_T, R_M) pair which is "*the most geometrically similar*". We use some methods from image processing theory to solve this problem. We first model the map as a graph, $G_M(V_M,E_M)$. V_M is the corner of map, such as A in Fig. 2(a), and the E_M denotes the corridor between two adjacent corners. We also define the graph $G_{Mi}(V_{Mi}, E_{Mi})$ as the sub-graph of G_M, which is used to model all possible routes on a map. In addition, we model the user trajectory as a graph $G_T(V_T, E_T)$, where V_T stands for an ordered set of detected turnings (including fake ones), such as A',B',C',D' in Fig. 2(b), and E_T is the edge set whose element is the connection between two adjacent detected turnings. That is, assuming $V_T = \{V_1, V_2, \ldots, V_k\}$, V_k is the *k*-th detected turning, then $E_T = \{\overline{V_1V_2}, \overline{V_2V_3}, \ldots, \overline{V_{k-1}V_k}\}$. We next define the graph $G_{Tj}(V_{Tj}, E_{Tj})$ as follows. There exists a set V' which is the power set of V_T (the set that contains all subsets of V_T), i.e., $V' = \{\phi, \{V_1\}, \{V_2\}, \ldots \{V_1,V_2\}, \{V_1, V_3\}, \ldots \{V_1, V_2, V_3, \ldots V_k\}\}$. Here we let V_{Tj} be an ordered set , $V_{Tj} \in V'$ and $|V_{Tj}| > 1$. The E_{Tj} is the edge set which contains the edge between any two adjacent elements in V_{Tj}. In other words, G_{Tj} is used to model all possible routes that could be generated based on the detected turnings (including fake ones), and G_{Mi} stands for the accessible route on the map.

Again, the idea here is to find a (G_{Mi}, G_{Tj}) pair which is the most geometrically similar. In other words, we want to eliminate those hypothetical routes generated in G_{Tj} that can not be found on the real map. We consider two graphs are geometrically similar if they have similar shapes, angles (i.e., the angle between two connected edges) and edge lengths (after normalization). We design a two-phase filtering mechanism and employ three filters: shape filter, angle filter and edge filter. In phase one, we input all (G_{Mi}, G_T) pairs into these three filters to remove those which are not geometrically similar. The purpose of phase two is to remove the non-existing routes caused by fake turnings, and we input all (G_{Mi}, G_{Tj}) pairs into these filters to remove the non-existing routes, as shown in Fig. 3.

3.3 Two-phase geometry filters

The aim of the above two-phase filtering mechanism is to produce a (G_{Mi}, G_{Tj}) pair which is geometrically similar. However, when we do not have "*sufficient*" trajectory data (e.g., when there is no detected turning in the trajectory data), it is possible that we can still have multiple candidate routes remaining after the geometry-similarity filtering. Therefore, when multiple candidate routes exist, we employ an RSSI-based filter by using the existing WiFi-signal-based landmarks, as described previously, to further select the correct one among multiple candidates. Next, we discuss the details of each filer.

The shape filter.

We adopt the idea of a shape descriptor [11] to implement our shape filter by comparing the shapes of two graphs. Considering the nature of our input data and the computational overhead, we modify the original shape descriptor method to suit our scenario. To reduce the computational overhead, we calculate the centroid [14] of each graph and create a line every 10 degrees from 0°to 180°to pass through this. We then calculate how many crossing points on the edge can be made by each line and record them in a one-dimensional array, as shown in Fig. 6. Finally, we compare the arrays generated based the two graphs to determine their similarity using Euclidean distance (L-2 norm) [13]. We set a threshold to judge the similarity between these two graphs. If the value is over the threshold, the system will consider these two graphs are different and remove them from the set of candidates.

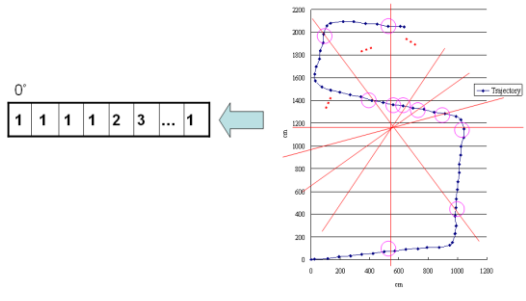


Fig. 6. Example of the shape filter

The angle filter.

We adopt the concept of chain code [12] to implement the angle filter. The chain code uses a sequence of numbers to represent a series of different moving directions and transform a graph to a one-dimensional expression. However, we can not directly apply a full-fledged chain code to our system. First, it is difficult to decide the number of sampling points for the trajectory data, since each step distance could be different. Second, the computational overhead of using the chain code is proportional to the number of steps in the trajectory data and the number of candidate routes on the map. Given that what we consider here is a real-time localization system and the computation capability of a smart-phone is limited, we can not just naively use the chain code. Therefore, we adopt the concept of the chain code and implement it separately via two different filters: the angle filter and the edge filter. Conceptually, the outputs of the chain code include the angle information (e.g., *A* and *A'* in Fig. 2), the direction of the edge and the normalizes edge ratio (i.e., divide the distance of each edge by the distance of the longest edge). In our angle filter, for example, we compare every matching angle and the direction of each edge between *GT* and GM_i, and use a threshold to determine whether they are all close enough, as shown in Fig. 7. After filtering out those GM_i which do not have similar angles, we then implement the 2nd part of the chain code with an edge filter, as described below.

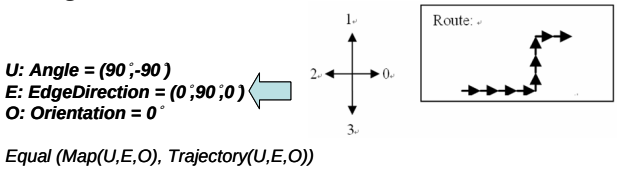


Fig. 7. Features of the angle filter

The edge filter.

With this filter, we check whether the normalized edge ratios between two graphs, for example, GT_j and GM_i, are similar or not. The system calculates the displacement between any two adjacent vertices in the graph and stores these displacements as a vector. Figure 17 shows an example input to the edge filter. We then use the Euclidean distance and set a threshold to determine whether the vectors produced for GT_j and GM_i are similar or not. If the value of the L-p norm of these two graphs is over the threshold, the system then removes the corresponding (GT_j, GM_i) pair from the candidates.

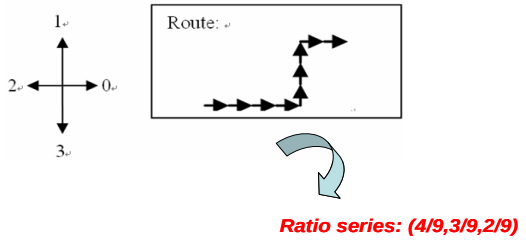


Fig. 8. Features of the angle filter

4 Evaluation

To implement our algorithm, we use a variety of Android phones from HTC and Samsung. We perform two set of experiments on the smart-phones. One is placing the smart-phone in a shirt pocket and the other is putting it in a pants pocket. We use a laser distance meter to measure the actual travel distance of the user. In addition, to record the user's actual location, we pasted markers on the ground at precisely known locations. Each of these markers had a number on it, and the user recorded the numbers when they walked passed them.

As discussed, we use a sliding-window-based algorithm to detect the user's possible turnings from the gyroscope data. To determine if a person is making a turn, we compare the standard deviation of the window with a threshold. Generally, when the chosen window size is too small, all the walking steps that happen during a turning might not be able to be included within a window. On the other hand, when the chosen window size is too big, two different turnings can be included in the same window if the number of walking steps between two corners is less than the window size. As shown in Fig. 9 and 10, the accuracy of detecting turnings becomes lower when the chosen window size is too small or too big. Therefore, in our experiments we choose the window size from the range defined below, in which D is the shortest distance between two adjacent corridors.

$$2 < WindowSize < \frac{D}{AvgStepLength}$$

And the threshold is obtained using the standard deviation of the window during the period when the user is walking straight.

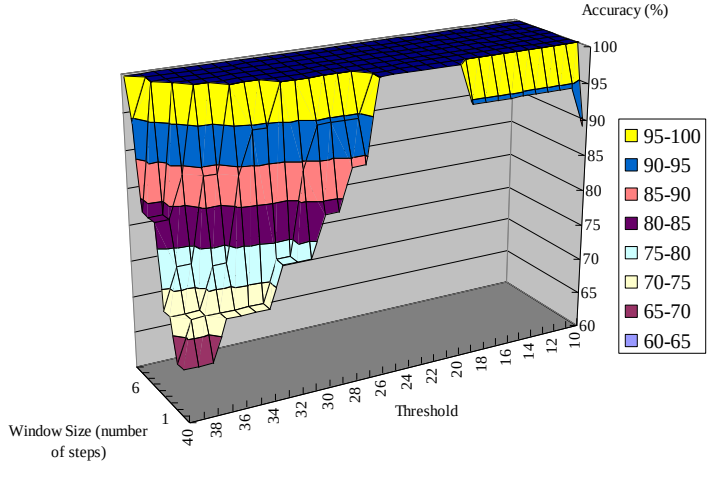


Fig. 9. The number of walking steps between two corners is larger than the window size

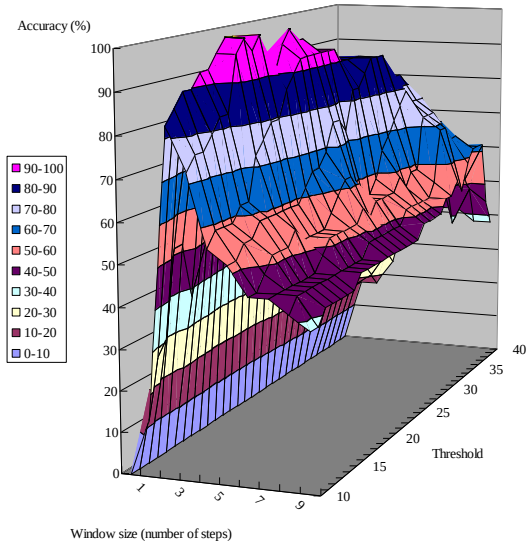


Fig. 10. The number of walking steps between two corners is smaller than the window size

In the shape filter, we set a line, every 10 degrees from 0˚to 180˚, that passes through the centroid of the graph, and use the crossing points on the edges obtained in this way to determine if two graphs have similar shapes. To understand the effects of the gaps between two crossing lines (default 10˚) on determining shape similarity, we vary this gap from 0˚to 70˚. Generally, the system will have less computational overhead when the gap is larger. However, a larger gap also suggests that poorer results will be obtained, since fewer crossing points will be generated for the comparison of shape similarity. As shown in Fig. 11, we start getting inaccurate results when the gap is larger than 15˚.

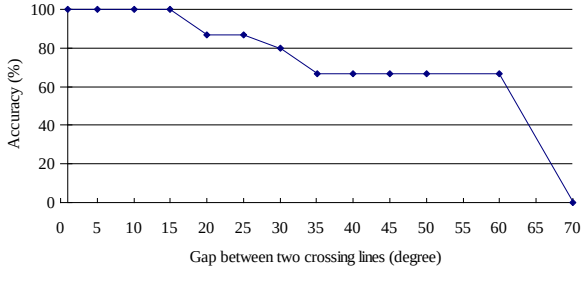


Fig. 11. The discrimination of different slopes

Finally, to test the performance of our localization system, we choose a route which is about 40 meters long and includes four corridors and corners, as shown in Fig. 12. We deliberately created some ‘fake turnings’ by having a walker to walk straight down the route but wander about around at certain spots (we put markers on these spots

before the experiment starts), shown as the green dashed circle in Fig. 12. The solid circles in Fig. 12 indicate when this walker made a 'real' turn at the corner. We repeat this experiment 10 times and our results show that our location error is about 0.48 meter, and the standard deviation is about 0.43 meter.

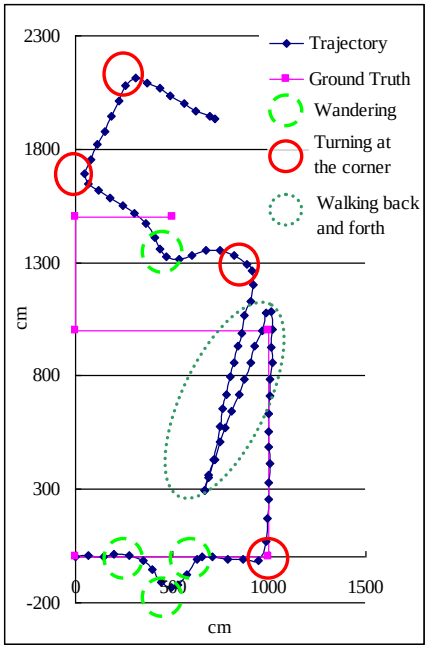


Fig. 12. The testing environment and route

5 Discussion and conclusion

In this work, we assume that the floor plan is a scaled-down version of the physical layout of the floor. In some cases, this assumption might not be always true. Here we propose a bootstrap phase based on the participatory sensing approach [19] to obtain the scale information of the map. The idea is simple: in this bootstrap phase the system first collects the users' trajectories, and then compare them with the link-model of the map to estimate the relative distance of every corridor. Note that, while the first few users may experience inferior location accuracy, a little more data will bring the system to convergence. In addition, such a bootstrap phase only needs to be performed once for each building.

To conclude this paper, we implement a waist-mounted PDR method on the smartphone by using the accelerometer in the phone to measure the user's walking distance. Furthermore, we design a map matching algorithm to calibrate the direction errors from the gyroscope using building floor plans, which are readily available. Our map matching algorithm implements three filters, namely the shape filter, angle filter and edge filter, to infer the user's last-visited corner. Our results show that we can achieve the overall location error is about 0.48 meter in our experiment.

References

1. T.S. Cinotti, L.D. Stefano, G. Raffa, L. Roffia, M. Pettinari, and M. Mola, “Dead reckoning supports stereo vision in pedestrians tracking,” presented at Fourth IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW'06), 2006.
2. F. Carl, M. Kavitha, H. Mike, and G. Hans, “Ultrasound-aided pedestrian dead reckoning for indoor navigation,” presented at Proceedings of the first ACM international workshop on Mobile entity localization and tracking in GPS-less environments, San Francisco, California, USA, 2008.
3. C. Burcu, “HeadSLAM - simultaneous localization and mapping with head-mounted inertial and laser range sensors,” in 12th IEEE International Symposium on Wearable Computers, 2008, pp 3-10.
4. S. Godha, G. Lachapelle, and M.E. Cannon., “Integrated GPS/INS system for pedestrian navigation in a signal degraded environment,” presented at ION GNSS 2006, 26-29 September 2006.
5. RaúlFeliz, E. Zalama, and J.G. García-Bermejo, “Pedestrian tracking using inertial sensors,” Journal of Physical Agents, vol. 3, 2009, pp. 35-42.
6. D.H. Titterton, and J.L. Weston, “Strapdown Inertial Navigation Technology”, 1997.
7. P.-Y. Gillieron, and B. Merminod, “Personal Navigation System for Indoor Applications,” in Proceedings of the 11th IAIN World Congress on Smart Navigation, Systems and Services, Berlin, 2003.
8. V. Renaudin, O. Yalak, P. Tom´e, and B. Merminod, “Indoor Navigation of Emergency Agents”, European Journal of Navigation, vol. 5, no. 3, 2007.
9. C. Krishna, I. Anand Padmanabha, and N.P. Venkata, “Indoor localization without the pain,” presented at Proceedings of the sixteenth annual international conference on Mobile computing and networking, Chicago, Illinois, USA, 2010.
10. Tomoya Ishikawa, Masakatsu Kourogi, Takashi Okuma, Takeshi Kurata, “Economic and Synergistic Pedestrian Tracking System for Indoor Environments,” 2009 International Conference of Soft Computing and Pattern Recognition, pp.522-527, 2009.
11. Z. Hao, and J. Malik, “Learning a discriminative classifier using shape context distances,” in Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on, 2003, pp I-242-I-247 vol.1.
12. J.A. Saghri, and H. Freeman, “Analysis of the Precision of Generalized Chain Codes for the Representation of Planar Curves,” Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. PAMI-3, 1981, pp. 533-539.
13. N. Dunford and J. T. Schwartz, “Linear operators,” volumn I, Interscience Publishers, 1958.
14. Johnson, R. A. “Modern Geometry: An Elementary Treatise on the Geometry of the Triangle and the Circle.” Boston, MA: Houghton Mifflin, pp. 173-176, 249-250, and 268-269, 1929.
15. Google Latitude Available: http://www.google.com/intl/en/mobile/latitude/
16. D. Gusenbauer, C. Isert, Kro, x, and J. sche, “Self-contained indoor positioning on off-the-shelf mobile devices,” in Indoor Positioning and Indoor Navigation (IPIN), 2010 International Conference on, pp 1-9
17. Aggarwal, P., Thomas, D., Ojeda, L. & Borenstein, J., 2011. Map matching and heuristic elimination of gyro drift for personal navigation systems in GPSdenied conditions. Measurement Science and Technology, 22, 025205.
18. H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef, and R. R. Choudhury. No Need to War-Drive: Unsupervised Indoor Localization. In Mobisys, 2012.
19. P. Mohan, V. N. Padmanabhan, and R. Ramjee, “Nericell: rich monitoring of road and traffic conditions using mobile smartphones,” in Proceedings of the 6th ACM conference on Embedded network sensor systems, ser. SenSys ’08. New York, NY, USA: ACM, 2008, pp. 323–336.
20. Lan, Kun-Chan, and Wen-Yuah Shih. "Using simple harmonic motion to estimate walking distance for waist-mounted PDR", 2012 IEEE Wireless Communications and Networking Conference (WCNC), 2012

Linux-based Measuring Platform for Time-Based Location Observables in IEEE 802.11 Networks

Israel Martin-Escalona[1], Francisco Barcelo-Arroyo[1], Enrica Zola[1]

[1] Dept. Enginyeria Telemàtica, Universitat Politècnica de Catalunya (UPC)
c/ Jordi Girona 1-3, C3-214, 08034 Barcelona, Spain
{imartin, barcelo, enrica}@entel.upc.edu.com

Abstract. Positioning has been a hot topic in research for several years. GPS is accepted as a global solution for positioning outdoors, but indoor positioning still remains an open issue. Time-based multilateration techniques are presented as a good trade-off between performance and complexity for indoor positioning. Although several proposals for time-based multilateration has been presented, only some of them has been really implemented and of them, only few can be reproduced by other researchers (mainly due to hardware customization). This work presents a system for measuring time-based location observables in IEEE 802.11 networks. This measuring system has been implemented in Linux, so it can be deployed easily by any researcher. The current implementation supports the measurement of two kind of observables: round trip times (for two-way time-of-arrival techniques) and passive TDOAs (for the passive TDOA location technique). First experiments, presented in this paper, are focused on demonstrating the feasibility of the system for measuring these two location observables.

1 Introduction

Positioning has been a hot topic in research for several years. Firstly, location solutions were addressed to professional fields, for tracking items and persons. With GPS receivers becoming more and more affordable, several location based services appeared: in-car navigation, parcel tracking, etc. GPS provided a global solution for positioning, with excellent accuracy, global coverage and good availability. However, those good skills are sided to positioning outdoors. Performance of GPS indoors is much the contrary: poor accuracy and availability, and eventually, even no positioning at all.

Several location techniques were proposed for indoor positioning, with the goal of providing a performance close to the one provided by GPS outdoors. First proposals were based on taking advantage of the infrastructure deployed for the public land mobile networks. The Enhanced Observed Time Difference (E-OTD) and the Observed TDOA (OTDOA) [1] were solutions based on measuring the time-difference of arrival of a signal sent from several base stations, proposed for GSM and

UMTS networks respectively. Their strength is in the vast availability of the terrestrial networks in which they are based. However, the poor accuracy achieved, with average positioning errors in the order of hundreds of meters, discourage the network operators to support them.

The standardization of the smartphones pushed indoor positioning to the next level. With mobile devices including lots of sensors (e.g. GPS, gyroscope, accelerometer, compass, etc.) and supporting a lot of network technologies (e.g. GSM, UMTS, LTE, Bluetooth, IEEE 802.11, NFC, etc.), location techniques had a large amount of data that could be used for fixing the customer position, either outdoors and indoors . GPS is almost the standard technique used for outdoors. However, there is no a reference technique for indoor positioning. Lots of solutions have been proposed for play this role. Most of them are based on personal and local area communication networks.

Recent works propose using the Bluetooth network for computing the position of mobile devices [2]. Bluetooth present several advantages compared with other technologies, such as low energy consumption and inquiry capabilities. However, the coverage offered by most of the devices goes below 10 meters, which is usually not enough for most of the location based services indoors. IEEE 802.11 is probably the preferred technology for indoor positioning, mainly due to its excellent coverage (one hundred meters per access point) and availability (IEEE 802.11 networks deployed almost everywhere). However, the IEEE didn't account for positioning in the network definition. Therefore, location solutions proposed for IEEE 802.11 observe network parameters in the device to be located, so that they can be subsequently used for computing the device's position. The received signal strength indicator (RSSI) is one of these parameters.

There are two approaches for using RSSI in positioning: fingerprinting and multilateration. Fingerprinting [3] consists in setting a grid of points in which the RSSI of all the access points at sight are measured. These data together with the positions associated with them are then stored in a database. When a position is requested, the involved device measures the RSSI of all the access points at sight and reports such data to a location server, which matches the reported data with the stored information. The most feasible position according to the reported data is then returned as the device's position. Fingerprinting offers good performance, with short response times and accurate positions (depending on the access-point density) but it requires building the database before deploying the location solution. Moreover, changes in the environment involve updating the data stored in the database, which usually involves some cost and impact in the performance of the location system.

Multilateration does not involve off-line stages such as fingerprinting. This approach consists of computing the distances from the device to locate to several access points. Distance can be inferred from the RSSI if the radio model is known [4]. However, two issues often discourage using RSSI for estimating distances: radio model knowledge and signal variability. The first reveals that it is hard to precisely define the radio model, since one single scenario can involve several different radio propagation conditions. Even if the radio model is completely characterized, the

received signal strength tends to be extremely variable, which leads to inconsistent positions in terms of accuracy. Distances can also be estimated by means of the time of flight, i.e. the time elapsed from a signal transmission until its reception [4]. Time of flight tends to be more consistent than RSSI and consequently favored in location solutions. Unlike the RSSI, which is a provided parameter in most of the wireless technologies, the time of flight needs to be usually estimated. Time-of-flight estimation is hard to achieve in non-synchronized networks, such as the IEEE 802.11 networks. Measuring the round-trip-time instead of the time of flight usually overcomes this issue, at the cost of dealing with noisier measurements [5].

This paper provides a measuring system for positioning based on Linux. This system measures time-based location observables in IEEE 802.11 networks and it is aimed at providing accurate measurements for a wide range of location time-based observables. The rest of the paper is structured as follows. Section 2 provides introduces the measuring systems addressed to time-based observables collected in IEEE 802.11 networks. Section 3 provides a short description of the network architecture used in Linux for IEEE 802.11 provisioning. Section 4 gives a detailed description of the design of the solution proposed for measuring time-based location observables. Section 5 provides a short description of the location observables supported by the measuring platform and section 6 presents the first results achieved. Finally, section 7 draws the main conclusions.

2 Measuring systems for time-based positioning in IEEE 802.11

There are only few measuring systems proposed for IEEE 802.11 networks. They can be classified according on the layer in which the measurements are taken: PHY, MAC and APP. PHY measurement systems include specific hardware for computing the time-of-flight (or any other related metric). Using hardware customized for time-of-flight estimation offers the best performance in terms of accuracy, but it involves costly solutions and often forces the user to carry a location device (i.e. a tag) in addition to the legacy device he/she uses. The solution proposed in [6] is an example of this approach, where the authors use an FPGA connected to a wireless 802.11 card to analyze the frames on the fly and compute the round trip time of those frames from the device to the access point and back again to the device. Although accurate positions are achieved (errors close to 1 meter), legacy IEEE 802.11 hardware needs to be modified, which is not always possible.

The rest of measurement systems (i.e. those that are not PHY measurement systems) use the hardware capabilities available in the device to compute the time-of-flight (or any related metric). Thus, the software running in the device is modified to allow timestamps to be computed. This means that legacy devices can be used for positioning if software modifications can be run in them. The feasibility of these software-based approaches is studied in [7], according to the specific layer (according to the OSI reference model) in which timestamps are computed. It can be seen that this software approach is feasible, but with errors much higher than those achieved in

PHY-based measurement systems. APP measurement systems compute the observables at layers higher than the link layer. Application layer is often used for this purpose mainly due to the easy development, the great upgradeability and the portability to several architectures and technologies. However, the higher the layer in which timestamps are added, the noisier the measurement. In [8], authors report positioning errors higher than 4 meters, which also means ranging errors larger than 4 meters in the best case. The best trade-off between complexity and performance is achieved when timestamps (and measurements) are computed in the MAC layer (i.e. link layer according to the OSI reference model). Two approaches can be followed in MAC measuring systems in the case of IEEE 802.11: TSF and legacy clocks. The IEEE 802.11 protocol stack includes the Timing Synchronization Function (TSF), which adds a timestamp to each received frame. The TSF provide timestamps in the order of microseconds, which means estimating distances with errors in the order of hundreds of meters, clearly higher than the required for indoor positioning. Another approach consists in using the legacy clocks provided by the device in the MAC layer for adding timestamps to the exchanged frames. In [9], the authors proposed modifying one Linux driver to include timestamps computed using the CPU clock. Although this solution is feasible in terms of positioning, it only supports one specific manufacturer of IEEE 802.11 devices, which limits the upgradability and portability of the solution.

The measurement system proposed in this paper is aimed at providing a generic, upgradeable and portable solution for estimating any kind of time-based location observable. This solution will be implemented using the Linux operating system. Thus, the next section will provide a detailed explanation of the current Linux network architecture, so that the reader can understand the modifications introduced by the designed measurement system.

3 Linux network architecture

The Linux operating system provided excellent support for network operations since it was born. Currently, it is the preferred operating system for running most of the high-load network services. The network architecture of the Linux operating system is quite complex, but it can be summarized according to Fig. 1. Network devices are managed by means of drivers. Those drivers are loaded once the Linux kernel is running and waits for events, i.e. requests for transmitting or receiving data. Those data are usually generated or consumed by user's applications. Those applications use the socket library provided by the operating system to exchange data with the TCP/IP protocol stack (or any other running in the machine), which implements all the transmission and reception chains according to these standards. The TCP/IP stack in Fig. 1. includes the IEEE 802.11 protocol stack, even if they are formally different stacks with separately implementations.

The first implementations of the IEEE 802.11 stack in Linux did not provide most of the MAC facilities, which are assumed to be provided by the network device

hardware. This approach simplified the development of device drivers and their portability to different operating systems, but constrained the consistency of the network operations (i.e. different cards working different under the same conditions). The SoftMAC approach was proposed to overcome this issue. The SoftMAC consists of implementing most of the MAC functionalities of the IEEE 802.11 stack, common to all device drivers, as a new software layer. Thus, the network operation becomes much more consistent and easier to maintain and debug.

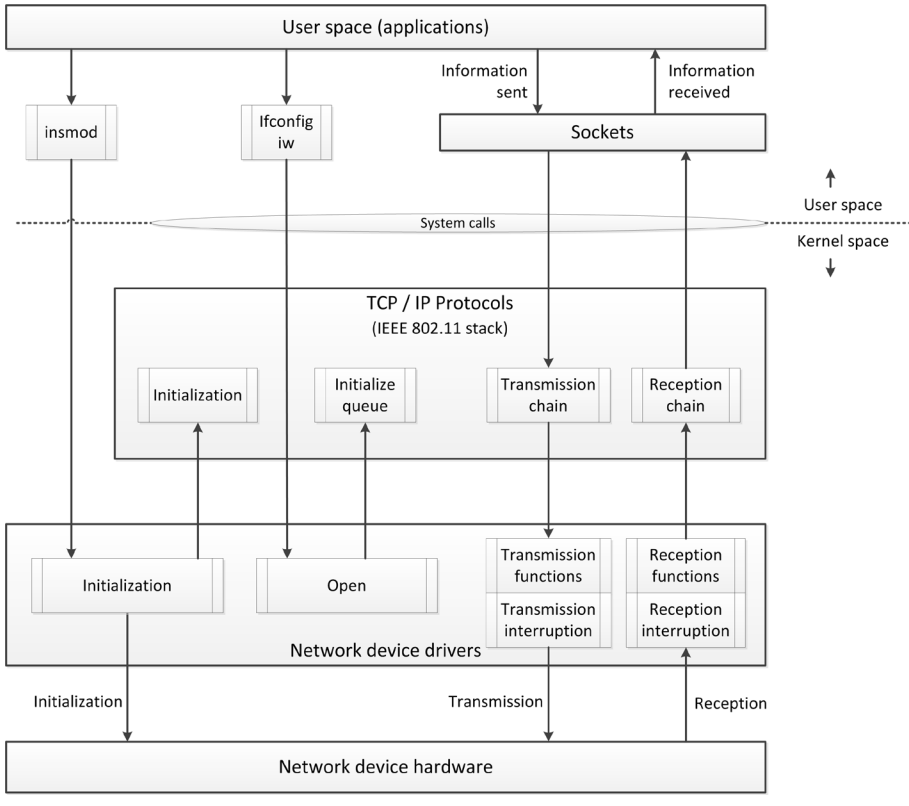


Fig. 1. Network architecture used in the Linux operating system

There are two main implementations of the SoftMAC architecture in Linux: net80211 and mac80211. Net80211 is a partial port of the FreeBSD SoftMAC implementation, which only supports the hardware made by Atheros. Net80211 is bundled together with the *madwifi* driver, which actually supports the hardware operations of the cards including the Atheros chipsets. Focusing the SoftMAC solution in one single manufacturer hardware was perceived as a drawback by the Linux community.

The net80211 framework was succeeded by the mac80211, which is the framework finally included in the main branch of the Linux kernel for the SoftMAC

implementation. Fig. 2 shows the main blocks in which the mac80211 framework consist of: the *cfg80211* and *mac80211* modules. The first one is in charge of managing the network cards parameters (e.g. setting the BSSID), while the mac80211 module represents the core of the SoftMAC implementation, i.e. implement most of the MAC functionalities common to all the SoftMAC device drivers. The *nl80211* and the *cfg80211_ops* interfaces were introduced for exchanging data between user's applications and the SoftMAC modules (i.e. cfg80211 and mac80211). Wireless extensions (i.e. known as *wext*) are still maintained for backward compatibility, but it is expected to be removed in a near future.

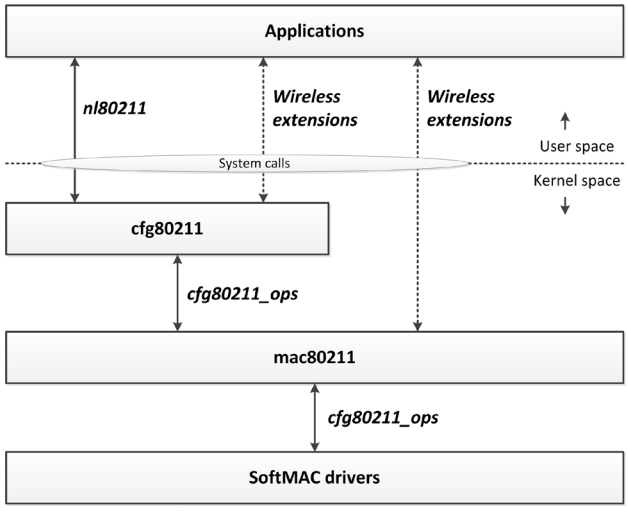


Fig. 2. Module and interfaces used by the IEEE 802.11 stack in the Linux operating system

4 The Proposed Linux-Based Measuring system

The measuring system proposed in this paper is aimed at:

- Hardware independence, meaning that the system must support any kind of time-based location observable.
- Accurate, i.e. trying to provide location data as accurate as possible.
- Upgradeable, in the sense that the system has to provide a fast and easy way to include new features and to fix the issues of the existing ones.
- Portable, meaning that the location system could be implemented in different hardware architectures.

The design of the measuring system is made accounting for the features stated above. Fig. 3 shows the main blocks in which design of the measuring system consists of. Measuring capabilities have been introduced in the mac80211 module. This

emplacement allows the measurement system to support any legacy hardware without further modifications, which basically addresses the goal of being independent of the used hardware.

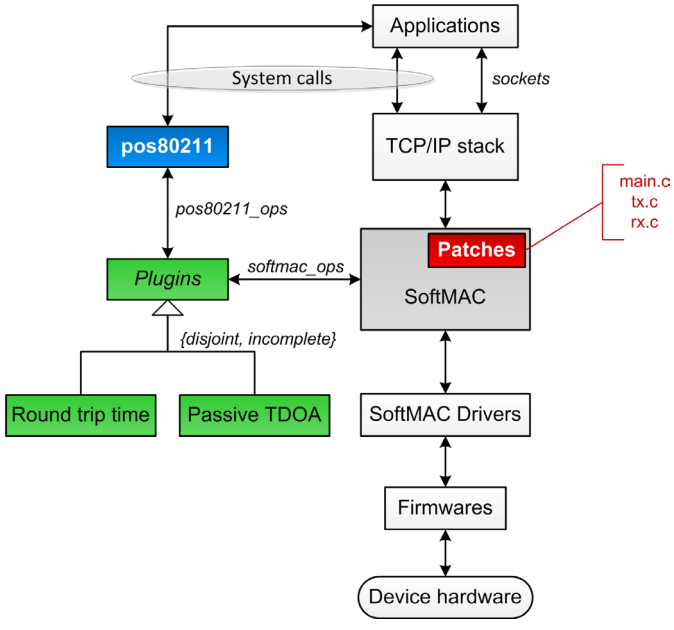


Fig. 3. Design of the measuring platform, including the enhances to the mac80211 framework

The measuring capabilities are structured as plugins. All the plugins implement an interface, named *softmac_ops*, which defines the basic functions that any plugin must implement: transmission and reception handlers. The first one runs every time the transmission of a frame is request, previously to forwarding the request to the SoftMAC driver. The reception handler is run as soon as a new frame is received in the SoftMAC layer. The *softmac_ops* functions implemented by plugins are called from the mac80211 module, when a transmission or a reception request is handled.

The sources of the mac80211 module are patched to include the registration and deregistration of the measuring plugins. The implementation of these patches is aimed at reducing the amount of files and lines of code updated, so that these patches can be easily applied to different architectures and versions of the Linux kernel. Accordingly, only three mac80211 files are impacted by the patches: *main.c*, *tx.c* and *rx.c*. Few details about these files can be found in the Table 1.

A new module is included in the SoftMAC architecture so that the location measurements can be accessed from user applications. This new module creates a virtual character device which is used to send commands to the measuring system and collect the data from it. All the plugins must again implement an interface, called *pos80211_ops*, which defines the operations that they must support in order to exchange data with the pos80211 module and consequently with user applications. The operations defined by the pos80211_ops are related with the system calls

attended by the pos80211 module: *open*, *release*, *read* and *unlocked_ioctl*. Table 2 summarizes the purpose of these operations. The current implementation of the pos80211 module has one constrain: exchanges data with one single plugin. The plugin to which the module in linked is indicated as a parameter when module is loaded. Further versions of the module are expected to overcome this limitation.

Table 1. Explanation of the patches introduced by the measuring system

File	File description	Code	Patch description
main.c	Un/allocate the mac80211 resources	6 lines	Un/register all the plugins available in the measuring system
tx.c	Transmission chain at the mac8011 layer	6 lines	Calling the specific transmission handler for each of the registered plugins, when a transmission event happens.
rx.c	Reception chain at the mac8011 layer	5 lines	Calling the specific reception handler for each of the registered plugins, when a transmission event happens.

Table 2. Operations defined by the pos80211_ops interface

Operation	Description
open	It initializes the device for exchanging the information. If the device is already opened, an error is returned
release	It is the counterpart of the open system call
read	It handles the demand of RTTs. The reading process is blocking, so that the system call impacts neither the kernel nor the user-space application performance
Unlocked_ioctl	It provides a way for sending commands to the platform. The only command currently available resets the platform.

5 Supported observables

The current implementation of the measuring system support two kind of observables: round trip times and passive time-difference of arrivals. The next sections provide details about the network operations required for measuring them and how the plugins responsible for collecting such observables are implemented.

5.1 Round trip times (RTTs)

RTTs are used in two-way time-of-arrival (TOA) techniques to infer the time of flight and subsequently the distance between two nodes (e.g. a node and an access point in the case of IEEE 802.11). Fig. 4 shows the procedure followed in the IEEE 802.11

network to measure the RTT. A data frame is sent from the node to be located to the access point, which sends back the corresponding acknowledgment (ACK). The RTT is computed as the time from the data frame is sent until the ACK is received. This RTT is computed using the timestamps added to the transmitted and received frames. These timestamps can be calculated either in nanoseconds or in amount of CPU cycles.

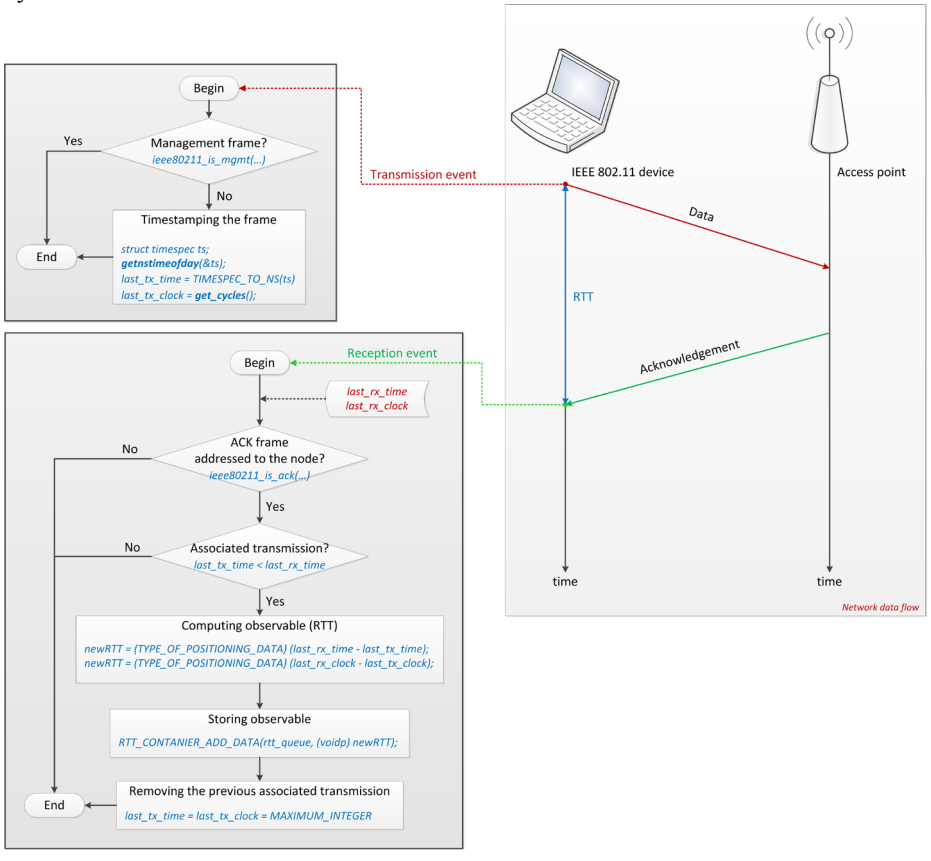


Fig. 4. Implementation of the transmission and reception handlers of the round-trip-time plugin

5.2 Passive time-difference of arrivals (passive TDOAs)

Passive TDOAs are the observables of the passive TDOA technique [10], which performance is described in Fig. 5. The procedure followed to compute passive TDOAs is similar to the one described for the RTTs. The main difference is that two different received frames are handled for computing a single passive TDOA, one for the data transmitted by the active node to the access point and the other one for the ACK sent back by the access point to the active node.

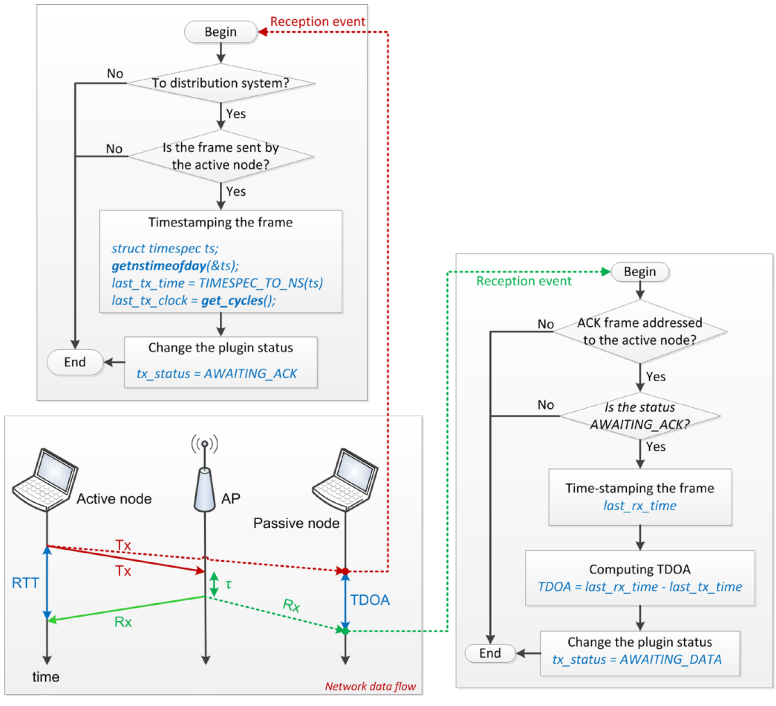


Fig. 5. Implementation of the transmission and reception handlers of the passive TDOA plugin

6 First results

A simple experiment has been proposed for assessing the feasibility of the measuring system. Two laptops have been set together in a square shaped area. An IEEE 802.11g access point is placed *d* meters far away from these two laptops. Distances *d* from 1 meter to 10 meters in steps of 1.5 meters are evaluated. 10000 pings are sent from one laptop to the access point and this procedure is repeated 10 times for the same distance *d*, yielding to 10x10000 pings. Pings do not overlap one to each other. Laptops are dedicated to positioning only. Few IEEE 802.11 networks are at sight, so slightly interference is expected. The next sections evaluate the performance of the observables collected by the two plugins.

6.2 Round trip times

Fig. 6 shows the average RTT of the collected observations for each of the 10 runs. Average RTT grows along with the distance, which demonstrates the feasibility of the measuring system. Averaging raw observables as shown in Fig. 6.a produce quite noisy measurements (i.e. variable). Filtering the data with a simple Gaussian filter that removes those data higher than 95th percentile produces the results shown in Fig 6.b,

the expected linear behavior along with the distance is much more clear. More aggressive filters will be necessary to use these measurements in two-way TOA multilateration techniques.

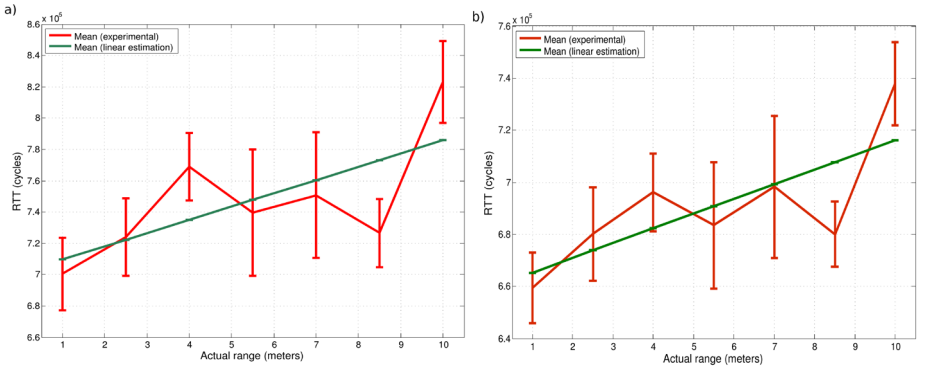


Fig. 6. Average RTT along with distance with a) no filtering and b) Gaussian filtering

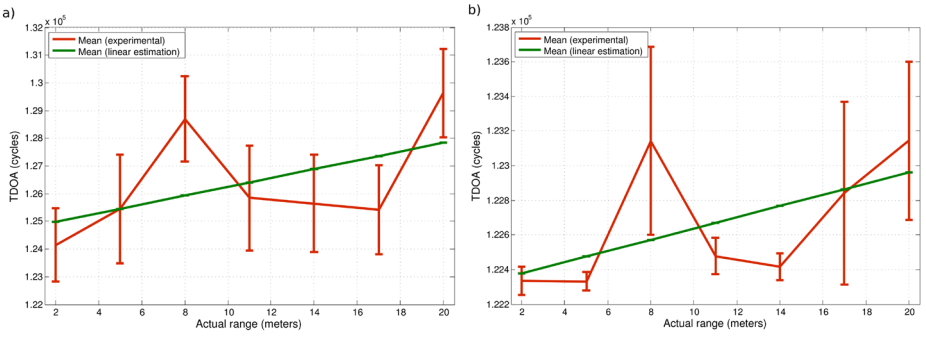


Fig. 7. Average TDOA along with distance with a) no filtering and b) Gaussian filtering

6.3 Passive time-difference of arrivals

The measurements coming from the passive TDOA plugin are more stable if compared with those produced by the RTT plugin. Notice that, since both laptops are one close to the other, estimating passive TDOAs is equivalent to estimating RTTs. Fig. 7 shows the average passive TDOA computed by each of the 10 runs. In relative terms, the error associated with the estimation is comparable with the achieved in the case of the RTT. However, the absolute values are one order of magnitude below those reported for the case of the RTT plugin. This is due to the fact that received events are attended in first place and impacted thus by less random latencies. In the case of filtering the results, the variability (and thus the error) of the measurements is reduced another order of magnitude, which yields to measurements at different distances to be closer one to each other. Again, a good filtering stage, either in the measurement system or in the ranging model becomes mandatory.

7 Conclusion

This paper presents a measuring system for time-based location observables for being used in IEEE 802.11 positioning solutions. The modular design allows enhancing the system with further observables without impacting the rest. First results of the implementation made in Linux demonstrate the feasibility of the system for measuring RTTs and passive TDOAs. Further research on aggressive filtering to remove the delays impacting the measurements is required for using the measurements to feed multilateration algorithm.

Acknowledgements

This research was funded by the ERDF and the Spanish Government through project TEC2009-08198 and the Catalan Government through the project 2010VALOR-00065

References

1. 3GPP: Stage 2 functional specification of User Equipment (UE) positioning in UTRAN. 3GPP TS 25.305(v5.6.0). (2003).
2. Dimitrova, D. C., Burgi, U., Martins Dias, G., Braun, T., Staub, T.: Inquiry-based bluetooth parameters for indoor localisation - an experimental study. ERCIM workshop on e-Mobility, (2001) 13-21.
3. Kjærgaard, M. B.: A taxonomy for radio location fingerprinting. Lecture Notes in Computer Science. , 4718 (2007), 139-156.
4. Gu, Y., Lo, A., Niemegeers, I.: A survey of indoor positioning systems for wireless personal networks. IEEE Communications Surveys Tutorials.11(1) (2009) 13-32.
5. Günther, A., Hoene, C.: Measuring Round Trip Times to Determine the Distance Between WLAN Nodes. Lecture Notes in Computer Science. 3462 (2005), 768-779.
6. Ciurana, M., Barcelo-Arroyo, F., Izquierdo, F.: A ranging system with IEEE 802.11 data frames. IEEE Radio and Wireless Symposium. (2007), 133-136.
7. Muthukrishnan, K., Koprinkov, G.T., Meratnia, N., Lijding, M.E.M.: Using time-of-flight for WLAN localization: feasibility study. Electrical Engineering, Mathematics and Computer Science (EEMCS). (2006).
8. Hoene, C., Willmann, J.: Four-way TOA and software-based trilateration of IEEE 802.11 devices. IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC). (2008), 1-6.
9. Ciurana, M., Giustiniano, D., Neira, A., Barcelo-Arroyo, F., Martin-Escalona, I.: Performance stability of software ToA-based ranging in WLAN. International Conference on Indoor Positioning and Indoor Navigation (IPIN). (2010), 1-8.
10 Martin-Escalona, I., Barcelo-Arroyo, F.: A New Time-Based Algorithm for Positioning Mobile Terminals in Wireless Networks. EURASIP Journal on Advances in Signal Processing. 2008(1) (2008), 1-10.

A framework towards adaptable and delegated end-to-end transport-layer security for Internet-integrated Wireless Sensor Networks

Jorge Granjal, Edmundo Monteiro, Jorge Sá Silva

DEI/CISUC, University of Coimbra
Polo 2, Pinhal de Marrocos, 3030-290 Coimbra, Portugal
{jgranjal, edmundo, sasilva}@dei.uc.pt

Abstract. Sensing applications envisioned for the Internet of Things (IoT) are expected to employ constrained wireless sensing devices and require appropriate security mechanisms protecting end-to-end communications between Internet hosts and much more constrained wireless sensing devices. While noting that technologies designed at standardization groups such as 6LoWPAN (IPv6 over Low power Wireless Personal Area Networks) and CoRE (Constrained RESTful Environments) are able to support end-to-end standard Internet communications involving constrained wireless sensing devices, several important security issues remain to be addressed before interconnecting a WSN with the Internet. In this paper we discuss a framework supporting the usage of mechanisms providing secure communications in the context of Internet-interconnected WSN. We also address the usage of adaptable end-to-end transport-layer security supporting delegated ECC public-key authentication. The proposed mechanisms are experimentally evaluated against their impact on the energy of constrained sensing devices.

Keywords: End-to-end transport-layer security, wireless sensor networks, delegated public-key authentication, 6LoWPAN, DTLS, CoAP.

1 Introduction

It is reasonable to expect security to represent one important enabling factor of a future Internet encompassing wireless sensing applications. In addition to the difficulties of designing appropriate security mechanisms for Wireless Sensor Network (WSN) environments employing constrained sensing devices, its integration with the Internet will certainly bring new challenges. On the one side, future WSN applications may employ a myriad of sensing platforms that are very different in respect of its characteristics and capabilities. On the other, the integration of WSN with the Internet will also bring many new challenges to WSN security, since such networks must be protected from Internet-originated threats, while at the same time supporting end-to-end communications with external (i.e. Internet) entities when required. Some wireless sensing platforms may be unable to support standardized security approaches such as public-key authentication, a limitation we may currently observe with sensing platforms such as the TelosB [1]. Security mechanisms designed

for constrained sensing applications are usually required to carefully optimize its usage of energy and computational power, so as not to compromise the goals of sensing applications. Applications and protocols must be well adapted not only to the limitations of sensing platforms but also to the characteristics of communications in low-energy personal area networks (LoWPAN) environments as IEEE 802.15.4 [2].
In contrast with the previous conception of a WSN as targeting the deployment of isolated sensing applications, recent communication technologies from working groups as the 6LoWPAN (IPv6 over Low Power Personal Area Networks) [3] and CoRE (Constrained RESTful Environments) [4] of the IETF will enable standard Internet communication technologies in LoWPAN (WSN) domains. 6LoWPAN provides adaptation mechanisms to enable the transmission of IPv6 packets over IEEE 802.15.4 [2] LoWPANs, while CoRE is currently designing the Constrained Application Protocol (CoAP) [5] with the goal of enabling RESTful web communications in such environments. The CoAP protocol currently adopts the Datagram Transport Layer Security (DTLS) Protocol [6] to secure application-layer messages. While DTLS provides a solution for end-to-end transport-layer security in the context of Internet-integrated WSN, various other issues remain to be addressed for this integration to be feasible from the standpoint of security. In this paper we present a framework supporting quantifiable and controlled security for end-to-end communications in the context of Internet-interconnected WSN. With such goals in mind, we also present a system architecture supporting adaptable end-to-end transport-layer security with delegated ECC public-key authentication.
The paper proceeds as follows. Section II describes related work, and Section III proposes a framework for the usage of end-to-end transport-layer security in the context of Internet-interconnected WSN. Section IV discusses the usage of transport-layer security in various usage modes in the context of our framework, which we experimentally evaluate in Section V. Section VI concludes the paper and discusses future work.

2 Related Work

Security for Wireless Sensor Networks (WSN) is a prolific research area, mostly focused on the usage of WSN supporting closed or isolated sensing applications. Many existing proposals were designed without considering the integration of WSN with the Internet, and thus usually target lower-layers (link-layer) communications [7]. Research proposals considering the integration of WSN with the Internet using standard communication approaches are more recent, as we proceed to discuss.
One of the first proposals targeting end-to-end transport-layer security for WSN is Sizzle [8]. Sizzle implements a compact web server supporting HTTP communications protected by SSL with 160-bit ECC (Elliptic Curve Cryptography) keys. The limitations of Sizzle are that it doesn't support mutual authentication and also that it requires a reliable transport-layer protocol, which may be inappropriate for LoWPAN environments. In particular, CoAP and 6LoWPAN are currently defined for the User Datagram Protocol (UDP) only. A similar proposal is SSNAIL [9], which on the other end supports mutual authentication using ECC but still required a reliable transport-layer protocol.

More in line with our approach to end-to-end security, authors in [10] address the compression of DTLS headers with the goal of reducing the overhead of communications. Even considering that header compression is a good approach for constrained LoWPAN environments, this proposal presents the limitations of being incompatible with the current DTLS specification [6] and of requiring modifications to the network stack of devices already supporting DTLS. Authors in [11] propose an architecture supporting DTLS with mutual RSA authentication. This architecture doesn't support ECC, and also requires that sensing devices are able to employ specialized trusted-platform modules (TPM) supporting public-key RSA cryptography. Therefore, it is incompatible with the current 6LoWPAN and CoAP specifications and also inappropriate to its employment with off-the-shelf sensing platforms, as is our goal.

3 A framework for the usage of secure end-to-end transport-layer communications with Internet-integrated sensing applications

Although a communications and security architecture supporting Internet-integrated LoWPAN environments is currently not completely defined, one may expect it will employ complementary security mechanisms at different protocol layers. Such an approach is already visible in the design of LoWPAN technologies such as 6LoWPAN and CoAP addressing communications at particular protocol layers. Contrary to the current Internet security architecture, in the context of which new mechanisms are usually designed for devices without serious resource constraints, mechanisms appropriate to Internet-integrated WSN must be carefully designed to cope with the characteristics and limitations of wireless sensing devices and low-energy wireless communications. With such limitations in mind, we approach the design of a framework to enable the design and evaluation of mechanisms supporting quantifiable and controllable end-to-end security for Internet-integrated sensing applications. This framework is illustrated in Figure 1.

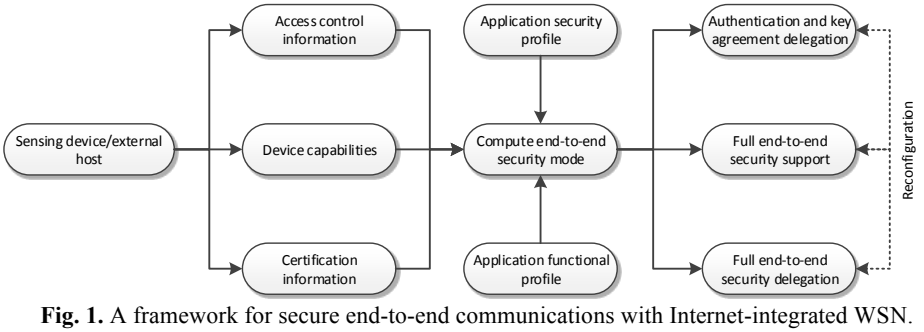


Fig. 1. A framework for secure end-to-end communications with Internet-integrated WSN.

One important goal of this framework is to enable end-to-end security mechanisms that are able to adapt to the resources available in WSN environments, in order to limit the impact of security on energy and communications, two scarce resources on LoWPAN environments. In this context, applications may be able to decide on the security mode

to be employed, in a per-device basis and considering requirements predefined for the application at hand. The following aspects are observed in the framework illustrated in Figure 1:

- For each device in the context of a given sensing application we may publish the information required for the usage of new security mechanisms, namely access control information, information about its capabilities, and an identifying public-key certificate.

- For each application, we may also consider its security and functional requirements. The security mode to employ may be decided considering such requirements, together with the knowledge about the characteristics or capabilities of the devices at hand.

- Very-constrained wireless sensing devices may be alleviated from costly security-related computations, as long as mechanisms are in place to enable its delegation.

- The employed security modes may be dynamic in the context of a given application, that is, security may adapt and reconfigure as necessary during the lifetime of the application.

In the next section we discuss the usage of a system architecture that is in line with this security framework. This system architecture enables the employment of end-to-end transport-layer security in three usage modes, together with other mechanisms related with security and communications in the context of Internet-integrated WSN.

4 A system architecture supporting delegated end-to-end transport-layer security for Internet-integrated WSN

The system architecture discussed next provides the ground for the employment of mechanisms supporting the end-to-end security modes illustrated in Figure 1 (full DTLS security, delegated DTLS handshake and fully delegated DTLS security), with the help of mechanisms to support the delegation of DTLS authentication and an auxiliary LoWPAN authentication protocol, that we previously proposed in [12] and describe later in the paper.

4.1 A system architecture for flexible end-to-end security

The system architecture illustrated in Figure 2 is designed in accordance with the previously discussed framework, in particular for the support of transport-layer DTLS [6] security as required to protected CoAP [5] application-layer communications, in various usage modes. The following concepts are materialized by this architecture, in line with our security framework:

- End-to-end security for transport-layer communications is supported in the three usage modes illustrated in Figure 1. The cipher suites illustrated in Figure 2 correspond to the security mode that supports delegation of DTLS authentication

and key agreement, which we discuss later in the paper. The decision on the end-to-end security mode to employ may be supported by a 6LoWPAN border router (6LBR).

- Applications may employ multiple LoWPAN domains, as Figure 2 illustrates. For each domain the communications with another WSN or with the outside (Internet) is supported by a 6LBR. Communications between different 6LBR and with Access Control (AC) servers are assumed to employ a communications backbone without the constraints of LoWPAN domains.

- Each AC server may store information related with each LoWPAN constrained device. Access control information may also include rules to control accesses to CoAP resources on LoWPAN devices. The AC server also supports the LoWPAN authentication protocol, and the configuration of trust relationships between AC servers on different LoWPAN domains, as may be required for the support of mobility.

In Figure 2 we also illustrate the employment of a Certification Authority (CA) server, required to support the usage of public-keys and certification information in the context of a given sensing application.

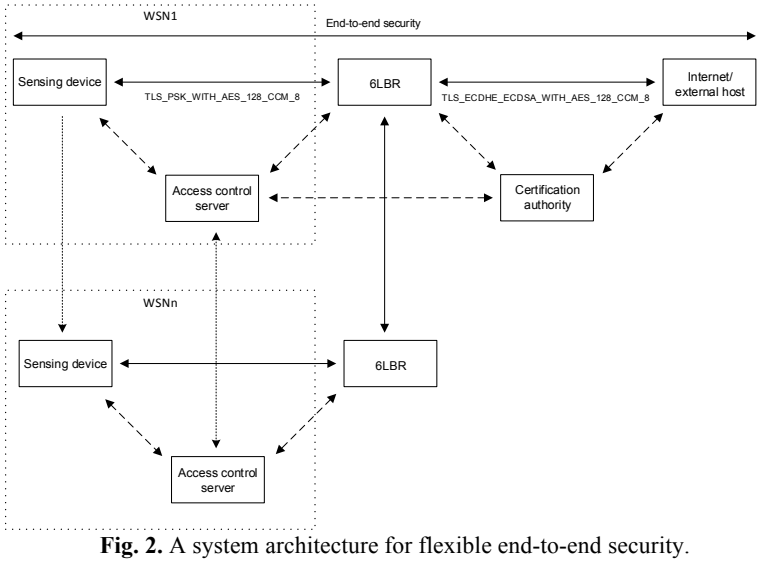


Fig. 2. A system architecture for flexible end-to-end security.

4.2 Delegated ECC public-key mutual authentication for transport-layer security

Security as currently proposed for CoAP requires the support of DTLS [6] encryption and of ECC public-key cryptography. On the one side, DTLS encryption and decryption may be efficiently supported using AES/CCM at the hardware in sensing

platforms implementing IEEE 802.15.4 [2], such as the TelosB. On the other, ECC cryptography can represent a bottleneck for some classes of constrained sensing platforms, in particular in supporting the Elliptic Curve Digital Signature Algorithm (ECDSA) [13] for authentication and the Elliptic Curve Diffie–Hellman with Ephemeral Keying Algorithm (ECDHE) [13] for key agreement. With this in mind, one possible approach consists in enabling the delegation of ECC public-key computations to the 6LBR gateway, a device we assume without the constraints of LoWPAN devices. As illustrated in Figure 1, two other end-to-end security modes may be supported for other classes of devices, depending on the application.

The initial DTLS handshake involves the usage of ECC cryptography, and requires more effort from the server than from the client. Since many IoT sensing applications will employ constrained CoAP sensing devices, DTLS handshake delegation also enables the usage of CoAP security with constrained sensing devices. Side-by-side with the support of DTLS authentication and key negotiation, this approach may also enable the activation on the 6LBR of security mechanisms protecting LoWPAN domains from Internet-originated threats.

Figure 3 illustrates the operation of the proposed mutual and delegated DTLS handshake supported by the 6LBR. It is important to note that, despite the support of the initial handshake by the 6LBR, the illustrated mechanisms guarantee full compatibility with DTLS as currently standardized and adopted for CoAP. This implies in practice that, from the point of view of both ends of the communication session, the interception and mediation of the handshake by the 6LBR is completely transparent. In parallel, after the completion of the initial handshake, we also guarantee that both ends of the communication session share a pre-master shared secret. From this pre-master shared secret both devices derive a shared master secret, and from this the keying material for the security session [6].

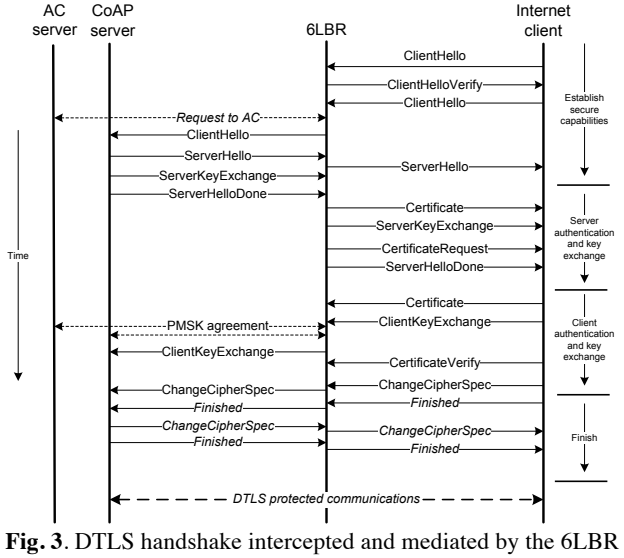


Fig. 3. DTLS handshake intercepted and mediated by the 6LBR

As previously illustrated in Figure 2, two separate cipher suites are employed in the context of the delegated handshake, one supporting secure communications on the LoWPAN domain, and the other on the Internet side. From the point of view of the Internet host, the security session supports the *Certificates* CoAP security mode using TLS_ECDHE_ECDSA_WITH_AES_128_CCM_8. This is the most useful security mode regarding the deployment of future IoT applications integrated with an established Internet security infrastructure. On the LoWPAN domain, security is supported by the *PreSharedKey* security mode with the cipher suite TLS_PSK_WITH_AES_128_CCM_8. While both cipher suites at the end depend on the usage of AES/CCM encryption, in reality they represent a very different impact on the resources required from constrained sensing platforms. The later mode and cipher suite is based on pre-shared key authentication, not requiring the usage of costly ECC public-key computations.

As Figure 3 illustrates, DTLS messages are transparently intercepted and forwarded by the 6LBR, as required for the support of the authentication and key negotiation handshake. One major goal of this handshake is to enable the two communications parties to agree on the cipher suite and on the encryption keys to employ in protecting communications. Encryption keys are obtained from a master key that the client and server must share [6] and this master key is obtained from a pair of client and server random values, together with a pre-master secret key. The handshake transports the client and server random values, while how the pre-master shared key is obtained depends on the authentication procedure. With public-key authentication (TLS_ECDHE_ECDSA_WITH_AES_128_CCM_8) the client may generate and send the pre-master shared key, encrypted with the server's public-key. On the other end, with pre-shared key based authentication (TLS_PSK_WITH_AES_128_CCM_8) this is in principle not possible, as at the initial stage the two communication entities do not share a secret required to protect communications. One important aspect of the proposed mechanism is that, as this single limitation would prevent end-to-end agreement of the pre-master secret key, we modify TLS_PSK_WITH_AES_128_CCM_8 to enable the forwarding on the 6LBR of the pre-master secret to the CoAP sensing device. In order not to compromise end-to-end security by accepting lower security on the LoWPAN side of communications, we introduce an auxiliary authentication protocol described later in the paper.

The *ClientHello* message confirming the initial request transports the client random value, the protocol version and the list of cipher suites supported by the client [6]. After this message the 6LBR requests, from the AC server, security-related information concerning the destination CoAP device. Such information is obtained in the context of the LoWPAN authentication protocol, and includes the supported cipher suites, the X.509 certificate of the device and a description of its capabilities. As previously discussed, this information may be used as input in the process of computing the appropriate end-to-end security mode for this communications session. As illustrated in Figure 3, in the next message flight the 6LBR authenticates the CoAP server on its behalf, while also requesting for the client to authenticate using its certificate. The *ClientKeyExchange* message contains the pre-master secret key generated by the client, which the 6LBR forwards to the CoAP sensing device. The

same message transports the client random value, while the server random value is transported in the *ServerKeyExchange* message.

It is important to note that this approach also enables the employment of other cipher suites and delegation approaches, as appropriate for different types of sensing devices and as long as compatibility is guaranteed for the pair of ciphers employed. Very-constrained devices may require the full delegation of all DTLS security operations to the 6LBR, while on the other end more powerful devices may fully support DTLS. In all situations, the 6LBR is able to learn the pre-master secret key and random values for a given DTLS security session, thus enabling the computation of the final master key and derivation of the keying material. This may provide the ground for the employment of other security mechanisms, for example those involving the filtering at the application-layer of encrypted CoAP message exchanges.

4.3 LoWPAN authentication and PMSK key exchange

Before pre-master secret key (PMSK) agreement between the 6LBR and the CoAP device, both entities authenticate each other using the LoWPAN authentication protocol illustrated in Figure 4. TLS_PSK_WITH_AES_128_CCM_8 is modified to support the exchange of the pre-master secret key in the context of the handshake, and the authentication protocol enables the maintenance of high end-to-end security during the critical initial step of authentication and key agreement. The authentication protocol is based on Kerberos [14], while with modifications to support its integration with the two-phase delegated DTLS handshake and the transportation of the pre-master secret key. In the context of the system architecture in Figure 2, we assume that the AC server maintains a secret key shared with each entity in the network. In addition to this key, the AC server also stores the entity identification (ID), its X.509 ECC certificate and the list of supported ciphers and compression methods. The ID for a CoAP device may be its LoWPAN IPv6 link-local address.

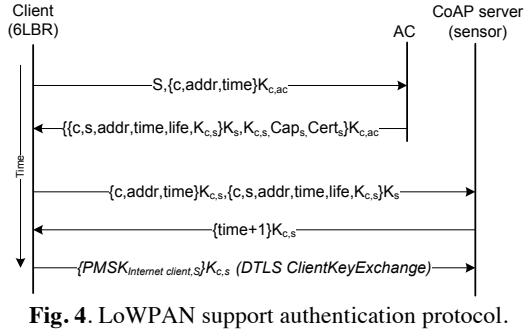


Fig. 4. LoWPAN support authentication protocol.

The secret key $K_{c,ac}$ is shared between the 6LBR and the AC server and is employed to encrypt the first message flight of the authentication protocol, as Figure 4 illustrates. This allows the 6LBR to obtain the security-related information for the destination CoAP device. The 6LBR returns an authentication token encrypted with the secret key K_s of the destination device, which the 6LBR forwards to the CoAP

device as a proof of authorization to access the CoAP server. It also returns a key $K_{c,s}$ to protect subsequent messages exchanged between the 6LBR and the CoAP sensing device. Finally, this reply also contains the capabilities of the destination device and its certificate.

The second message flight in Figure 4 supports mutual authentication between the 6LBR and CoAP sensing device, and finally the secure exchange of the pre-master secret key. The CoAP server compares the information contained in the token generated by the 6LBR against the information included in the authenticator generated by the AC server, in order to authenticate the 6LBR. The timestamp and lifetime values offer protection against message replay attacks. The following reply message is encrypted with this key and authenticates the CoAP server to the 6LBR, by having the server transmit the received timestamp plus one.

5 Experimental evaluation of end-to-end security

The mechanisms previously discussed may provide a contribution to the security of Internet-integrated LoWPANs and to the intelligent allocation of limited resources of CoAP sensing devices to security. We proceed with a description of a partial experimental evaluation of the previously discussed security mechanisms.

5.1 Application security and functional parameters

We address the experimental evaluation of the previously discussed mechanisms from the point of view of its impact on energy using a TelosB [1] sensing platform. Other sensing platforms may be employed in the future for experimental evaluation purposes, which may provide important feedback on the most appropriate end-to-end security mode for alternative classes of devices. Regarding the definition of appropriate functional and security profiles, we consider two types of applications:

- Applications requiring a moderate number of DTLS sessions per hour, also with a moderate number of CoAP requests per DTLS session. For experimental evaluation purposes we consider from 1 to 400 DTLS sessions per hour with 2 CoAP requests per DTLS session.

- Applications requiring a higher number of DTLS sessions per hour, also with a higher number of CoAP requests per DTLS session. For experimental evaluation purposes we consider from 14 to 7200 DTLS sessions per hour with 10 CoAP requests per DTLS session.

Please note that a CoAP request involves two messages, one containing the request sent to the server and (at least) other containing the corresponding reply. We are also interested in evaluating two end-to-end security modes, one with full end-to-end DTLS security supported by the sensing device, and the other with the proposed DTLS handshake plus the LoWPAN authentication protocol.

5.2 Impact on the lifetime of Internet-integrated sensing applications

Our experimental evaluation study employs a TelosB [1] and the Linux operating system, with the TelosB supporting the TinyOS [15] operating system containing the Berkeley Low-IP (BLIP) 6LoWPAN stack, CoAP and DTLS. We also use standalone AES/CCM encryption [16] and ECC using code based on TinyECC [17]. The 6LBR, the CA server, the AC server and the Internet CoAP client are supported using Linux. The CoAP client on the Internet hosts uses *libcoap* [18] with DTLS. The TelosB and the AC server support the LoWPAN authentication protocol.

Energy required to support end-to-end security was obtained using experimental measurements of the voltage across a current resistor placed in series with the battery pack of the TelosB. We measure the energy required to support the DTLS handshake and the energy required to support DTLS encryption using AES/CCM. For all measurements we consider the usage of 6LoWPAN 102-byte messages. From the experimental measurements we may derive expected lifetime values for sensing applications of the two types previously discussed, which we illustrate in Figures 5 and 6. We consider the usage of a TelosB powered using two new AA LR-6 batteries.

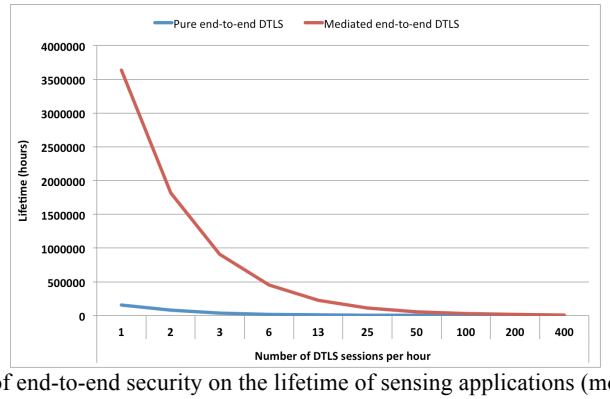


Fig. 5. Impact of end-to-end security on the lifetime of sensing applications (moderate usage of DTLS security and of CoAP communications).

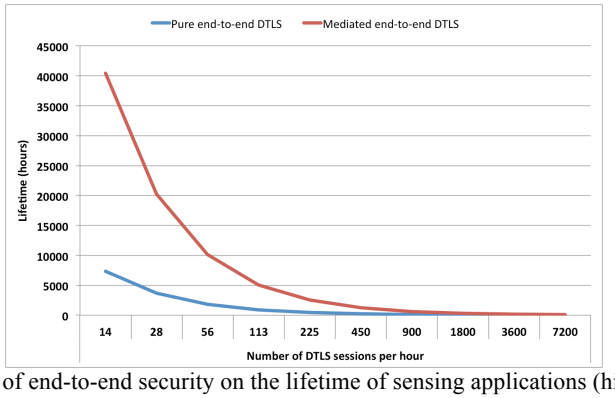


Fig. 6. Impact of end-to-end security on the lifetime of sensing applications (higher usage of DTLS security and of CoAP communications).

For both usage scenarios we may observe a clear advantage of the proposed delegated DTLS handshake, particularly for a lower number of DTLS sessions per hour. The illustrated values already consider the energy required to support the LoWPAN authentication protocol. This advantage in less expressive for a higher number of DTLS sessions, due to the higher impact of AES/CCM encryption in comparison with the impact of the DTLS handshake. Considering that many IoT applications will probably require low or moderate transmission rates, the proposed mechanisms prove to be effective for devices with the characteristics of the TelosB.

7 Conclusions

Many of the currently envisioned IoT sensing applications may require or at least benefit from the usage of end-to-end standard Internet communications between constrained sensing devices and Internet hosts or external backend servers. We address the design of a framework for quantifiable and controllable end-to-end security in the context of Internet-integrated LoWPAN environments. We also propose a system architecture in line with this framework, in the form of a system architecture providing support for the usage of expensive ECC public-key authentication and key negotiation with constrained wireless sensing platforms.

Despite our preliminary evaluation of the proposed system architecture, other challenges remain to be addressed in its context. Mechanisms may be designed to support end-to-end security in the presence of mobile (roaming) devices. One may also design different end-to-end security approaches, or new techniques to decide on the security mode in the presence of particular sensing platforms or application profiles. The LoWPAN authentication protocol may also provide the ground for the employment of different security approaches on the LoWPAN. For example, one may employ AES/CCM supporting integrity only (by using only CBC-MAC) or encryption mechanisms better appropriate to sensing platforms that do not support AES/CCM at the hardware.

Other aspect that may be addressed in the context of the proposed system architecture is the mobility of LoWPAN sensing devices, as Figure 2 illustrates. If different IPv6 prefixes are employed in the origin and destination WSN domains, a change of address may take place. In this context, mechanisms may be designed to guarantee the transparency of mobility from the point of view of end-to-end transport-layer security, so that a device moving between different LoWPAN domains is able to continue using previously negotiated security sessions and its associated keying material.

Acknowledgements. The work presented in this paper was partly financed by the iCIS project (CENTRO-07-ST24-FEDER-002003), which is co-financed by QREN, in the scope of the Mais Centro Program and European Union's FEDER.

References

1. TelosB Mote Platform, http://www.xbow.com/pdf/Telos_PR.pdf.
2. Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs), IEEE std. 802.15.4, 2006.
3. IPv6 over Low power WPAN (6lowpan), https://datatracker.ietf.org/wg/6lowpan/charter/.
4. Constrained RESTful Environments (core), https://datatracker.ietf.org/wg/core/charter/.
5. Shelby Z. et al. Constrained Application Protocol (CoAP), draft-ietf-core-coap-13, 2013.
6. Rescorla E et al. Datagram Transport Layer Security Version 1.2, *RFC 6347*, 2012.
7. Chen Xiangqian, K. Makki, Yen Kang and N. Pissinou, "Sensor network security: a survey," *IEEE Communications Surveys & Tutorials*, vol. 11, no. 2, 2nd Quarter 2009, pp.52-73.
8. Gupta V et al. Sizzle: a standards-based end-to-end security architecture for the embedded Internet. *Proceedings of the 3rd IEEE International Conference on Pervasive Computing and Communications (PerCom 2005)*, Kauai Island, HI, USA, 2005. DOI: 10.1109/PERCOM.2005.41.
9. Jung et al. SSL-based Lightweight Security of IP-based Wireless Sensor Networks. *Proceedings of the International Conference on Advanced Information Networking and Applications Workshop (WAINA '09)*, Bradford, UK, 2009.
10. Raza S et al. 6LoWPAN Compressed DTLS for CoAP. *Proceedings of the IEEE 8th International Conference on Distributed Computing in Sensor Systems (DCOSS 2012),* Hangzhou, China, 2012. DOI: 10.1109/DCOSS.2012.55.
11. Kothmayr T et al. A DTLS Based End-To-End Security Architecture for the Internet of Things with Two-Way Authentication. *Proceedings of the Seventh IEEE International Workshop on Practical Issues in Building Sensor Network Applications (IEEE SenseApp 2012),* Clearwater, FL, USA, 2012.
12. Granjal J et al. End-to-end transport-layer security for Internet-integrated sensing applications with mutual and delegated ECC public-key authentication. *Proceedings of The IFIP Networking 2013 Conference*, New York, USA, 2013 (to appear).
13. SECG-Elliptic Curve Cryptography-SEC 1, http://www.secg.org (accessed Apr 2013).
14. Neuman B, Ts'o T. Kerberos: an authentication service for computer networks. *IEEE Communications Magazine,* 1994, 32(9), pp. 33-38, DOI: 10.1109/35.312841.
15. TinyOS Operating System, http://www.tinyos.net/ (accessed Apr 2013).
16. Standalone hardware AES Encryption using CC2420, http://cis.sjtu.edu.cn/index.php/The_Standalone_AES_Encryption_of_CC2420_(TinyOS_2.10_and_MICAz (accessed Apr 2013).
17. Liu A., Ning P. TinyECC: A Configurable Library for Elliptic Curve Cryptography in Wireless Sensor Networks. Proceedings of the 7th h international conference on Information processing in sensor networks (IPSN '08), 2008.
18. LibCoAP, http://sourceforge.net/projects/libcoap/ (accessed Apr 2013).

Middleware Group Communication Mechanisms in M2M environments

André Riker, Jorge Granjal, Marilia Curado, and Edmundo Monteiro

University of Coimbra, Coimbra, Portugal
ariker@dei.uc.pt, jgranjal@dei.uc.pt, marilia@dei.uc.pt, edmundo@dei.uc.pt

Abstract. Machine-to-Machine (M2M) communication is a technology that will bring new horizons for the current concept of smart systems. However, efficient M2M communication requires the design of middleware/platform components able to deal with multiple application requirements and heterogeneous wireless environments. In order to address this challenge, this paper proposes the Communication Manager Component (CMC) to integrate the M2M middleware. CMC enables the management of communication mechanisms, such as data-aggregation, sleep-schedule, uplink-schedule and signaling-aggregation, aiming to save energy and to satisfy multiple application data requests. The management is performed dynamically taking into account the applications requests, the base-station overload indicators and the M2M devices' status (e.g. energy level, location).

Keywords: Machine-to-Machine; M2M; Middleware; Data-aggregation; Sleep-schedule; Uplink-schedule; Signaling-aggregation

1 Introduction

Machine-to-Machine (M2M) communication is characterized as the autonomous information exchange between electronic devices. Although M2M communication involves any number of devices and unrestricted network technologies, a special attention has been given for M2M communication of a massive number of devices using wireless technologies. Generally, large part of the M2M devices are resource-constrained in terms of memory, Central Processing Unit (CPU) and battery, and communication is performed via mobile and capillary wireless networks.

M2M has emerged as the technology able to remotely control devices, forming a new era of smart applications and enabling new forms of services/applications (e.g. smart systems for transportation, utilities meter, surveillance and healthcare). The exploration of these M2M services has caught the attention of organizations, government and industries, since every entity in the global business wants to maximize their profits by providing better services, reducing cost and faults.

There are many M2M applications being implemented and deployed. However, the current M2M solutions require from the companies the development of

comprehensive M2M solutions. For small and middle scale systems, customized solutions could have satisfactory performance. However, by extending the M2M applications for large scenarios, involving millions of devices, it is clear the need of an efficient M2M middleware/platform. The M2M middleware is defined as a service platform that enables different M2M applications to share a set of common functionalities and enables the service providers to reuse the essential functionalities of M2M communication without the need to design a complete communication architecture.

In M2M communication some problems can occur. First, the 3GPP study about M2M communications [1] shows that in M2M communication a huge number of M2M devices may try to connect at the same point in time. If this occurs, the mobile base-station will be overloaded and the communication with all requested devices will not be possible, damaging the M2M communication as well as the traditional Human-to-Human (H2H) communication. Besides the overload due to the amount of connection and signaling messages, the number of traffic sessions and the number of attached devices also can cause the base-station overload. Second, in several scenarios, M2M communication involves resource-constrained devices, which have low power resources. Without an appropriate use of the devices' energy, these devices will need human maintenance, which increases operational costs and reduce the network lifetime. Third, the M2M communication involves applications with a high level of heterogeneity in terms of amount of traffic, frequency of transmissions and delay tolerance. For example, some M2M applications require near real-time communication (e.g. tracking of objects) and other applications are delay tolerant (e.g. smart metering). The problem emerges from the fact that most of these heterogeneous applications must be able to share data. Thus, without an adequate data management, the heterogeneous M2M applications will be detached and they will not be aware about the events detected by other applications.

Knowing these problems and the M2M characteristics, a solution that aims to address these problems together should fulfill (at least) the following requirements: (i) prolong the lifetime of the constrained-resource devices; (ii) avoid the base-station overload; (iii) reduce the application programming complexity, allowing the applications to express their data interests in a high level of abstraction; (iv) manage multiple application interests; and (v) be adapted dynamically according to the level of resources available in the devices involved in the communication.

Four communication mechanisms, namely data-aggregation, sleep-schedule, uplink-schedule and signaling-aggregation, show great potential to solve the problems and to satisfy some of the mentioned requirements. Data-aggregation and sleep-schedule are well-known techniques used to extend to network lifetime. Data-aggregation mechanisms process the data gathered from the network to reduce the amount of spatio-temporal data redundancies. On the other hand, the sleep-schedule mechanism aims to keep the devices as much time in sleep mode as possible to save the devices energy. In addition, the uplink-schedule and the signaling-aggregation mechanisms, proposed in [2] and [3], aim to re-

duce the base-station overload. The uplink-schedule mechanism enables a negotiation between the M2M devices and the base-station to schedule the next transmission time. Performing this scheduling, the base-station provides priority transmissions for devices with lower delay tolerance. The signaling-aggregation mechanism calculates the time interval in which the base-station waits for similar signaling messages from the M2M devices, supporting the aggregation of similar signaling messages.

In addition, a solution involving these four mechanisms should consider how each mechanism affects the others. Firstly, we consider only the data-aggregation and the sleep-schedule mechanisms. In the data-aggregation schemes, the aggregation device must wait to receive data from the neighboring nodes. In the sleep-schedule schemes each device must define the time intervals in which it will be in active mode. If the aggregation device waits too much, the aggregation rate will be high, but it will damage the application requirements. On the other hand, if the sleep-schedule defines a long period of sleep-mode, the savings of energy will be high, but it will affect the application requirements. It is clear that both mechanisms, separately, need to avoid excessive delay and maintain the low energy consumption. However, it is also necessary that both mechanisms must be designed together, considering how the data-aggregation delay affects the sleep-schedule delay and vice-versa.

Besides the data-aggregation and the sleep-schedule integration problem, also there is a synchronization problem involving the sleep-schedule, uplink-schedule and signaling-aggregation mechanisms. The uplink-schedule and the signaling-aggregation mechanisms must be aware of the time that the device will be in active mode, otherwise the uplink-schedule could erroneously determine the next time transmission. Without synchronization the next transmission could be scheduled to a period that the device is in sleep mode, which means that no transmission will occur and no signaling message will be send. Therefore, only with the integrated design of these four mechanisms the M2M communication will achieve high performance, since all these four mechanisms provide performance gain to the network, but without the integration and the dynamic management of these mechanisms, the M2M communication performance is negatively affected.

In this paper, we propose the architecture of the Communication Manager Component (CMC) as part of the M2M middleware that dynamically manages the four communication mechanisms according to the applications requirements, the base-station overload conditions and the M2M devices resources.

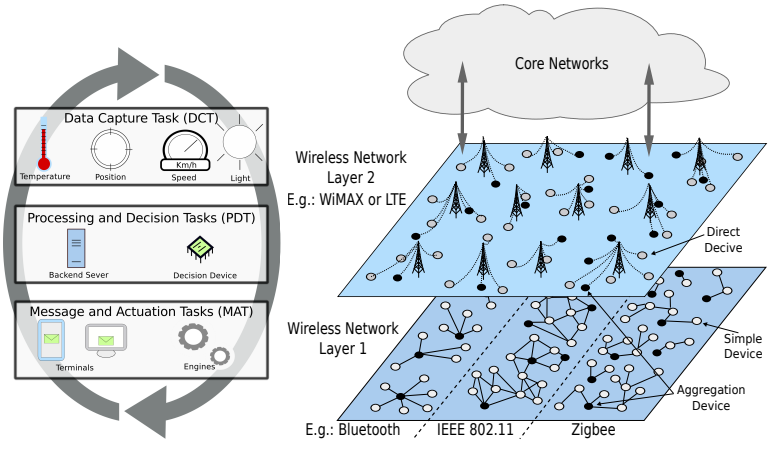
The remainder of this paper is organized as follows. Section 2 gives an overview about M2M communication and middleware. Section 3 describes the related work. Then, Section 4 describes the CMC architecture. Finally, Section 5 presents some concluding remarks and future works.

2 M2M background

In this Section we begin showing the main concepts and the network architecture of the M2M communication. Then, we show how the middleware integrates the M2M system.

2.1 M2M communication overview

To comprehend the M2M communication, Fig. 1.a shows the continuously workflow performed by the M2M system. Firstly, the machines perform the Data Capture Task (DCT), which is the data acquisition from the sensed environment. The electronic sensors conduct the DCT (e.g. temperature, humidity and flow measurement). Then, the machines perform the Processing and Decision Task (PDT), which requires computational power capabilities to manage the data received, and support decision-making functionalities. Finally, some devices perform the Message and Actuation Tasks (MAT), comprising the messages delivery and actions' execution (e.g. alerts/information, or commands to actuators, or relevant events).



a) M2M workflow b) M2M architecture

Fig. 1. M2M communication

Many communication technologies can be deployed to enable the data-flow between the machines that execute DCT, PDT and MAT functionalities. One of the most appropriate network architectures for M2M communication is the heterogeneous wireless architecture. Using this network architecture, the connections could be via short, local or wide wireless technologies, depending on the application requirements and according to the machines resources. In this direction, some of the recent research ([4], [5], [6]) has driven to a Heterogeneous Hierarchical Architecture (HHA).

HHA aims to alleviate the costs, reducing the complexity of nodes. To achieve that, HHA deploys Simple Devices (SD), which are nodes designed as simple as possible, generally, equipped with short wireless technology (see Fig. 1.b). On the other hand, the HHA concentrates some of the vital and complex services in a reduced number of nodes, called Aggregation Devices (AD). The AD nodes perform the complex tasks, such as data-aggregation, Quality of Service (QoS) management, multimedia conversion and remote access. Besides, the AD nodes could be equipped with dual network cards (e.g. short/local and wide wireless cards), acting as Gateway to forward the SD data to the core network or/and to interface the communication between the short and the wide wireless network.

In addition, some M2M scenarios (e.g. vehicular and video surveillance) require nodes with a large bandwidth capacity. Therefore, in some cases Direct Devices (DD) can be deployed, which are nodes equipped with resources to access directly with the wireless technologies.

2.2 M2M middleware

The M2M middleware is the software component situated between the applications and the devices. Fig. 2 shows an overview of the integration of the M2M middleware within the M2M system.

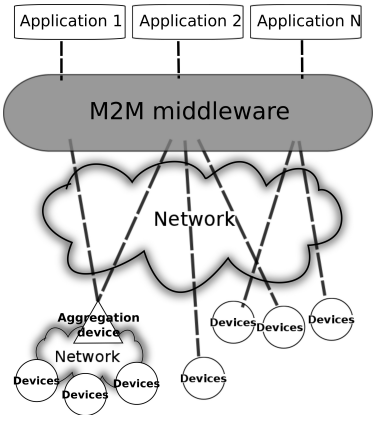


Fig. 2. The M2M middleware

The M2M middleware enables the M2M applications to access their devices using a common set of services, reducing the costs with programming and enabling the interaction of different applications from different stakeholders. For the future M2M communication it is essential the development of an efficient middleware able to support the interaction of multiple applications and to offer to them a common application infrastructure to access the communication mechanisms.

The M2M middleware receives from the applications the data requests and must answer each of these data requests with the appropriated data. To answer the data requests, the middleware can use different transmission technologies to collect the data from the devices. Besides, the M2M middleware will execute its functionalities over distributed devices (e.g. over servers, AD and DD). Then, with the M2M middleware, the development of M2M applications becomes less complex since the developer can use API's to access the middleware functionalities. The middleware approach hides from the applications the heterogeneity in terms of the communication (e.g. wired and wireless) and hardware (e.g. devices from different suppliers).

These middleware characteristics allow the applications to produce data requests with a high level of abstraction, which means that the applications are not aware of communication mechanisms, the network conditions and the devices resources involved in each specific data request. Then, the communication management is not performed at the applications level but by the M2M middleware. Therefore, a fundamental service for the future M2M communication is a management module able to deal with all these communication aspects.

3 Related work

A prominent sleep-schedule mechanism designed to M2M environments is presented in [7]. This solution considers the existence of multiple data types and devices with different sensing capabilities. This proposal defines a monitoring time for each sensing region. In each region, at least one node should transmit data to the M2M gateway during every monitoring time. This monitoring time is defined according to the data sensed. In every monitoring time, the devices have the options to transmit the data or to stay in sleep-mode. Then, this proposal schedules the sensed data transmission to save power while maintaining the gateway with freshness-sensed data.

The uplink-schedule proposed in [2] avoids that M2M devices send synchronized connection requests to the GSM base-station. In GSM networks, for a device to establish a connection, it is necessary to send to the base-station a random access burst via the Random Access Channel (RACH). After receiving the connection request, the base-station assigns a slot in the Access Grand Channel (AGCH). However, while there are approximately 217 RACH slots available per second, only 25 AGCH slots can be assigned per second. Therefore, to solve the AGCH bottleneck problem, the authors in [2] propose that in moments when the base-station is overloaded, the device and the base-station communicate in order to assign the time of the next AGCH slot. To assign an AGCH slot, the proposed mechanism uses the desired time, the report size and the delay tolerance.

The signaling-aggregation proposed in [3] avoids the signaling overload in LTE networks. Frequently, in LTE networks the devices send signaling messages to the base-station (e.g. Tracking Area Updating (TAU) request, massive attach/connect). The Base-station receives these messages and should take the

appropriate operation, which in several cases is to inform the Mobility Management Entity (MME) or the Serving GPRS Support Node (SGSN). If a large number of devices sent signaling messages, the MME and the SGSN can be overloaded. Therefore, the mechanism proposed in [3] aggregates the similar signaling messages received by the base-station and informs the MME or the SGSN in a single bulk message. Thus, this mechanism reduces the traffic between the base-station and the MME/SGSN. Moreover, it reduces the resources used to open, maintain and finish MME/SGSN connections.

However, the solutions presented in [7], [2] and [3] do not address the management aspects of these mechanisms, otherwise the management do not taking into account the existence of other communication mechanisms. Moreover, to reduce the scope, most of the mechanism proposals (e.g. data-aggregation, sleep-schedule, uplink-schedule and signaling-aggregation mechanisms) assume the existence of the necessary middleware support, but to support these mechanisms is not a trivial task.

A large number of middlewares proposed to Wireless Sensor Networks (WSN) [8–12] support high level of application abstraction. These middleware solutions reduce the application programming complexity and most of them support data-aggregation schemes to reduce the data volume. Data-aggregation support means that the applications can inform the middleware about the data-aggregation function (e.g. lossy/lossless aggregation, duplicate sensitive and mathematical functions) and the middleware takes the necessary decisions to delivery the requested data to the applications. However, the main drawbacks of these solutions are: (i) only a reduced number of middleware solutions, such as [13–15], support sleep-schedule mechanism. Moreover, even these solutions do not address the sleep-schedule integration with other mechanisms; (ii) except the middleware proposed in [16], none of the analyzed middleware solutions is designed to manage multiple application data requests. Generally the solutions consider a single application making data requests to the devices; and, (iii) the middleware solutions are not designed to be adjusted dynamically according to the devices' resources (e.g. the devices' mobility and energy level).

Therefore, according to the best of our knowledge, none of the middleware solutions consider the integration and the dynamic management of the four mechanisms satisfying multiple applications in a M2M environment.

4 Communication Manager Component Architecture

In this section, we propose the Communication Manager Component (CMC) solution, which integrates the M2M middleware and aims to dynamically adapt a set of communication mechanisms to satisfy multiple M2M applications according to the devices' resources involved and the MCN overload level. Therefore, we start this section showing an overview of the component architecture proposed. Then, we describe the input parameters of the proposed component as well as the Requests and Description Modules. Finally, we show the configuration profiles of the communication mechanisms and the Group Communication Module.

4.1 Overview

As shows Fig. 4.1, the CMC architecture is composed by the Requests and Description Modules and the Group Communication Module. The Requests and Description Modules processes the input data coming from the devices and the applications, and select a temporary set of devices that could participate in the communication.

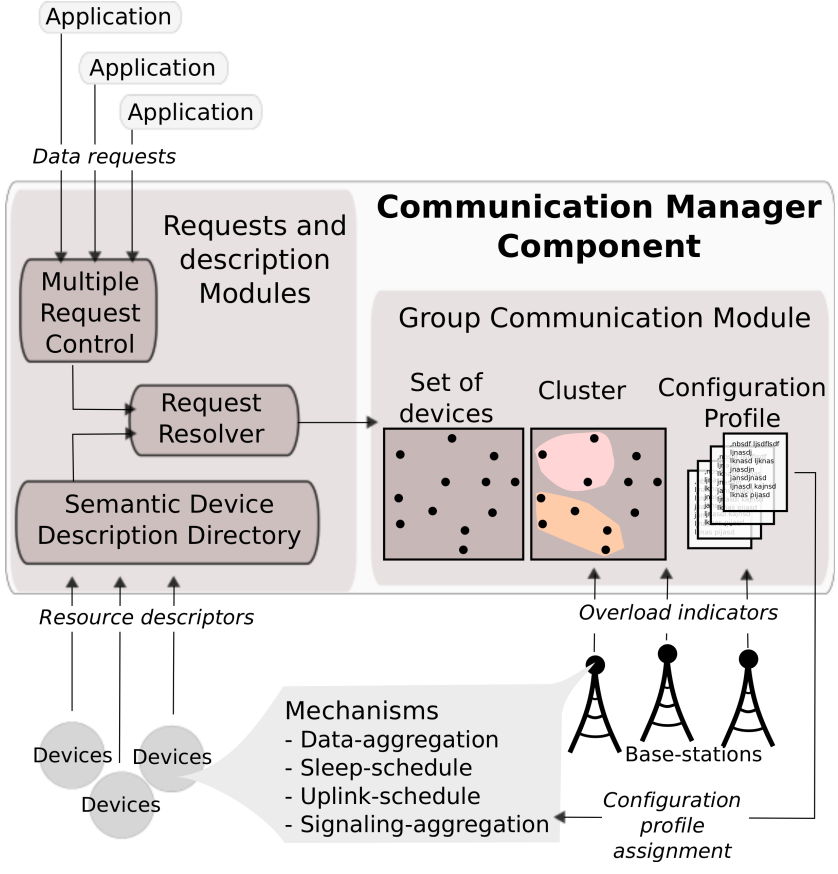


Fig. 3. CMC overview

On the other hand, the Group Communication Module (GCM) receives the temporary set of devices (provided by the Request and Description Modules) and the overload indicators (from the base-stations), and finally performs the core tasks of the CMC, which is the selection of a definitive set of devices, the clusterization of these devices and the assignment of a configuration profile for each

cluster. The configuration profiles define the behavior of the four communication mechanisms.

4.2 Input parameters

The selected CMC input parameters are (i) the applications requests, (ii) the semantic resource description of the devices and (iii) the current MCN overload level.

Due to the fact that application requests are expressed in high level of abstraction, the applications do not deal with some of the network complexities, such as the IP addresses of the target group of devices and the type of network technology involved in the communication. In general, the application request denotes the application interests by specifying the data type, location, delay tolerance, level of data accuracy and the desired aggregation function (e.g. the average, min/max). In addition, these application requests could result on a single data transmission or on multiple data communication (e.g. periodical measurements) and it could involve single or multiple devices, but a special attention is given for communication of multiple devices, since data-aggregation is more relevant in this scenario.

The semantic description reveals key characteristics of the devices, such as the type of data sensed, the node position (when possible), the mobility pattern, the network interface(s), the energy level, the CPU and memory capacity. These characteristics are important information for the CMC decision. However, to maintain the consistency of some of these dynamic characteristics it is necessary to periodically access the devices and this access must be executed avoiding the excessive network overhead.

Finally, the management scheme will also receive as input the level of overload from the base-stations involved in the communication. This information gives the overall base-station capacity in terms traffic load and the number of sessions supported.

4.3 Requests and description modules

The multiple application requests received by the M2M middleware must be tested in order to verify the compatibility level between the received request and the active older ones. This functionality is performed by the Multiple Request Control (MRC) module. In the case the new request does not present related requests, it means that the new request does not have any conflict with other active requests. Otherwise, it is necessary to know the data request tolerance ranges (e.g. data accuracy and delay tolerance delay range) and verify the possibility to resolve the conflict.

Another module is the Semantic Device Description Directory (S3D). This mechanism aims to maintain the consistency of information about the M2M devices. As mentioned before, the devices information is related to static hardware characteristics (e.g. CPU, memory and network interface) and dynamic status (e.g. location, energy level and mobility pattern).

Finally, the Request Resolver (RR) module is designed to receive a data request and filter the S3D, returning a temporary set of nodes able to satisfy the data request. However, this set of devices is non-definitive, since it can be modified in case of existence of conflicting data requests or data accuracy level that do not necessitate the data collection from the whole set of devices.

4.4 Configuration profiles of the communication mechanisms

The uplink-schedule and the signaling-aggregation mechanisms actuate in the base-stations, which can have hundreds of devices attached. On other hand, the data-aggregation and the sleep-schedule mechanism actuate in the M2M devices. Therefore, in a hierarchical point of view, the uplink-schedule and the signaling-aggregation are at a higher level of actuation than the data-aggregation and the sleep-mechanism.

The data-aggregation and the sleep-schedule profiles can be configured in terms of energy consumption and communication delay. Considering these two parameters, several profiles can be composed. For example, in some situations could be a good strategy to configure the data-aggregation to have low energy consumption and high delay. Simultaneously, the sleep-schedule can be configured to generate high energy consumption and low delay. By doing this, the sleep-schedule and the data-aggregation will balance the energy consumption and the communication delay. In other situations, it could be a good strategy to configure both mechanisms to have low energy consumption and high communication delay.

According to the profile assumed by the data-aggregation and sleep-schedule, the base-station must synchronize the signaling-aggregation and the uplink-schedule mechanisms. This synchronization means update the signaling-aggregation with a new wait time and the uplink-schedule with a new priority assignment scheme.

4.5 Group communication Module

The CMC performs two decisions. The first is the definition of the definitive set of devices and the second is the selection of the configuration profile that will be assumed by the communication mechanisms. Both decisions are based on the answers provided by the RR module, the base-stations overload indicators and the data request specifications.

The task to define the definitive set of devices starts with the initial set of devices provided by the RR module. This set could be modified (delete and/or include devices) according to data-accuracy tolerance specified in the data request. For example, consider a M2M application aims to measure and control the city water consumption. The application interest is expressed by a data request, which orders the measurement of the water consumption in a particular city region with a data accuracy of 90%. Initially, the RR returns the temporary set of devices that match with the request specifications. As the data request has a data accuracy tolerance of 10%, it means that 10% of devices are not mandatory

to provide data, since this data is only for estimation purposes. Therefore, 10% of devices can be deleted from the temporary set. Several metrics can be used to select the devices that must be deleted (e.g. delete the devices with lower energy level or delete devices attached to overload base-stations).

After the definition of devices that will participate in the communication, the second decision task is to cluster the devices and assign a configuration profile to each cluster. A cluster of devices could involve multiple aggregation devices and direct devices (See Fig. 1.b). The selected configuration profile will regulate the data-aggregation and the sleep-schedule mechanisms in order to satisfy the application delay and minimize the energy consumption. Moreover, the signaling-aggregation and the uplink-schedule will be updated in order to reduce the traffic overhead and the communication priority, respectively.

5 Conclusion and future works

In the M2M era, new forms of communication are possible and new services and applications will be available. By exploring the extensive number of services and the wide range of scenarios, an immense market potential has emerged for the M2M networks, including transportation, utilities, security, retails services and healthcare. However, to provide efficient group communication in M2M environments is a challenge because it requires the design of heterogeneous network technologies, as well as new mechanisms for efficient communication involving multiple applications. Besides, the M2M middleware should be prepared to deal with the multiple M2M applications and to manage the communication mechanisms to achieve high network performance, which includes saving energy and satisfying the application requirements.

The manager component proposed in this paper allows multiple M2M applications to use efficiently the network resources according to the applications' needs, the network overload level and the device resources. This component has importance in a scenario of massive devices like the M2M environment, since to date, there are in the world around five billion of M2M devices connected to mobile networks [17] and the M2M communication will increase this number to 50 billion by the end of this decade.

Some aspects of the proposed middleware component are under definition. The priority for future works is to design of the configuration profiles, strategies and select metrics as well as rules/polices for the decision tasks. Another aspect that will be studied is the overhead impact of the proposed component.

Acknowledgment

This work was partially funded by the Fundação para a Ciência e a Tecnologia (FCT, Portugal) by CMUPT /RNQ/0015/2009 MORFEU; iCIS project (CENTRO-07-ST24-FEDER-002003), co-financed by QREN, in the scope of the Mais Centro Program; and CAPES and CNPq (Brazil) through of the Ciência sem Fronteiras Program/2013.

References

1. 3GPP: System improvements for machine-type communications. **V0.5.1.** (July 2010) TR 23.888
2. Lioumpas, A.S., Alexiou, A.: Uplink scheduling for machine-to-machine communications in lte-based cellular systems. In: GLOBECOM Workshops (GC Wkshps), 2011 IEEE, IEEE (2011) 353–357
3. Taleb, T., Kunz, A.: Machine type communications in 3gpp networks: potential, challenges, and solutions. Communications Magazine, IEEE **50**(3) (2012) 178–184
4. Zhang, J., Shan, L., Hu, H., Yang, Y.: Mobile cellular networks and wireless sensor networks: toward convergence. Communications Magazine, IEEE **50**(3) (march 2012) 164 –169
5. Igarashi, Y., Ueno, M., Fujisaki, T.: Proposed node and network models for an m2m internet. In: World Telecommunications Congress (WTC), 2012. (march 2012) 1 –6
6. Wu, G., Talwar, S., Johnsson, K., Himayat, N., Johnson, K.: M2m: From mobile to embedded internet. Communications Magazine, IEEE **49**(4) (april 2011) 36 –43
7. Fu, H.L., Chen, H.C., Lin, P., Fang, Y.: Energy-efficient reporting mechanisms for multi-type real-time monitoring in machine-to-machine communications networks. In: INFOCOM, 2012 Proceedings IEEE, IEEE (2012) 136–144
8. Kumar, M., Schwiebert, L., Brockmeyer, M.: Efficient data aggregation middleware for wireless sensor networks. In: Mobile Ad-hoc and Sensor Systems, 2004 IEEE International Conference on. (2004) 579–581
9. Manjhi, A., Nath, S., Gibbons, P.B.: Tributaries and deltas: efficient and robust aggregation in sensor network streams. In: Proceedings of the 2005 ACM SIGMOD international conference on Management of data, ACM (2005) 287–298
10. Nath, S., Gibbons, P.B., Seshan, S., Anderson, Z.R.: Synopsis diffusion for robust aggregation in sensor networks. In: Proceedings of the 2nd international conference on Embedded networked sensor systems, ACM (2004) 250–262
11. Lindsey, S., Raghavendra, C., Sivalingam, K.: Data gathering algorithms in sensor networks using energy metrics. Parallel and Distributed Systems, IEEE Transactions on **13**(9) (2002) 924–935
12. Madden, S., Franklin, M.J., Hellerstein, J.M., Hong, W.: Tag: A tiny aggregation service for ad-hoc sensor networks. ACM SIGOPS Operating Systems Review **36**(SI) (2002) 131–146
13. Schiele, G., Becker, C.: Experiences in designing an energy-aware middleware for pervasive computing. In: Pervasive Computing and Communications, 2008. PerCom 2008. Sixth Annual IEEE International Conference on. (2008) 504–508
14. Vasanthi, N.A., Annadurai, S.: Sleep schedule for fast and efficient control of parameters in wireless sensor-actor networks. In: Communication System Software and Middleware, 2006. Comsware 2006. First International Conference on. (2006) 1–6
15. Xiong, F., Bai, L.: Interoperable wireless sensor network model using multi-agent-based middleware. In: Intelligent Signal Processing and Communication Systems (ISPACS), 2010 International Symposium on. (2010) 1–4
16. Majeed, A., Zia, T.: Multi-set architecture for multi-applications running on wireless sensor networks. In: Advanced Information Networking and Applications Workshops (WAINA), 2010 IEEE 24th International Conference on. (2010) 299–304
17. OECD: Machine-to-machine communications: Connecting billions of devices. OECD Digital Economy Paper (192) (2011)

TR-MAC: An Energy-Efficient MAC Protocol for Wireless Sensor Networks exploiting Noise-based Transmitted Reference Modulation

Sarwar Morshed and Geert Heijenk

University of Twente, The Netherlands
{s.morshed, geert.heijenk}@utwente.nl

Abstract. Energy-constrained behavior of sensor nodes is one of the most important criteria for successful deployment of wireless sensor networks. The medium access control (MAC) protocol determines the time a sensor node transceiver spends listening or transmitting, and hence the energy consumption of the overall node. Transmitted reference (TR) modulation as the underlying physical layer provides new opportunities and challenges to be explored in the MAC layer. To utilize the advantages and overcome the challenges provided by the TR modulation, a new energy-efficient MAC protocol TR-MAC that uses noise-based carrier for wireless sensor networks is proposed in this paper. TR-MAC realizes multiple access using individual frequency offsets for a pair of nodes, allows both transmitter-driven and receiver-driven communication and is suitable for asynchronous low data rate application. TR-MAC enables energy-driven communication since nodes can adapt their duty cycle based on available energy, thus the protocol becomes energy-efficient.

Keywords: TR modulation, energy-efficiency, MAC protocol, TR-MAC

1 Introduction and Motivation

The Medium Access Control (MAC) protocol is responsible for addressing and providing a channel access mechanism to enable various nodes within a network to communicate with each other in a shared wireless communication medium. The physical layer underlying the MAC layer modulates the data to the reference signal in order to send it with a specified frequency through the medium. As opposed to regular modulation techniques, Transmitted Reference (TR) modulation [1] not only sends the modulated signal, but also sends the reference signal with a known time offset, as presented in Fig. 1 on the left. Following this, a receiver can restore the original data by correlating the received signal with a delayed version of itself with the same time offset since all multi-path components contain identically distorted pulses with consistent mutual delay. This interesting property of TR modulation allows to use noise as information carrier that is also easy to generate [2]. The receiver can restore the original signal without rake receiver or channel state information or power-hungry stable oscillators.

Consequently, the signal acquisition process becomes faster allowing for shorter synchronization time, giving a potential for reducing power consumption. Furthermore, frequency offsets can be used in place of time offsets, because the former are easier to implement on a chip [2]. Moreover, multiple nodes can transmit simultaneously by employing various frequency offsets without the need for mutual timing coordination. Therefore, TR modulation is suitable for asynchronous low data rate communication offering additional flexibility to the upper MAC layer. However, TR modulation consumes more power than a general modulation technique to transmit individual bits since the reference signal is also sent.

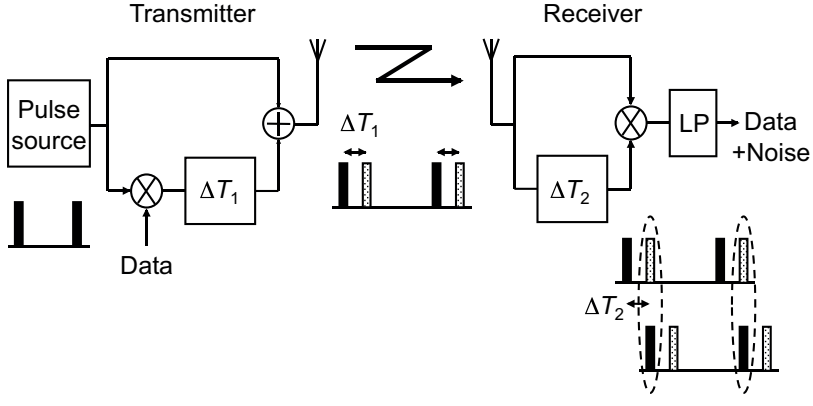


Fig. 1. Transmitted Reference modulation

In this paper we will investigate how we can optimally exploit the characteristics of TR modulation at the MAC layer in order to realize a wireless sensor network technique compatible with energy harvesting. We introduce a new MAC layer protocol, called TR-MAC, to exploit all the advantages provided by the TR modulation technique minimizing its drawbacks. Instead of sending long preambles to inform a receiver that wakes up about an upcoming data packet, TR-MAC sends data right away with a very short preamble as data packets in wireless sensor networks are generally very small. Moreover, the transmitter listens for acknowledgement from receiver after sending each data burst. Thus the transmitter is able to reduce the length of the consecutive data-listen bursts if it receives an acknowledgement from the receiver for unsynchronized links. When the link between a node pair becomes synchronized, then communication in TR-MAC can be transmitter-driven or receiver-driven as both the transmitter and receiver store each other's next wake up time. Finally, TR-MAC enables energy-driven communication by allowing the nodes to adapt their duty cycle based on the local energy availability.

The contributions of this paper are as follows: (1) we introduce a MAC protocol, TR-MAC, exploiting the characteristics of TR modulation; (2) we provide

basic models to analyze the energy consumption of this MAC protocol and two reference protocols; and (3) we evaluate the energy consumption and show that the introduced MAC protocol in combination with TR modulation compares favorably to the reference protocols.

This paper is organized in 6 sections. Related work in Section 2 is followed by TR-MAC protocol design in Section 3. Afterwards, Section 4 describes the TR-MAC modeling and Section 5 gives the results and analysis. Finally, Section 6 provides our conclusions and future work.

2 Related Work

In this section, we will analyze existing MAC protocols from the perspective of energy-efficiency. The proposed MAC protocols in the literature are classified into three categories: reservation-based, protocols with common active period, and asynchronous preamble sampling MAC protocols by [3]. After extensive analysis, the authors of [3] claimed that preamble sampling protocols are the most energy efficient category of MAC layer protocols. The preamble sampling protocols allow the nodes to wake up and sleep independently of the other nodes, thus they are termed Low Power Listening (LPL) protocols. However, the receiver has to wake up periodically to check for data transmission in the channel. The transmitter precedes the data packet with a preamble of maximum length equal to the receiver's sleep or check interval whenever it wishes to send any data. If the receiver detects some activity in the channel during its periodic wake up time, it continues to listen in order to receive the data from the transmitter. Compared to the other categories, the preamble sampling protocols have a greater energy saving capability with less need for network-wide management, thus are very suitable for low data rate asynchronous applications.

The preamble sampling protocols can be realized in three ways as mentioned in [3], and the references therein. Firstly, the transmitter can replace the long preamble by short preamble packet bursts with destination address to allow the target receiver to wake up later to receive data, whereas a non-target receiver can go back to sleep after receiving a single burst. Alternatively, the transmitter can send preamble-listen bursts to shorten its preamble length by an acknowledgement from the intended receiver if it wakes up. However, these protocols do not adapt preamble length for future transmissions and do not send any acknowledgement after successful data transmission. Protocols like X-MAC [4], SpeckMAC-B [5], ContikiMAC [6] are examples of this category of MAC protocols. We take X-MAC [4] as a reference protocol of this category. Secondly, the transmitter can adapt its preamble length by remembering the receiver's wake up time for forthcoming communications. However, these receiver-driven protocols are unfavorable for broadcast traffic where one transmitter has to wake up multiple times for its multiple neighbors. Furthermore, these protocols have to send the longest possible length of preamble for the first time communication. WiseMAC [7], CSMA-MPS [8], TrawMAC [9], SyncWUF [10] falls into this category. We take WiseMAC [7] as a reference for this class of protocols. Finally,

there are some protocols where the sensor node can adapt its duty cycle based on requests from the neighborhood, traffic load, or topology information. Nevertheless, these duty cycle adaptable protocols are suitable only for application specific scenarios, not for all scenarios; and they have no mechanism to adapt the communication based on energy availability on individual nodes.

3 TR-MAC Protocol Design

As introduced in Section 1, TR modulation is characterized by fast synchronization, allowing for the use of very short preambles; and inherent multiplexing, allowing for implicit identification of possibly simultaneous transmissions. To exploit these characteristics, and to mitigate the transmit power penalty of TR modulation, a new energy-efficient protocol, TR-MAC, is proposed that combines the best characteristics from all three categories of preamble sampling protocols. TR-MAC allows the receiver to detect any transmission in the channel with a very short preamble because of the inherent benefit provided by underlying TR modulation. Moreover, small data packet can be included in the preamble as the data packets in wireless sensor networks are generally very small, within a range of few bytes. As the preamble is a part of the data packet and TR-MAC sends data right away with the preamble, therefore from now on preamble-data will be referred as only data in this paper.

TR-MAC sends data multiple times to deal with uncertainty regarding the receiver's wake up time. Furthermore, just one bit in the data packet is enough to instruct the receiver to continue listening in case more large data packets are following the initial small ones. As transmission is costly for TR modulation, therefore TR-MAC introduces some listen periods just after every data packet where the transmitter listens the medium for acknowledgement from the receiver. Thus the transmitter minimizes the total data-listen duration based on the reception of acknowledgement from the receiver for first time communications. After first time communication, both transmitter and receiver synchronize their future communication by storing each other's next periodic wake up time to reduce the data-listen duration in order to save energy. Hence data transmission can be either transmitter-driven or receiver-driven when the link is synchronized between a pair of nodes, thus giving a considerable flexibility to the upper layers. Moreover, the duty cycle of a sensor node can be adapted based on either the available energy on nodes or application requirement. Therefore the newly proposed TR-MAC protocol is energy-driven, thus energy-efficient. Multiple access is another critical issue to address for any wireless sensor networks. Nodes following the traditional MAC protocols adapt their transmit times to deal with multiple transmitters attempting to access the channel simultaneously. However, the new TR-MAC protocol achieves multiple access using individual frequency offsets for a pair of nodes, a key advantage created by the underlying TR modulation. Hence collision can be avoided as future communication will take place in different virtual channels by using different frequency offsets.

The newly proposed TR-MAC protocol has three states, as shown in Fig. 2, namely (1) first time communication; (2) unsynchronized link; and (3) synchronized link. In the first stage for first time communication, a node does not have any information about its neighbors. Thus one node transmits data-listen bursts in the default frequency offset if it wants to send any data. The receiver periodically listens to the default frequency offset to detect any data transmission, like other preamble sampling protocols. When the intended receiver wakes up and receives a single the data burst, then it responds with an acknowledgement indicating a successful transmission. In this stage, the nodes perform the process of neighbor discovery, exchange the full MAC address, establish a link identifier and agree on the frequency offset to be used in the following communications.

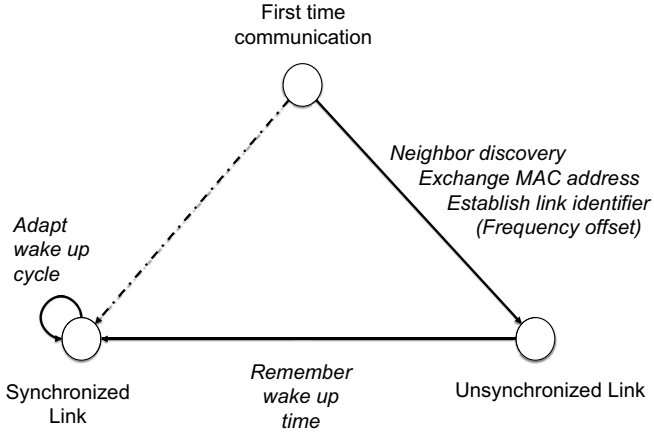


Fig. 2. TR-MAC: Three states

After the first time communication, the protocol moves to the next, unsynchronized link stage, as presented in Fig. 3. During this stage, the transmitter sends short data-listen bursts at the previously agreed upon frequency offset until it receives an acknowledgement from the receiver. The receiver listens to the agreed frequency offset for any data transmission. If the receiver is able to detect any data packet, then it can derive the link identifier from the listening offset and preamble of the data packet. A very small preamble is enough to detect any transmission in the channel because of the TR modulation. The receiver's next wake up time is specified in the acknowledgement packet indicating whether the check interval will be a normal one, or a half or double of the previous one based on traffic load or application requirement. Also a request for the transmitter to acknowledge in its next communication whether the transmitter will follow it or not is indicated. The transmitter mentions whether it agrees or not on the proposed time in its next transmission. Hence the nodes decide whether future communication will be transmitter-driven or receiver-driven. At this point, the

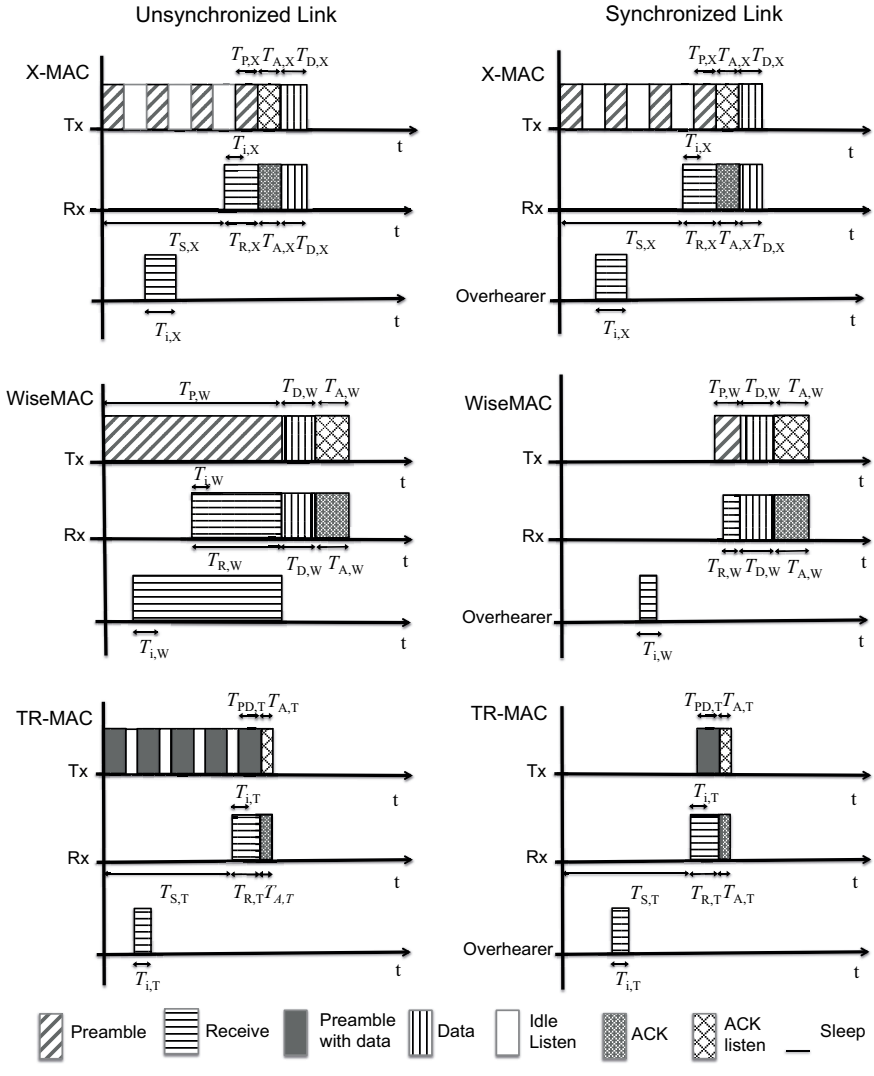


Fig. 3. Comparison of MAC Protocols

protocol advances to its final synchronized link stage where the link between a pair of node is synchronized, as shown in Fig.3. During this stage, the nodes already know the frequency offset and the specific time to wake up to reach a particular node, thereby optimally minimizing the data-listen burst length to as minimum as possible. Moreover, in this final stage, the nodes can adapt their duty cycles to meet the requirements of the network or another node having less energy to increase network lifetime. A new node joining the network sometimes can move to synchronized stage skipping the unsynchronized stage if the link identifier can be finalized between a pair of nodes.

The use of both transmitter-driven and receiver-driven duty cycling provides some interesting opportunities to realize energy-efficient multi-hop communications at the network level. For instance, TR-MAC is able to create a ripple effect while broadcasting still maintaining its energy efficiency. The transmitter can instruct its first hop neighbors to follow its lead and those can in turn instruct their respective neighbors to follow them in order to broadcast more efficiently: saving both energy and time. Finally, a system of Green Waves [11] can be created to deliver packets to their destinations with limited delay.

4 Modeling

In this section, we present a mathematical representation of TR-MAC for unsynchronized links in terms of energy consumption to send or receive a packet and for periodic listening. We also derived the analytical models for X-MAC and WiseMAC for the previously mentioned scenarios and compared with that of TR-MAC. The comparison results are presented in Section 5. In the symbols below, we use the comma separated subscript T, X, W to denote a symbol specific for TR-MAC, X-MAC and WiseMAC respectively. If the subscript is omitted, the symbol applies to multiple or all three MAC protocols.

To model the energy consumption to send or receive a packet, lets consider P_{Tx}, P_{Rx} and P_{S} to represent power to send, receive and sleep having values 1 mW, 1 mW and 15 μW respectively. TR-MAC sends both reference and modulated signals, thus its power level $P_{\mathrm{Tx,T}}$ is 2 mW. The data rate is considered as 25 kbps and a single data packet duration is considered with 32 bits having duration 1.28 ms. T_{W} represents the periodic check interval that is a summation of sleep duration, T_{S}, and periodic listen duration, T_{i}. The power consumption and time for switching from sending to receiving and vice-versa are much smaller compared to the other values, and are neglected in our modeling. The TR-MAC data packet, $T_{\mathrm{PD,T}}$, consists of 8 bits of preamble, $T_{\mathrm{P,T}}$, 16 bits of header, T_{H}, followed by 32 bits of data, T_{Data}, thus having 56 bits with duration 2.24 ms. We also consider the preamble duration, $T_{\mathrm{P,T}}$, is enough for the receiver to detect any transmission in the channel because of the inherent advantage provided by the TR modulation. The acknowledgement packet, $T_{\mathrm{A,T}}$, consists of 8 bits preamble, $T_{\mathrm{P,T}}$, and 16 bits header, T_{H}, in total 24 bits having duration .96 ms. The data packets of X-MAC and WiseMAC, $T_{\mathrm{D,X}}$ and $T_{\mathrm{D,W}}$ respectively, includes 16 bits header, T_{H}, and 32 bits data, T_{Data}, thus have 48 bits with dura-

tion 1.92 ms. The symbols and values for TR-MAC, X-MAC and WiseMAC are given in Table 1. The values used for TR-MAC reflect the main characteristics of TR-MAC, i.e., its very short preamble at the cost of increased transmission power.

Table 1. System parameters

Parameters	TR-MAC	X-MAC [4]	WiseMAC [7]
Preamble duration, T_{P}	8 bits (.32 ms)	65 bits (2.6 ms)	T_{W}
ACK duration, T_{A}	24 bits (.96 ms)	65 bits (2.6 ms)	80 bits (3.2 ms)
Header duration, T_{H}	16 bits (.64 ms)	16 bits (.64 ms)	16 bits (.64 ms)
Data duration, T_{Data}	32 bits (1.28 ms)	32 bits (1.28 ms)	32 bits (1.28 ms)
Data+header duration, T_{D}	56 bits (2.24 ms)	48 bits (1.92 ms)	48 bits (1.92 ms)
Power to send, P_{Tx}	2 mW	1 mW	1 mW
Power to receive, P_{Rx}	1 mW	1 mW	1 mW

All these preamble sampling protocols have the periodic listen as their background energy consumption, E_{PL}, given by Eq. 1. Here the periodic listen of TR-MAC, X-MAC and WiseMAC are $T_{\mathrm{i,T}}$, $T_{\mathrm{i,x}}$ and $T_{\mathrm{i,W}}$ respectively and the sleep duration are $T_{\mathrm{S,T}}$, $T_{\mathrm{S,X}}$ and $T_{\mathrm{S,W}}$ respectively.

$$E_{\mathrm{PL}} = \frac{P_{\mathrm{Rx}} T_{\mathrm{i}} + P_{\mathrm{S}} T_{\mathrm{S}}}{T_{\mathrm{S}} + T_{\mathrm{i}}} \tag{1}$$

TR-MAC periodic listen duration, $T_{\mathrm{i,T}}$, has to be greater than or equal to the duration of the acknowledgement duration, $T_{\mathrm{A,T}}$, plus two times preamble duration, $T_{\mathrm{P,T}}$, in order to detect a data transmission in the medium given by

$$T_{\mathrm{i,T}} \geq T_{\mathrm{A,T}} + 2T_{\mathrm{P,T}}. \tag{2}$$

We take the minimum duration for the periodic listen of TR-MAC in our calculation to minimize power consumption because that is enough to detect a transmission. Similarly the condition for periodic listen X-MAC is given by Eq. 3 and we take the minimum value of $T_{\mathrm{i,X}}$ for calculation.

$$T_{\mathrm{i,X}} \geq T_{\mathrm{A,X}} + 2T_{\mathrm{P,X}} \tag{3}$$

For WiseMAC, the minimum listen duration, $T_{\mathrm{i,W}}$, is taken as the minimum preamble duration, $T_{\mathrm{P,T}}$, because the receiver keeps listening in case it detects any transmission in the channel. Therefore, the periodic listen for TR-MAC is taken as 40 bits with duration 1.6 ms, for X-MAC is taken as 195 bits with duration 7.8 ms and for WiseMAC is taken as the minimum duration to detect any transmission, that is 8 bits with duration .32 ms, respectively.

The expected energy to send a single packet for TR-MAC, $E_{\mathrm{Tx,T}}$, is given by Eq. 4 and is derived as follows. The energy required for a single cycle of sending a packet of preamble and data, followed by listening for an acknowledgement (regardless of its receipt) is given by $P_{\mathrm{Tx,T}}T_{\mathrm{PD,T}} + P_{\mathrm{Rx}}T_{\mathrm{A,T}}$. Always at least one such cycle is needed for sending the data, hence the +1. Extra cycles might be needed depending on when the receiver wakes up. If we assume that the first cycle of sending a packet will start at an arbitrary moment between the start of two consecutive listen periods with duration $T_{\mathrm{i,T}} + T_{\mathrm{S,T}}$, we can derive the expected number of extra cycles. If the packet starts within the first $T_{\mathrm{i,T}} - T_{\mathrm{P,T}}$ seconds of the listen period of the receiver, no extra cycles are needed as the receiver will receive the complete preamble from which it can derive that it needs to stay awake for the rest of the packet. If the packet transmission starts later, with probability ${}^{(T_{\mathrm{S,T}}+T_{\mathrm{P,T}})}/_{(T_{\mathrm{i,T}}+T_{\mathrm{S,T}})}$, extra cycles will be transmitted until the next listen period of the receiver. On average, the number of extra cycles will be $\frac{1}{2}{}^{(T_{\mathrm{S,T}}+T_{\mathrm{P,T}})}/_{(T_{\mathrm{PD,T}}+T_{\mathrm{A,T}})}$. The energy needed to send a packet is thus given by

$$E_{\mathrm{Tx,T}} = \left(\frac{1}{2}.\frac{\left(T_{\mathrm{S,T}} + T_{\mathrm{P,T}}\right)^2}{\left(T_{\mathrm{i,T}} + T_{\mathrm{S,T}}\right)\left(T_{\mathrm{PD,T}} + T_{\mathrm{A,T}}\right)} + 1\right)\left(P_{\mathrm{Tx,T}}T_{\mathrm{PD,T}} + P_{\mathrm{Rx}}T_{\mathrm{A,T}}\right). \quad (4)$$

In addition to the periodic listen energy, the receiver has to spend extra energy to receive a packet. To calculate this additional energy, the periodic listen duration, $T_{\mathrm{i,T}}$, has to be subtracted from the expected extended listening duration, $\overline{T}_{\mathrm{R,T}}$, for the reception of the data packet. Furthermore, the energy to send an acknowledgement has to be added. The additional energy to receive a packet $E_{\mathrm{Rx,T}}$ is given by

$$E_{\mathrm{Rx,T}} = P_{\mathrm{Rx}}\left(\overline{T}_{\mathrm{R,T}} - T_{\mathrm{i,T}}\right) + P_{\mathrm{Tx,T}}T_{\mathrm{A,T}}. \quad (5)$$

The receiver listen duration has to be at least the duration of a data packet, and it can be extended up to the acknowledgement packet duration plus two times data packet duration depending on the random wake up time of the receiver. It is given by

$$T_{\mathrm{A,T}} + 2T_{\mathrm{PD,T}} > T_{\mathrm{R,T}} \geq T_{\mathrm{PD,T}}. \quad (6)$$

Given that the wake-up time of the receiver is uniform by distributed over the interval, the expected extended listening duration, $\overline{T}_{\mathrm{R,T}}$, is given by

$$\overline{T}_{\mathrm{R,T}} = \int_{t=T_{\mathrm{PD,T}}}^{T_{\mathrm{A,T}}+2T_{\mathrm{PD,T}}} tP(T_{\mathrm{R,T}} = t)dt = \frac{1}{2}T_{\mathrm{A,T}} + \frac{3}{2}T_{\mathrm{PD,T}}, \quad (7)$$

which is taken as 104 bits with duration 4.16 ms for calculation.

Similarly, the expected energy to send a packet for X-MAC, $E_{\mathrm{Tx,X}}$, is derived like TR-MAC and given by Eq. 8 with the exception that X-MAC needs to send the data packet separately.

$$E_{\mathrm{Tx,X}} = \left(\frac{1}{2} . \frac{\left(T_{\mathrm{S,X}} + T_{\mathrm{P,X}}\right)^2}{\left(T_{\mathrm{i},X} + T_{\mathrm{S,X}}\right)\left(T_{\mathrm{P,X}} + T_{\mathrm{A,X}}\right)} + 1 \right) \left(P_{\mathrm{Tx,X}} T_{\mathrm{P,X}} + P_{\mathrm{Rx}} T_{\mathrm{A,X}}\right) + P_{\mathrm{Tx,X}} T_{\mathrm{D,X}} \tag{8}$$

And the additional energy to receive a packet for X-MAC does not have the energy to receive a preamble because that is already calculated within the periodic listen energy. Therefore only the energy to send an acknowledgement and the energy to receive the data packet is present in the expression to calculate the additional energy to receive a packet, $E_{\mathrm{Rx,X}}$, given by

$$E_{\mathrm{Rx,X}} = P_{\mathrm{Tx,X}} T_{\mathrm{A,X}} + P_{\mathrm{Rx}} T_{\mathrm{D,X}}. \tag{9}$$

Finally, the expected energy to send a packet for WiseMAC, $E_{\mathrm{Tx,W}}$, includes the energy to send a preamble, then to send the data and later to receive the acknowledgement,

$$E_{\mathrm{Tx,W}} = P_{\mathrm{Tx,W}} T_{\mathrm{P,W}} + P_{\mathrm{Tx,W}} T_{\mathrm{D,W}} + P_{\mathrm{Rx}} T_{\mathrm{A,W}}. \tag{10}$$

The receiver has to spend additional energy $E_{\mathrm{Rx,W}}$ to listen for the preamble, then to listen for the data packet, at last to send the acknowledgement. So,

$$E_{\mathrm{Rx,W}} = P_{\mathrm{Rx}} \left(\overline{T}_{\mathrm{R,W}} - T_{\mathrm{i,W}}\right) + P_{\mathrm{Rx}} T_{\mathrm{D,W}} + P_{\mathrm{Tx,W}} T_{\mathrm{A,W}}, \tag{11}$$

where the average receiver listen duration is $\overline{T}_{\mathrm{R,W}} = \frac{T_{\mathrm{W}}}{2}$.

5 Results and Analysis

We compute the analytical models of TR-MAC, X-MAC and WiseMAC for unsynchronized links in Matlab in order to compare their energy consumption to send or receive a packet and for background energy in periodic listening. We vary the check interval duration, T_{W}, for these protocols and measured the energy consumption. The symbols and corresponding values are given in Section 4 and Table 1. Fig. 4 depicts the energy to send a packet. It can be observed that TR-MAC consumes less energy than WiseMAC even though TR-MAC needs more power to transmit a single packet. The reason is that TR-MAC adapts its total data-listen duration based on acknowledgement from receiver whereas WiseMAC sends a full preamble of length equal to the check interval duration T_W. However, X-MAC has less energy consumption than TR-MAC because of the underlying

TR modulation technique that needs more power to transmit both reference signal and the modulated signal. Nevertheless, we expect to have advantage over X-MAC because unlike X-MAC, TR-MAC has defined synchronized links where the transmitter is able to adapt the start of the data-listen sequence to an optimized value based on the receiver's next wake up time.

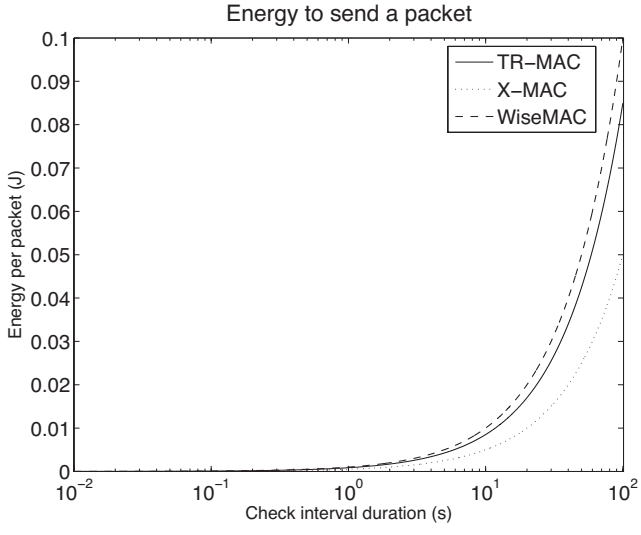


Fig. 4. Energy to send a packet

The total energy spent at the receiver side can be divided in two parts: periodic listen and additional energy to receive a packet, presented consecutively in Fig. 5 and Fig. 6. The periodic listen energy in Fig. 5 shows that TR-MAC is better than X-MAC as TR-MAC is capable of detecting transmission with a smaller preamble, thus the listen duration can be smaller. Also the smaller acknowledgement duration in TR-MAC allows to have smaller periodic listen duration. Nevertheless, WiseMAC seems better than TR-MAC having small periodic listen duration. However, WiseMAC overhearers will spend much energy for receiving a packet since they have to listen the whole preamble duration if they detect any communication in the channel in order to receive the data afterwards. For each packet sent, an overhearer spends the additional energy shown in Fig. 6. However, a TR-MAC overhearer can go back to sleep just after receiving a single data burst. So no extra energy is spent by overhearers in TR-MAC besides the energy consumption shown in Fig. 5. Thus TR-MAC will spend much less energy in the long run though WiseMAC has better performance for background energy consumption due to periodic listen. Fig. 6 represents the additional energy consumption of the protocols to receive a packet where the curves for TR-MAC and X-MAC overlap and show very small energy consumption, but

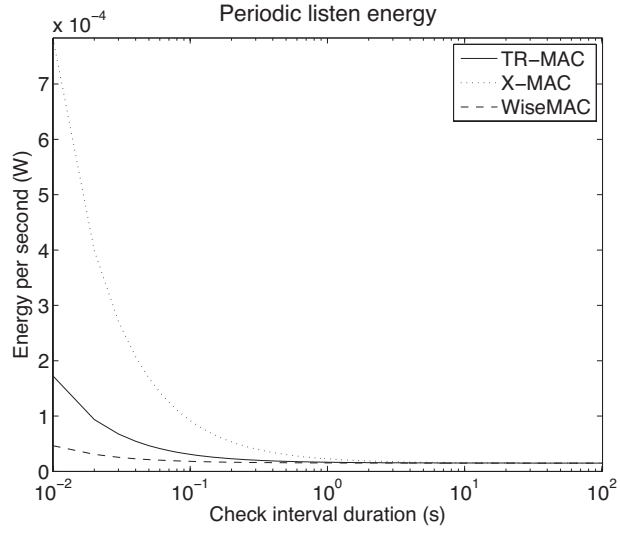


Fig. 5. Periodic listen energy

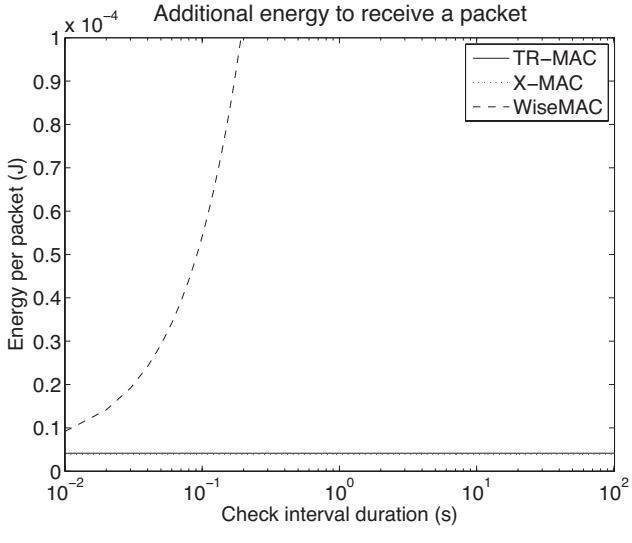


Fig. 6. Additional energy to receive a packet

WiseMAC consumes much energy as a receiver and a potential overhearer has to listen on average half of the check interval duration in order to receive a packet.

6 Conclusions and Future Work

The TR-MAC with noise-based TR modulation underneath is an energy-efficient MAC protocol suitable for short-range, low data rate applications that utilizes all the usefulness of TR modulation while minimizing its drawbacks. TR-MAC has many attractive characteristics. A transmitter using TR modulation can use noise as information carrier and a receiver can save energy by faster synchronization time without power hungry stable oscillators. In addition, TR-MAC is capable of sending data right away without long preambles and receivers can detect transmission by listening to the specified offset with the link identifier. Moreover, nodes can adapt their duty cycle based on available energy, thus the protocol is totally energy-driven. Furthermore, TR-MAC can be both transmitter-driven and receiver-driven based on the application requirement, thus gives much opportunities for energy-efficient routing in the network layer.

We modeled and compared the unsynchronized links stage of TR-MAC with X-MAC and WiseMAC. It turns out that TR-MAC has a very low energy consumption for periodic listening, which is not affected by overhearing transmission for other receivers, as in the case of WiseMAC. Furthermore, similar to X-MAC but contrary to WiseMAC, TR-MAC needs very little energy to receive a packet. Finally, transmitting a packet is more costly than in X-MAC, but this can be compensated by choosing a shorter check interval. Overall, TR-MAC is very promising for energy-efficient communications in noisy environments, where only a limited amount of data is transmitted between a single pair of nodes.

As our future work, we will model the synchronized link stage for TR-MAC and compare with X-MAC and WiseMAC. We expect to have better performance for synchronized link stage as TR-MAC has interesting features for that. In addition, we will compare TR-MAC with some other protocols that send data instead of preamble. We will also evaluate TR-MAC with traffic adaptivity in multi-hop networks. Finally, energy harvesting will be incorporated in future by letting transmitters and receivers adapt their duty cycle based on locally available energy.

Acknowledgement

This research is supported by the Dutch Technology Foundation STW, which is part of the Netherlands Organisation for Scientific Research (NWO), and partly funded by the Ministry of Economic Affairs. It is done in the context of STW project 11317 - Walnut: Hard to Crack - Wireless Ad-hoc Links using robust Noise-based Ultra-wideband Transmission. Also Arjan Meijerink is acknowledged for editorial assistance.

References

1. Hoctor, R., Tomlinson, H.: Delay-hopped Transmitted-Reference RF Communications. In: IEEE Conference on Ultra Wideband Systems and Technologies, pp. 265-269 (2002)
2. Meijerink, A., Cotton, S.L., Bentum, M.J., Scanlon, W.G.: Noise-based Frequency Offset Modulation in Wideband Frequency-selective Fading Channels. In: Proceedings of the 16th Symposium on Communications and Vehicular Technology in the Benelux, Louvain-la-Neuve, Belgium (2009)
3. Cano, C., Bellalta, B., Sfairopoulou, A., Oliver, M.: Low Energy Operation in WSNs: A Survey of Preamble Sampling MAC Protocols. In: Elsevier Computer Networks, Vol: 55, Issue: 15, pp. 3351-3363 (2011).
4. Buettner, M., Yee, G., Anderson, E., Han R.: X-MAC: A Short Preamble MAC Protocol for Duty-cycled Wireless Sensor Networks. In: Proceedings of the 4th International Conference on Embedded Networked Sensor Systems, pp. 307–320. Colorado, USA (2006)
5. Wong, K., Arvind, D.: SpeckMAC: Low-power decentralised MAC protocols for low data rate transmissions in specknets. In: Proceedings of the 2nd International Workshop on Multi-hop Ad hoc Networks: from Theory to Reality, pp. 71–78. New York, USA (2006)
6. Dunkels, A.: The ContikiMAC Radio Duty Cycling Protocol. Swedish Institute of Computer Science, Tech. Rep. T2011:13 (2011)
7. El-Hoiydi, A., Decotignie, J.: WiseMAC: An Ultra Low Power MAC Protocol for Multi-hop Wireless Sensor Networks. In: Algorithmic Aspects of Wireless Sensor Networks, pp: 18–31. Turku, Finland (2004)
8. Mahlknecht, S., Bock, M.: CSMA-MPS: A minimum preamble sampling MAC protocol for low power wireless sensor networks. In: Proceedings of the IEEE International Workshop on Factory Communication Systems, pp. 73–80 (2004)
9. Zhang, X., Ansari, J., Mähönen, P.: Traffic aware medium access control protocol for wireless sensor networks. In: Proceedings of the 7th ACM International Symposium on Mobility Management and Wireless Access – MobiWAC'09, p. 140 (2009)
10. Shi, X., Stromberg, G.: SyncWUF: an ultra low-power MAC protocol for wireless sensor networks. In: IEEE Transactions on Mobile Computing vol. 6, issue. 1, pp 115–125 (2007)
11. Guha, S., Chau, C., Basu, P.: Green Wave: Latency and Capacity-efficient Sleep Scheduling for Wireless Networks. In: Proceedings of 29th IEEE International Conference on Computer Communications INFOCOM, pp. 1900-1908. San Diego, CA (2010)