*Journal of Location Based Services*
Vol. 00, No. 00, Month 200x, 1–18

# Cortina: Collaborative Indoor Positioning Using Low-Power Sensor Networks

Gianni Giorgetti[1]*, Richard Farley[2], Kiran Chikkappa[2], Judy Ellis[2], and Telis Kalea[2]

[1]Navizon Inc., Miami Beach, FL, USA.
[2]Advanced Technologies Group, Qualcomm Inc., San Diego, CA, USA.

(*Received 00 Month 200x; In final form 00 Month 200x*)

Cortina is a distributed *Real-Time Location System* (RTLS) designed to track assets or people moving indoors. Our solution leverages a low-cost, low-power *Wireless Sensor Network* (WSN) based on the IEEE 802.15.4 radio standard. The network, which consists of wall-plugged nodes, is designed to be self-configuring, self-healing and self-calibrating, thus reducing deployment and maintenance costs. Assets and people are tracked using small battery operated wireless tags that collect Received Signal Strength (RSS) measurements from nearby nodes. The tags also include an accelerometer for activity recognition, and a barometric pressure sensor to detect the floor plan. We have conducted experiments over a 2000 m$^2$ area instrumented with eighteen sensor nodes. Our initial results show that the system can track people in real-time with an average error of 2.8 m.

## 1 Introduction

Current smart phones feature a rich set of sensors and include GPS receivers. The ability to sense the phone's environment and its position has enabled a multitude of context-aware and *Location Based Services* (LBS). These applications already assist us in many of our everyday operations. But even more powerful services and business opportunities can be created by extending the range of interaction with the physical world. Users will soon have the need to gather knowledge from their surroundings, interact with a variety of consumer electronic devices, and obtain accurate location information, both outdoors and indoors.

Cortina is a research project that leverages low-power, mesh *Wireless Sensor Networks* (WSNs) to instrument indoor spaces. The long-term goal is to provide a platform to develop novel context-aware applications that interact with a large number of sensors and actuators embedded in the user's environment. Our short-term efforts have focused on using this distributed sensing infrastructure to implement a *Real-Time Location System* (RTLS) to track people or assets. Possible applications for our system include elderly and children monitoring, offender tracking, security guards supervision, and tracking of test equipments and IT assets. We aim to demonstrate that the WSN approach is not only feasible, but also competitive with respect to other positioning technologies.

### 1.1 *Indoor Positioning Using WSNs*

In the past few years, indoor positioning has received a great deal of attention both from the research community and the industry (Liu et al., 2007; Mao, Fidan and Anderson, 2007). Several solutions are available to enable location awareness in environments with insufficient GPS reception. For example, applications demanding high accuracy can leverage *Ultra-Wide Band* (UWB) systems based on *Time Difference of Arrival* (TDoA). Lower cost applications can be supported by RTLS based on pre-existing infrastructure such as WLANs and cellular networks. Previous research has demonstrated that these solutions can locate users with low error, typically two to three meters when the systems are calibrated by collecting fingerprinting data (Bahl and Padmanabhan, 2000; Youssef et al., 2003).

Cortina is an indoor RTLS that exploits an inexpensive sensor network in the attempt to further optimize the trade-off between performance and cost. The aim of this work is to describe our system and outline

*Corresponding author. Email: gianni@navizon.com. Note: This work was done when G. Giorgetti was with Qualcomm, Inc.

*Journal of Location Based Services*
ISSN 1748-9725 print / ISSN 1748-9733 online © 200x Taylor & Francis
http://www.tandf.co.uk/journals
DOI: 10.1080/1748972YYxxxxxxxx

Figure 1. One of the wall-plugged nodes of the Cortina positioning system. On the left corner, a web page that displays the real-time position of the user.

technical details that can facilitate implementation of other future RTLSs. In the present paper, we have extended the initial description of the Cortina system (Giorgetti et al., 2011) with additional details and more insights about the achieved results.

The design principles for our system are discussed in Section 2. We have strived to create an RTLS that is accurate, but also low cost, easy to deploy, and easy to maintain. The proposed RTLS is based on wireless nodes that plug in the available power outlets (see Figure 1). The nodes form a network that is self-configuring, self-healing, and self-calibrating. Such network generates *beacons* that provide location information to small battery-operated radio tags. These tags enable positioning by measuring the *Received Signal Strength* (RSS) of the beacons transmitted from the fixed units. Given their small size and light weight, the tags can be worn by people or attached to asset to enable a multitude of location tracking applications.

Section 3 provides a general overview of the hardware and software components that implement Cortina. In addition to the sensor network architecture, we discuss the flow of information toward the server-side services, the positioning engine, user interfaces and diagnostic tools.

In Section 4 we provide details about the hardware, the firmware, and the positioning algorithms used in our system. The Cortina RTLS is based on many custom solutions that made it possible to achieve satisfactory performance with a minimum time expenditure for system deployment and maintenance. We believe that several of these solutions can simplify the implementation of other similar positioning systems.

Section 5 discusses experimental results. Our initial testbed consists of an 18 node sensor network deployed in the left wing of building QRC6, over an area measuring approximately $2000\,\mathrm{m}^2$. The system, which has been running continuously for more than 14 months, has been used to track in real time the locations of several volunteers. In controlled tests, we have measured an average localization error of 2.8 meters. In 80% of the cases, the users were located within four meters of their true position. The experience gathered during qualitative tests performed over the course of several months suggests that this level of accuracy should be sufficient to support many location aware applications. We conclude reporting some of the lessons learned and plans for future work.

## 2 Design Principles

Cortina is an RTLS designed to track the location of assets and people moving indoors. Our solution aims to be: 1) **low-cost**, 2) **easy to deploy**, and 3) **accurat**e. These design goals are achieved by leveraging low-power IEEE 802.15.4 WSNs in conjunction with custom solutions that will be discussed in the following sections.

The WSN is a computing paradigm that has gained popularity in the past years and already counts many

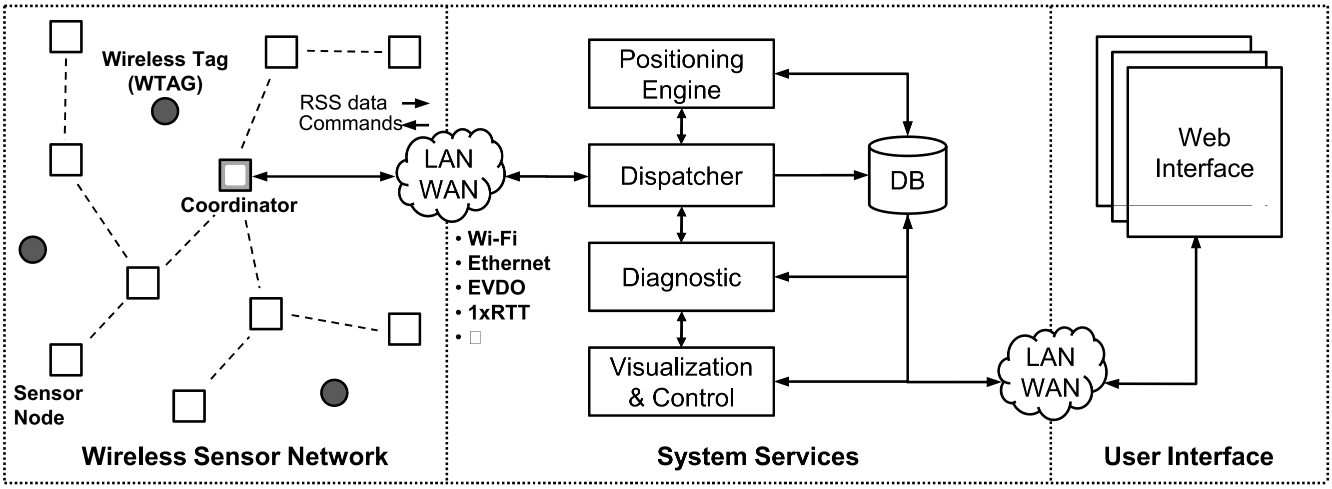



Figure 2. Architecture of the Cortina system.

applications in civilian and military domains (Akyildiz et al., 2002). A sensor network consists of a group of low-cost radio devices enabled with sensors and/or actuators. These units are capable of organizing themselves into an ad-hoc, multi-hop wireless network that enables data collection and dissemination. In Cortina, the WSN is used to create the fixed infrastructure of our RTLS. This infrastructure interacts with mobile tags and collects the information used to estimate their position. The simplicity of the hardware used and the self-configuring nature of sensor networks are in line with our two first design goals: the WSN-based infrastructure is both inexpensive and easy to deploy. Ease of deployment and maintenance are further strengthened by a novel RFID-based solution that enables the system to detect the location of any new plugged sensor. This solution is described in details in Section 4.1.2

The need to maintain a low overall cost has also directed us toward designing an RTLS based on RSS measurements. RSS-based positioning is attractive because it can be implemented using simple hardware, both on the fixed and mobile nodes. Most transceivers include circuitry to measure the RSS, and simple *beaconing* is sufficient to collect signal strength data. The simplicity of the RSS approach contributes to achieve a low system complexity and enable a low-power design, which is a critical factor in achieving long battery lifetime on the wireless tags used for tracking.

But effective implementation of RSS-based RTLS is challenged by multi-path propagation, scattering, and diffraction of the RF signal. Since signal propagation varies widely from building to building, many of the best performing RTLS's leverage fingerprint models that are periodically recalibrated by measuring the RSS at known locations (Liu et al., 2007).

Cortina avoids the need for calibration by adopting a collaborative approach that use RSS measurements collected by the sensor nodes at fixed positions. The nodes in our network are programmed to monitor changes in the RSS, which are taken into account to continuously recalibrate the positioning algorithms (see Section 4.5). The collaborative approach not only alleviates the deployment cost, but also makes the system more robust to changes in the environment. For example, people moving in and out of conference rooms, or relocation of big pieces of equipment, cause significant fluctuations in the RSS that are automatically taken into account by the recalibration process. The positioning algorithms are also designed to compensate for the variability in RSS originated by using transceivers with uncalibrated output power and different antenna gains.

## 3 System Overview

Figure 2 shows the abstract view of the Cortina system and its three main components: 1) the wireless sensor network, 2) a collection of software services and diagnostic tools, and 3) user interfaces. Each of these subsystems is discussed in the following sections. The targets to track are equipped with small Wireless

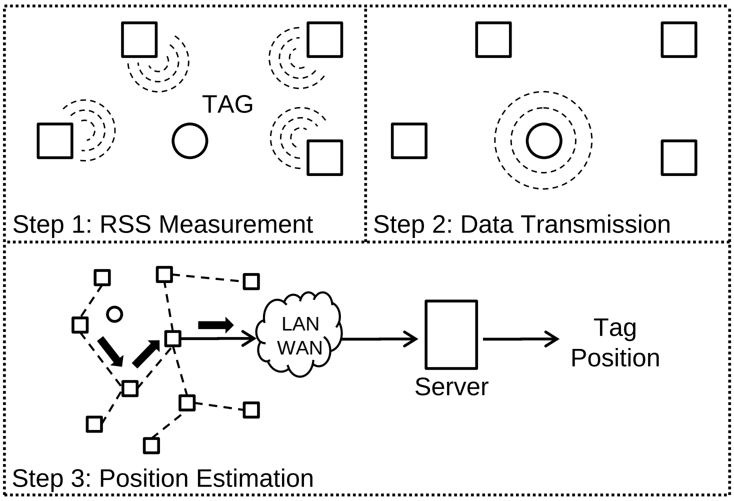



Figure 3. The three steps of the position estimation process.

Tags (WTAGs) that interact with sensor nodes plugged into the power outlets (see Figure 3). Periodically, each WTAG wakes up and measures the RSS of the beacons broadcasted by the surrounding nodes. The RSS data is aggregated in a few packets and transmitted back to the WSN. Finally, this information is relayed to a central server where it is used to estimate the target position.

### 3.1 *Wireless Sensor Network*

The wireless sensor network includes both the fixed sensors and the WTAGs used to track assets and people. Our WSN is based on the IEEE 802.15.4 standard, a protocol that has become popular for implementation of low-power, low-rate sensor networks and *Personal Area Networks* (PANs). Numerous hardware vendors produce transceivers compliant with the IEEE 802.15.4 specification in the 868 − 915 MHz and the 2.45 GHz bands. Additionally, several upper layer protocols such as ZigBee, WirelessHART, 6LoWPAN, and TinyOS can be implemented using IEEE 802.15.4 networks.

As mentioned earlier, the devices in our WSN belong to two main categories: sensor nodes and wireless tags.

**3.1.1** ***Sensors Nodes.*** Sensor nodes implement the fixed infrastructure of our RTLS. Each node is a small radio unit designed to be plugged into a power outlet (see Figure 4-left). Sensor nodes operate continuously according to a schedule that includes three main activities:

(i) **Beaconing.** Sensors broadcast short beacons that serve as reference signals for the WTAGs and also enable our self-calibrating positioning algorithms.
(ii) **RSS Measurements.** Each sensor node constantly listens to beacons transmitted by neighboring nodes. Every time a beacon is received, the receiving node updates a table that keeps track of the transmitter's RSS, the number of messages received, and other statistics useful to evaluate the quality of the radio channel. Every five minutes, these tables are transmitted to the server where they are used to recalibrate the positioning models.
(iii) **Packet Routing.** Sensor nodes also listen to measurement messages generated by the WTAGs and other nodes. Depending on their location, nodes can help forward data packets to a special unit that serves as the network coordinator.

The WSN coordinator is a special node that connects to a remote server using an additional hardware interface. In our implementation, all data generated within the WSN is transferred to the server using an Ethernet connection (see Figure 4-center). The coordinator could be augmented with other interfaces such as WiFi, EVDO, 1xRTT, etc. depending on specific application requirements.

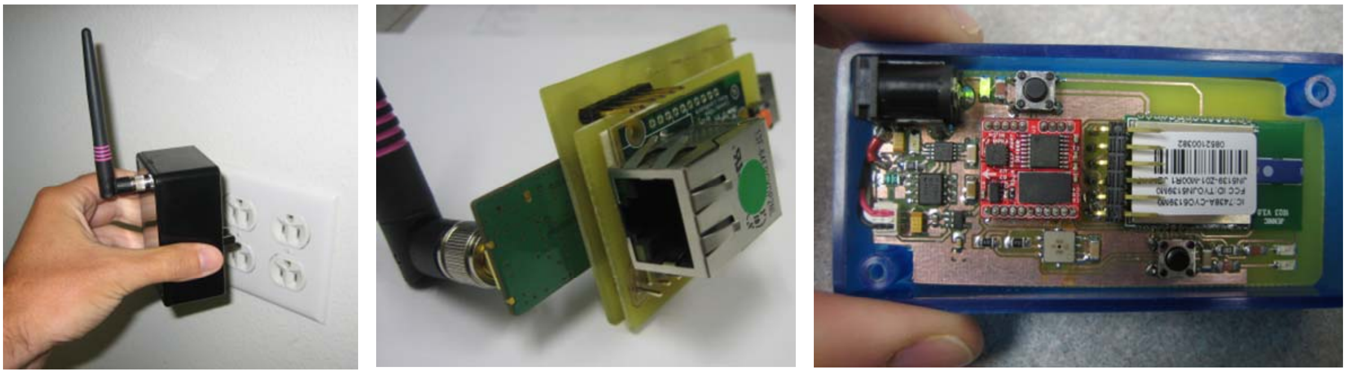


Figure 4. left) One of our sensor nodes plugged into a power outlet; center) a detail of the Ethernet module used on the coordinator to support LAN connectivity; right) one of our wireless tags (WTAG)

**3.1.2** ***Wireless Tags (WTAGs).*** WTAGs are small battery-operated devices that exchange information with the sensor nodes (see Figure 4-right). The WTAGs can be attached to assets or worn by people. Once a tag arrives in proximity of the Cortina WSN, it starts collecting RSS information from surrounding nodes. This information is transmitted back to the coordinator and used to estimate the position of the tag.

To conserve energy, the hardware can be kept in sleep mode for most of the time. By adjusting the duty cycle of the WTAGs, the system designer can achieve the desired tradeoff between frequency of the location updates and battery lifetime. To better exploit this approach, WTAGs are equipped with inertial sensors that can detect if the unit is being moved. The frequency of the location updates can be increased when movement is detected, and reduced to a minimum when the tag is stationary. Additionally, each tag also features a barometric pressure sensor to enable accurate 3D positioning in multi-floor buildings (see Section 4.1.4).

### 3.2 *System Services*

The data collected within the network is remotely transferred to a set of services that implements higher level functionalities of Cortina (see Figure 2).

**Dispatcher.** The Dispatcher module receives a stream of data from the WSN and stores the measurements in a database. It makes real time data available to other modules and allows sending configuration commands to the WSN. For example, if the system administrator wants to change the beaconing rate of the sensors, or their radio channel, appropriate commands can be transmitted to the WSN through the dispatcher.

**Positioning Engine.** The RSS measurements received by the WSN are processed by the location engine module to estimate the position of a mobile target. The location engine uses both historical information stored in the database and real time data transmitted by the WTAGs. Every five to ten minutes, the RSS data measured between each pair of nodes is retrieved from the database and used to calibrate the positioning models of the system. When RSS transmitted by WTAG are intercepted, the updated positioning models take part in the computation that estimates the target position.

### 3.3 *Diagnostic Tools, Visualization and Control Services*

This last suite of components complements the server services and allows users and system administrators to obtain information from the system. The information produced by Cortina is accessible through web pages that display real time position information, animated location traces, and diagnostic pages useful to evaluate the status of the wireless sensor and the wireless channel. More details about these tools are provided in Section 5.

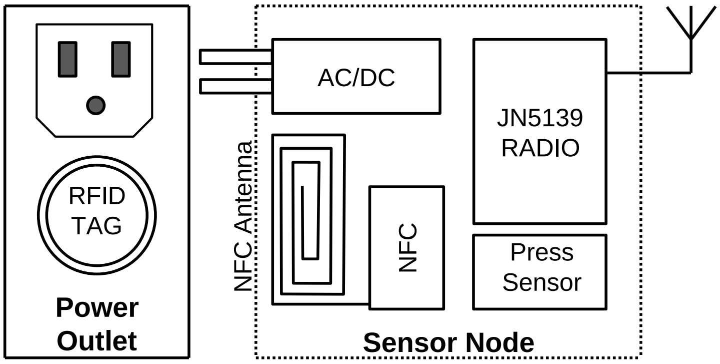



Figure 5. Diagram of a wall-plugged sensor. Upon plugging a node, the sensor identifies its location by reading an RFID tag attached to the power outlet.

## 4 Implementation

The following section contains implementation details about the hardware, the firmware that controls the WSN, and the positioning algorithms.

### 4.1 *Hardware*

**4.1.1** ***Sensor Nodes and Coordinator.*** We have designed our sensor nodes based on the Jennic JN5139 wireless microcontroller (Jennic JN5139), a module that includes a 16 MHz 32-bit RISC processor, 96 KB of RAM, 192 KB of ROM, and a 2.45 GHz transceiver compatible with the IEEE 802.15.4 standard. Since both the sensor nodes and the coordinator are plugged into the power outlets, achieving a low-power consumption on these units is not critical; therefore we have chosen the high power version of the JN5139, a unit that features an output power of 19 dBm and a receiver sensitivity of $-100$ dBm. Connected to a 1/2 wave whip antenna, this radio can provide coverage up to 4 kilometers in open space. Inside the QRC building, we have successfully established communication between nodes with a distance in excess of 70 meters. The use of radios with high link budget ensures reliable communications over large areas. In addition, high power transmissions increase the number of beacons received by each WTAGs. Since these messages are used to compute tag locations, a higher number of message improves the localization results.

Apart of the transceiver, each sensor node includes an AC/DC adapter that converts the 110VAC into 3.3 VDC, a Bosch BMP085 barometric pressure sensor (*Bosch BMP0085 Datasheet*), and a PN65K Near Field Communication (NFC) chip (*NXP PN65K Datasheet*) manufactured by NXP with respective antenna. Additionally, we have equipped our network coordinator with the Connect One Nano LANReach, an embedded module with an UART TTL interface that enables connectivity to 10/100BaseT Ethernet LAN (see Figure 4-center). All the traffic collected by the coordinator is routed to the server through this interface.

**4.1.2** ***NFC For Node Self-Positioning.*** The NFC chip mounted on the sensor nodes is part of a novel solution that enables Cortina to automatically detect the position of any new plugged sensor. To reduce maintenance cost and improve system robustness, we have devised a self-positioning scheme that leverages inexpensive RFID tags. Our approach requires attaching an RFID tag to each of the power outlets in the building. This is a onetime operation that can be performed before deploying the system, or, ideally, it can be carried out at the time the edifice is built.

As a result of the RFID tagging, each power outlet is associated with a unique code linked to its physical location. Every time one of the sensors is plugged into a power outlet, the NFC chip reads the code associated with the RFID tag. The code is then transferred wirelessly to the central server where it is used to retrieve the location of the node just plugged. We found this solution particularly effective

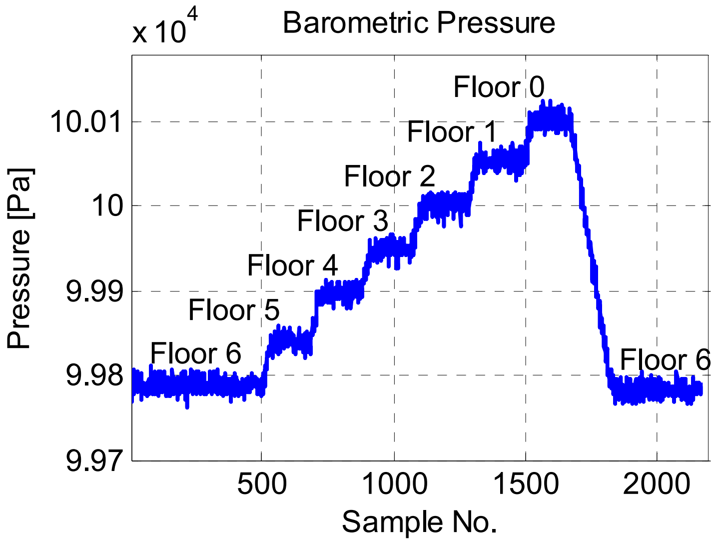



Figure 6. Barometric pressure readings collected when riding the elevator from the sixth floor of QRC to the basement. On the way down, the elevator was briefly stopped on each floor..

in reducing deployment and maintenance efforts and also avoid system malfunctioning due to misplaced nodes. Additionally, the RFID based self-positioning scheme allows the system to be maintained by non-skilled personnel: sensor nodes can be easily deployed or relocated, and defective sensors can be replaced with new units without requiring manual reconfiguration.

**4.1.3** ***Wireless Tag.*** The wireless tags share some of the same electronics used in the sensors (see Figure 4-right), but the components have been chosen to reduce the size and power consumption. The WTAG features the standard version of the JN5139, a module with the output power of 2.5 dBm and an on-board ceramic antenna. The device consumes about 37 mA when in a RX/TX mode, and just 2.6 $\mu$A when radio and CPU are in sleep mode. The communication range is limited to $20 - 30$ m indoors, which is sufficient as the tag only needs enough power to transmit to the closest sensor.

The use of a low-power radio makes it possible to extend the battery lifetime, which is a critical factor to reduce maintenance costs when tags are used for asset tracking or deployed in areas that are not easily accessible. Applications using the same type of radio may have a lifetime in the order of months or years when the duty cycle is adequately reduced (Jennic, 2008).

Additional hardware on the WTAG includes a Bosch BMP085 pressure sensor, the same as in the wall sensors, and the OceanServer OS-4000T (*OceanServer OS4000-T User Manual*) inertial navigation unit with a three axis, tilt compensated digital compass, and a three-axis accelerometer. Finally, each WTAG module is powered by a 610 mAh Li-Po battery and includes circuitry for battery charging.

**4.1.4** ***Barometric Pressure Reading.*** The barometric pressure sensors mounted on the fixed nodes and the WTAGs were included to improve 3D localization when tracking targets in multi-floor buildings. At the moment, all the sensor nodes of our RTLS are deployed on the same floor, but we are currently planning to extend the network to other floors. We also realize that, while many applications can tolerate errors of a few meters in the horizontal plane, similar errors along the vertical dimension can position a target on the wrong floor. Barometric pressure readings represent a viable solution to reduce the frequency of this type of errors. Figure 6 shows the output of the Bosch BMP085 sensor when riding the elevator from the sixth floor to the basement.

This preliminary data suggests that accurate floor detection is possible using barometric pressure measurements. Additionally, we plan to use the readings collected by the fixed sensor nodes to calibrate the system and account for variation in pressure that occur daily due to different weather conditions.

### 4.2 *Network Firmware*

The firmware installed on the nodes, coordinator and wireless tags controls the network formation, the exchange of messages between the units, and the collection of RSS measurements. This data is eventually transferred to a central server where it is converted into position estimates. Our firmware, which is implemented in C, interacts with the lower layers of the radio stack. This section first discusses the functionalities supported by these layers, and then explains the custom solution implemented in Cortina.

**4.2.1 *IEEE 802.15.4 Communication Stack.*** Cortina implements a wireless sensor network based on the popular IEEE 802.15.4 standard. The specification defines both the physical (PHY) and the Medium Access Control (MAC) layer. The PHY layer controls properties of the signals transmitted such as frequency, modulation, synchronization, etc., while the MAC layers regulates the access to the channel and the network formation.

In the IEEE 802.15.4 standard, devices willing to communicate must first join a personal area network. The PAN is established by the coordinator that selects the radio channel and broadcasts a unique PAN ID. Other devices can join the same network by scanning all the available radio channels and searching for the particular PAN ID.

The IEEE 802.15.4 only controls communication between devices within their radio range. The implementation of multi-hop network topologies requires upper layer protocols such as ZigBee or 6LoWPAN. To implement our WSN, we have chosen to use JenNet (Jennic JN-UG-3041), a network protocol provided by Jennic, the manufacturer of the radios used in our system. After evaluating different alternatives, the JenNet stack seemed to provide the best tradeoff between complexity and functionality.

JenNet implements a *collection tree protocol* that runs on top of an IEEE 802.15.4 network. From an application point of view, it makes it easy to transfer information to the coordinator. It also provides a solution that is scalable, easy to deploy, and robust. Devices running the JenNet stack dynamically select the most reliable route to the coordinator and are able to react to changes in the network topology. For example, if a radio link fails, the affected nodes can automatically find alternative routes to the coordinator (self-heal). New units can be added at any time by simply deploying them within the radio range of any arbitrary sensor that has already joined the network (self-configuration).

**4.2.2 *RSS Measurements Collection.*** An essential component of Cortina is the firmware that controls the collection of RSS measurements. RSS data is used both to calibrate the position models and to estimate the location of each target. While signal strength can be easily measured by exchanging radio beacons, different collection protocols are possible.

Most systems implement one of two main positioning paradigms: tracking or navigation. In both cases, the mobile units interact with nodes at known position, but there are differences in the transmission and reception of the signals used for positioning.

In a **tracking** system, a tag briefly wakes up to transmit beacons to the fixed units. These units intercept the messages and forward the information to a central server where the measurements are used to estimate the target position. Since the tag only wakes up to transmit a few messages, power consumption and hardware complexity are kept to a minimum. On the down side, all the fixed sensors have to relay the measurements to a central point; therefore the network bandwidth can limit the number of units that can be tracked, especially if the tags frequently wake up to update their positions.

In a **navigation** system, message transmission follows the reverse path. The fixed nodes transmit reference messages that are used by the mobile units to compute their position. This is the same approach used by the GPS. Since the mobile nodes have to compute their own location, these devices are more complex and more power consuming than tracking tags. On the up side, once a navigation system is in place, an arbitrary number of users can be supported.

Cortina initially adopted a tracking architecture, but we realized that the bandwidth of the IEEE 802.15.4 radios, 250 Kbit/s, could limit the number of targets that can be tracked. Our latest firmware reduces the bandwidth usage by exploiting a hybrid approach that shares some of the traits of a navigation

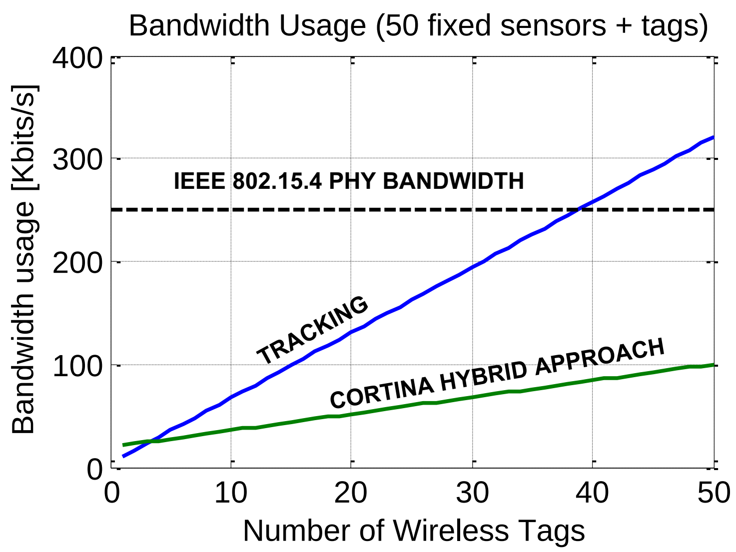



Figure 7. Simulation of bandwidth usage for an RTLS that uses a tracking approach and our Cortina implementation.

system. The information used to localize the target is transmitted by the fixed sensors at a rate of one beacon every TB time units - currently TB = 1 sec. When a target needs to update its position, the tag uses a navigation approach: the radio is turned on for a few seconds to measure the RSS from the nearby sensors. The wake up time is controlled by the TW parameter - currently TW = 4 sec. At the end of the wake up period, the available RSS measurements are packed into a few messages and transmitted back to the network. The location computation is performed at the server, the WTAG is only responsible for RSS data collection. Since the packets transmitted by the WTAG contain all the information measured by target, significant bandwidth savings can be achieved. Only one copy of the WTAG messages needs to be forwarded to the coordinator.

Figure 7 shows the result of a simulation to evaluate the bandwidth usage of our system. We assume 50 fixed nodes deployed in a common area where every node is in radio range of every other node. The bandwidth usage is computed for an increasing number of targets that update their position every five seconds.

### 4.3 *Routing Scheme*

The hybrid mode implemented in Cortina required some tweaking of the underlying protocol layers. In a standard IEEE 802.15.4 network, devices taking part in the communication are required to join a common PAN. With the Jennet stack in place, this operation normally takes between seven to ten seconds. This delay is not problematic for fixed nodes that only need to join the network once, but it would compromise the functionality of a wireless tag that often goes to sleep. To reduce the delay, we have developed a custom WTAG firmware that bypasses the JenNet stack and part of the IEEE 802.15.4 layer. The WTAG radio is programmed to operate in *promiscuous mode*, basically acting as a packet sniffer that intercepts all the messages on its radio channel. Since no association is required, RSS measurements can begin almost instantaneously after waking up the WTAG. The wake-up time for the JN5139 radio is 2.75 ms.

To achieve the bandwidth usage shown in Figure 7, we also had to implement a custom routing protocol that avoids flooding the network with indiscriminate packet retransmissions. The hybrid approach implemented in Cortina is effective only if RSS information collected by the WTAG is transmitted back to the coordinator along a single routing path. To limit the number of packets transmitted, we have developed a distributed scheme that only requires retransmission from the closest sensor. The WTAG messages contain RSS measurements from the nearby sensors. When these packets are transmitted back to the network, each receiving node inspects the message content to determine if its beacons were the ones with strongest RSS. If this condition is satisfied, the sensor assumes to be the closest unit to the WTAG and forwards the message to the coordinator. The other units discard the message. This simple scheme has proven effective in routing the WTAG packets with a minimum number of retransmission and good reliability.

### 4.4 *Activity Recognition Firmware*

The firmware on the WTAGs implements an algorithm that detects human body activity in real-time using a single tri-axial accelerometer. Activity recognition allows the system to gather additional information when people are being monitored, (e.g. children or elderly) and also allows the WTAG to save battery energy by reducing the frequency of the location updates when the person is not moving.

The firmware recognizes four main activities: i) Motionless, ii) Fidgeting, iii) Walking, and iv) Running. The code for data acquisition, data processing, feature extraction, and activity classification is executed on the WTAG. Since only few bytes of information need to be transmitted back to the network, bandwidth usage and power consumption are kept to a minimum. On the other hand, the JN5139 has only modest computational resources, so the algorithms had to be designed to work with a low amount of memory and computation. We also designed the activity recognitions software so that it can work independently of the position and the orientation of the WTAG on the person's body (see Figure 8).

In the initial stage of our research, we evaluated algorithms based on decision trees, nearest neighbor, Bayesian classification, and signal statistics such as mean, variance, energy, and spectral entropy. Due to computation and memory limitation previously discussed, our solution has subsequently focused on the two simple algorithms described in the sections below.

**4.4.1** ***Energy Correlation Across Axes.*** The first approach determines the user's activity by collecting accelerometer readings at 20 Hz over a three second interval and computing the cumulative energy for each axis. The ratio of the cumulative energy levels among the $x, y, z$ axis is then used to determine the user's activity by comparing the values against predetermined thresholds computed during controlled tests.

Comparing the energy ratios is computationally inexpensive, but this approach has the disadvantage that the WTAG needs to be mounted at a fixed location and in a fixed orientation. The algorithm also has to be recalibrated for different body types. For example, we noticed differences in the values returned when the WTAG was worn by subjects with fat or lean midriff.

**4.4.2** ***Number of Slope Inversions.*** The second approach does not place any restriction on the orientation or location for mounting the accelerometer. Using the same 20 Hz sampling rate and data collected over a three second interval, we count the number of slope inversion for each axis $x$, $y$, and $z$. The sum of slope inversions for the three axes is different when walking, running, standing, or fidgeting and provides a good metric to classify the user's activity. The activities are obtained by comparing the values against reference thresholds determined during preliminary observations. Using this approach, we were able to determine the user's activity with a high degree of accuracy. Qualitative results from one of our tests are shown in Table 1.

In general, the algorithm determines the correct activity when the WTAG is mounted on the ankle, waist, shoulder, or head. We did preliminary tests with ten people in the age group from twenty to forty years, and the results were accurate in most cases without the need to adjust the pre-determined thresholds. Less

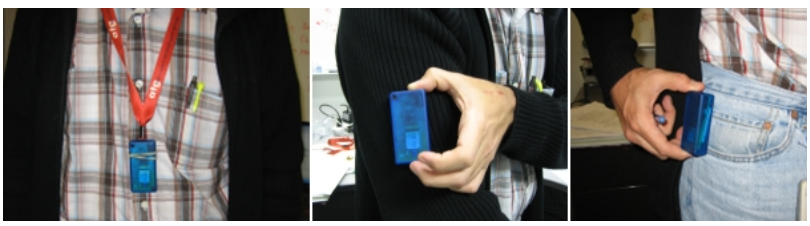


Figure 8. Examples of WTAG worn in different positions.

Table 1. Qualitative results of the activity recognition firmware

| | **Walking** | **Running** | **Motionless** | **Fidgeting** |
|---|---|---|---|---|
| **Ankle** | walking | running | motionless | fidgeting |
| **Knee** | running | running | motionless | walking |
| **Waist** | walking | running | motionless | fidgeting |
| **Shoulder** | walking | running | motionless | fidgeting |
| **Head** | walking | running | motionless | fidgeting |

reliable results were produced with the WTAG mounted on the knee, which exhibits an extreme range of motion at all gaits. In this case, the algorithm's thresholds need to be adjusted based on the range of motion exhibited by an individuals' gait.

### 4.5 *Positioning Engine*

The position of each WTAG is computed by the server on the basis of the RSS measurements transmitted by the WTAG itself. Currently, Cortina implements two collaborative positioning schemes that are described in the following sections: the first one is based on trilateration; the second one uses RSS maps.

**4.5.1** ***Trilateration.*** Basic geometry suggests that if a wireless target can accurately estimate its distance from at least three reference points, then its location can be computed without ambiguity. In practice, distance estimates are always noisy; therefore better results can be achieved using measurements from a larger number of reference points (Giorgetti, Gupta and Manes, 2008). Trilateration is a general technique to compute the position of a target using an arbitrarily large set of distance estimates from fixed points.

Computing the position of a target using trilateration requires a two-step process: First, the RSS values transmitted by the WTAG need to be converted into distance estimates. Second, the available distance estimates need to be processed to generate a single target position.

**Step 1: Converting RSS Values Into Distance Estimates** RSS values are converted into distance estimates using a model that is calibrated in real-time based on the data collected by the sensor nodes and their known distances. Currently, our system counts 18 fixed nodes deployed over an area of 2000 $\mathrm{m}^2$ in building QRC6. Since all the nodes are able to communicate with each other, 18x17=306 points are available for the model calibration. The cloud of dots in Figure 9a shows some of the RSS-distance pairs collected in a typical day. In our deployment, the separation distance between any two sensor nodes varies between 7 and 52 m, and the RSS values are comprised between −16 dBm and −90 dBm.

The measurements are used to estimate the parameters of a polynomial regression model:

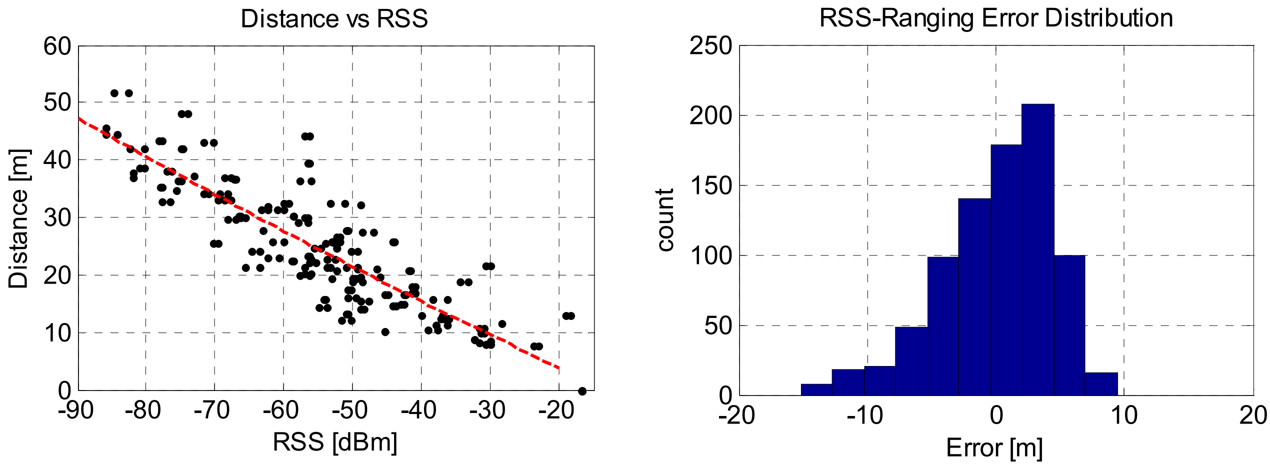
$$\hat{d} = d(\mathrm{RSS}) = a_0 + a_1\mathrm{RSS} + a_2\mathrm{RSS}^2. \tag{1}$$

The coefficients of the polynomial are computed by least square fitting, an operation that can be easily implemented in MATLAB using matrix division. The dotted line in Figure 9 shows the interpolating function for the available RSS data.

In our experiments in QRC6, we often measure a linear dependency between distance and RSS. The contribution of the $a_2\mathrm{RSS}^2$ term in (1) is almost negligible. When this happens, we can adopt an improved model that also accounts for differences in power transmission of the sensor nodes. Instead of using a constant coefficient $a_0$, we create a model that uses a different constant for each of the sensor nodes:

$$\hat{d} = d_i(\mathrm{RSS_i}) = \mathrm{a}_0^{(\mathrm{i})} + \mathrm{a_1RSS_i} + \mathrm{a_2RSS_i^2}, \tag{2}$$

where $\mathrm{RSS_i}$ is the signal strength measured from node $i$. This improved model compensates for the variability in output power that is normally measured across different units.




(a) RSS-Distance pairs collected by the fixed sensor nodes and the interpolating polynomial (red line).

(b) Distribution of the RSS ranging error. The errors are computed by moving a test unit over 65 fixed locations in QRC6 and using the RSS to estimate its distance from the sensors nodes. Seventeen nodes were deployed at the time of the test.

Figure 9.

To characterize the error of the distance estimates obtained using (2), we performed a controlled test at 65 fixed locations. At each location, we used a test unit to collect approximately four of five RSS values from each of the remaining nodes. The average RSS was used to estimate the distance between the test unit and each of the other nodes. We compute the estimation error as:

$$e_i^{(k)} = d_i(\mathrm{RSS}_i) - d_i^{(k)}, \tag{3}$$

where $k$ is the index of the test location, and $d_i^{(k)}$ is the distance between the location of node $i$ and the test point $k$. Figure 9b shows the error distribution of the distance estimates. The mean of the distribution is 0.16 m, and its standard deviation is 4.38 m. The error distribution using the simpler model given by (1) has similar mean, but a larger standard deviation equal to 4.8 m.

**Step 2) Converting Distance Estimates Into Absolute Positions.** Equation (2) is used to converted the RSS measurements collected by each WTAG into a set $D$ of estimated distances:

$$D = \{\hat{d}_j\} = \{d_j(\mathrm{RSS}_j)\}, j \in C, \tag{4}$$

where $\hat{d}_j$ is the estimated distance from node $j$ based on signal strength $\mathrm{RSS}_j$, and $C$ is the set of node for which a measurement is available. The absolute target position is computed by minimizing a quatdratic error function:

$$(\hat{x}, \hat{y}) = \arg\min_{(x,y)} \left[ \sum_{j \in C} \left( \hat{d}_j - \mathrm{dist}_j(x, y) \right)^2 \right], \tag{5}$$

where $\mathrm{dist}_j(x, y)$ denotes the Euclidean distance between the point at coordinates $(x, y)$ and the location of node $j$. This expression is solved iteratively using the `lsqnonlin` function available in MATLAB.

It can be shown that when the error for the distance estimates is described by *independent and identically distributed* (i.i.d.) normal random variables, then (5) is the *Maximum Likelihood Estimation* (MLE) for the target position.

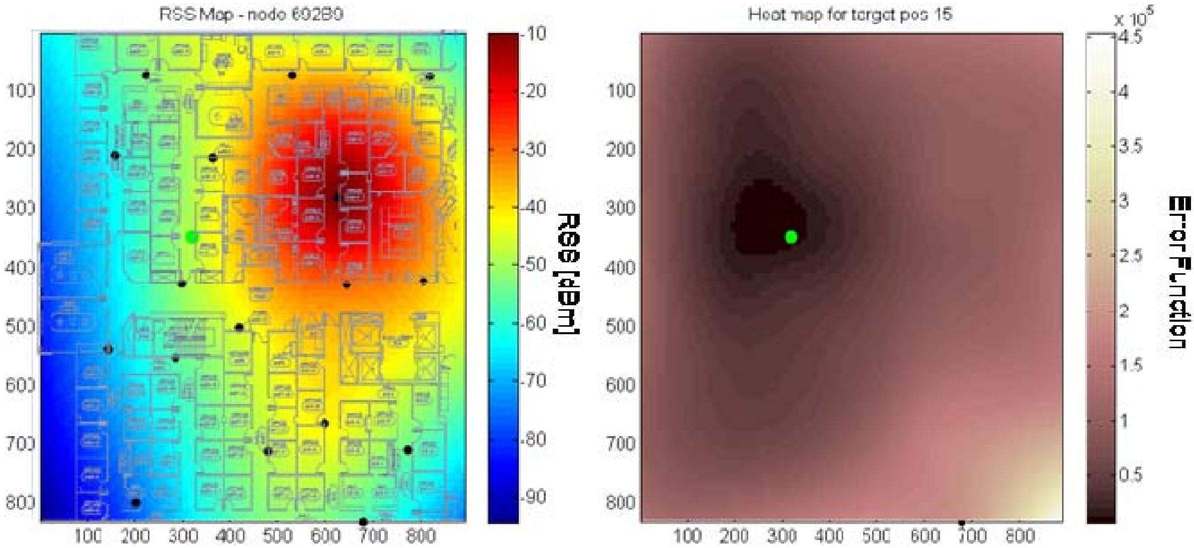



Figure 10. left) one of the maps computed using RSS values measured by the sensor nodes; right) heat map computed by comparing the WTAG's measurements to the available RSS maps. The true target position (green dot) is close to the minimum of the error function.

**4.5.2** ***Accounting For Unknown Receiver Gain.*** The propagation model in the previous section is derived using measurements among sensor nodes that share the same type of transceiver and the same type of antenna. The RSS measured by the WTAG are always attenuated due to the reduced antenna dimension and different polarization. The difference between the two antennas, which is evident in Figure 4, causes an attenuation of approximately 20 to 30 dBm when the WTAG is used to measure the RSS.

As a result of the WTAG attenuation all the distance computed using (2) are strongly biased. To remove the bias effect, the target position is estimated by minimizing a modified error function that uses pseudo-ranges:

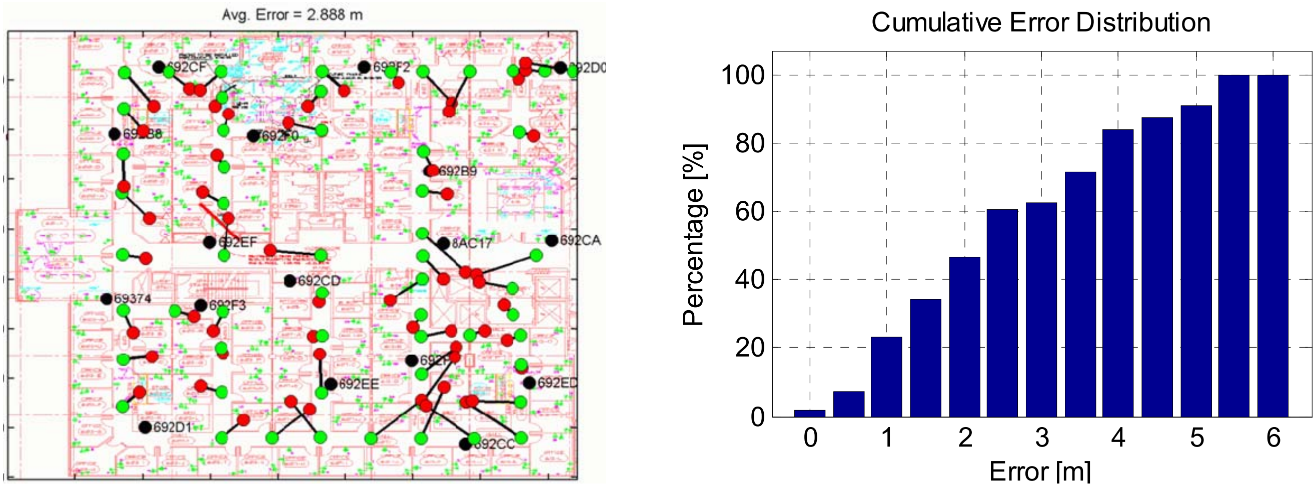
$$(\hat{x}, \hat{y}, \hat{b}) = \arg \min_{(x,y,b)} \left[ \sum_{j \in C} \left( \hat{d}_j - b - \text{dist}_j(x, y) \right)^2 \right]. \tag{6}$$

The bias $b$ is a nuisance parameter that we estimate together with the coordinate $(x, y)$ and compensate for unknown attenuation introduced by the WTAG. This expression is derived by noting in our environment, the distance is a linear function of the RSS, therefore a constant attenuation correspond to a constant bias.

## 4.6 *RSS Maps*

The second positioning scheme implemented in Cortina uses RSS maps derived from the available measurements. Ideally, if we had a map that accurately describes how the signal propagates within the deployment area, we could determine the target position by comparing the measured RSS values against the map. As already discussed, such a map is difficult to compute a-priori, but it can be approximated by taking RSS measurements at different fixed points. This approach is used in many fingerprinting localization schemes (e.g. (Bahl and Padmanabhan, 2000)) that use maps created by measuring the RSS form WiFi routers or cell towers.

Our second positioning scheme bears similarity to a fingerprinting scheme, but the maps are automatically computed using the values collected by the sensor nodes. Other previous work has adopted the concept of "sniffing" devices to recalibrate the maps (Krishnan et al., 2004). In Cortina, the "sniffers", i.e. the sensor nodes, are integral part of the system and collaborate with each other to detect changes




(a) Tracking error for a target moving on the sixth floor of the QRC building. The green dots are the true target positions, the red dots are the position estimates. The black dots represent the fixed nodes.

(b) Cumulative error distribution for the positions shown on the left map.

Figure 11. Location tracking results.

in the radio environment. Every time a sensor node $i$ at location $(x_i, y_i)$ transmits a beacon, the RSS is simultaneously measured by all the nodes in its radio range. For each node, we create an RSS map using a 2D linear interpolation algorithm that takes into consideration the available RSS values. Figure 10(left) shows one of these maps. The radio signal for the node in the figure does not propagate symmetrically: the presence of long corridors favors the propagation of the signal in the direction North to South.

Let $\mathrm{RSS}_{\mathrm{M}i}(x, y)$ be the function that describes the map for node $i$ at various locations of our deployment. If the WTAG had the same antenna gain as the other devices, we could estimate its position by comparing the RSS values with those of the maps. In practice, the WTAG introduces an unknown attenuation that need to be compensated. To remove the influence of this unknown gain, we compute the target position by minimizing an error function that takes into account the differences in RSS between a reference node $i$ and the other nodes:

$$(\hat{x}, \hat{y}) = \arg\min_{(x,y)} \sum_{i \neq j, i,j \in C} \left(\Delta\mathrm{RSS}_{ij} - \Delta\mathrm{RSS}_{\mathrm{M}ij}(x, y)\right)^2 \tag{7}$$

where $\Delta\mathrm{RSS}_{ij} = \mathrm{RSS}_i - \mathrm{RSS}_j$ is the difference in signal strength measured from two nodes, and $\Delta\mathrm{RSS}_{\mathrm{M}ij}(x, y) = \mathrm{RSS}_{\mathrm{M}i}(x, y) - \mathrm{RSS}_{\mathrm{M}j}(x, y)$ is the same difference predicted by the map at location $(x, y)$. Since (7) compares differences in RSS, the receiver's gain does not affect the estimated position. Figure 10-right shows an heat map computed using (7).

## 5 Results

The Cortina RTLS has been running continuously for more than 14 months, tracking in real time the location of about ten volunteers who offered to wear the one of the WTAGs. Except for hardware failures of a few nodes due to unfiltered surges in the power outlets, the system has worked reliably without requiring maintenance. The next few sections describes some of our test results and some of the user interfaces created as part of the project.

### 5.1 *Mobile Target Tracking*

One of the main goals of Cortina is to demonstrate target tracking capabilities using a WSN. In a controlled experiment, we marked 56 locations on the sixth floor of the QRC building. One of our collaborator walked

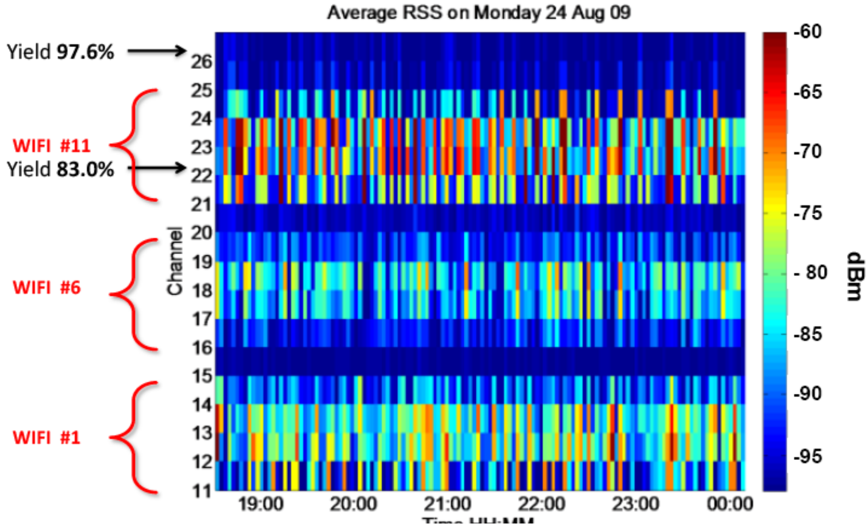



Figure 12. A plot of the floor noise measured on the sixteen IEEE 802.15.4 channels (ch 11 - 26). The noise is due to WiFi devices operating on channels 1, 6, and 11 of the IEEE 820.11.g spectrum.

from point to point holding one of the WTAGs in her hand. The WTAG was programmed to collect RSS measurements for approximately 5 seconds at each location. The collected data was used to obtain two position estimates: one obtained using trilateration and the other one using the RSS map algorithm. In the current approach, the system returns a position estimate that is the arithmetic average of the two values.

Figure 11a shows the floor plan of our test area with the fixed locations marked by green dots. The red dots are the estimated positions; the black line connecting green and red dots represent the error. The error in this controlled test is comprised between 0.38 and 5.95 m, with an average error equal to 2.88 m. Figure 11b shows the cumulative error distribution. In eighty percent of the cases, the error is below 4.25 m.

In analyzing the results in Figure 11 we note two things. First, all the test locations were chosen along corridors to avoid disturbing people working inside offices. Tracking WTAGs moving in corridors is generally less accurate because the corridors themselves act as wave guides for the signal coming from nodes at distant locations. The resulting effects is that location estimates are generally biased toward the center of the floor. The RSS propagation model used to compute the target location is estimated using the links between all pair of nodes, and each of these links generally crosses many walls. As a result, the computed model describes the signal propagation sufficiently well on a global scale, but is less effective in predicting the RSS along the corridors. During everyday usage, we have noted that the system is more accurate when estimating the position of WTAGs inside offices.

The second point to remark is that the locations of moving targets are computed independently from each other. Improved results could be obtained by applying filters that take into account the constraint of human motion, or, even better, the geometry of the building derived from the floor plan. Unfortunately floor plans are not always available, or come in a variety of formats that make it difficult to automate their processing. When developing Cortina, we found that the accuracy of the raw tracking traces was already sufficient for many context-aware applications; therefore we opted for a simple and straightforward approach that does not rely on filtering.

### 5.2 *Coexistence with WiFi Networks*

Some of our initial concerns were related to the coexistence of our network with the existing WiFi infrastructure. Both IEEE 802.15.4 and WiFi radios share the same ISM 2.45 GHz band, and some previous publications suggest that operating the two networks in the same physical space can lead to severe malfunctioning (Zensys Inc., 2007). Other work have reported minimal interferences between the two networks (Gilles Thonet et al., 2008; Jennic JN-AN-1079, 2008).

To better evaluate the effect of WiFi on our network, we have measured the noise floor over the 16

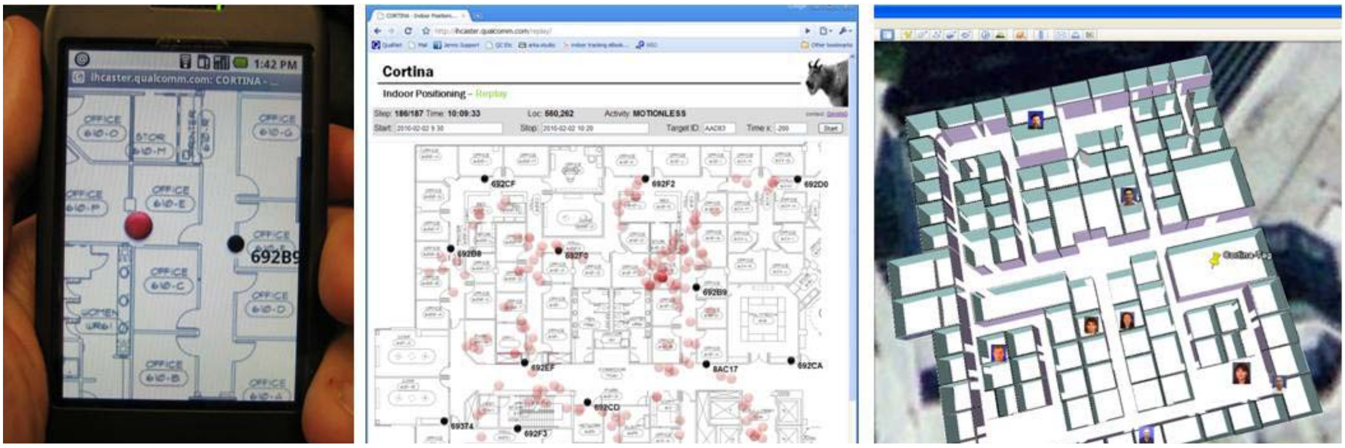



Figure 13. left) AJAX web page that displays real-time information about the position of a target; center) web page that displays location traces describing the movement of a target over a period of time; right) Google Earth real-time visualization of the target's position.

channels available in the IEEE 802.15.4 standard. These channels span a frequency band comprised between 2400 and 2483.5 MHz. At the moment of the test, eleven IEEE 802.11g access points were active on our floor.

Figure 12 shows that most of the energy radiated by WiFi is concentrated in three bands that correspond to the IEEE 802.11 channels 1, 6, and 11, which are the channels used by the access point in our building. Four of the sixteen IEEE 802.15.4 channels fall in between the WiFi channel and are free from noise.

We normally operate our WSN on channel 26 of the IEEE 802.15.4. In a controlled experiment involving transmission and receptions of about 50'000 packets between our nodes, we measured a very low Packet Error Rate (PER) equal to 2.4%. To quantify the effect of WiFi interferences, we repeated the same test setting our radios on channel 22, which is the one where the noise from WiFi is stronger. The packet error rate in this case was equal to 17%. Despite a lower yield, the system was able to collect measurement and most of the values were correctly stored in the database. Our conclusion is that coexistence between IEEE 802.11 and IEEE 802.15.4 system is not problematic even if they need to share overlapping channels. If the systems can be set to operate on non overlapping channels, no significant interferences should be expected.

### 5.3 *Network Management And Visualization Tools*

Operating an RTLS also requires the availability of tools to evaluate the status of the system. Diagnostic tools are important to assess the soundness of the data used during our research, and they are also fundamental to reduce downtimes in commercial grade system.

As part of our research work, we have focused on creating simple web interfaces that allows us to monitor the status of the network and the quality of the radio channel. We also worked on interfaces to display the real-time position of our WTAGs. Figure 13-left, shows an AJAX web page that displays the position of a selected user. The page can be accessed using a laptop or a smart phone. In addition to displaying location information, the page also reports information about the user's activity (e.g. stationary, walking, etc.).

Historical location information can also be accessed using a second web page that displays an animation of the user's movements during a selected period of time (see Figure 13-center). The web interface provides an option to replay the WTAG location using an accelerated time factor. Several hours of location traces can be replayed in a just a few minutes or seconds.

Finally, real-time position information and location traces are also accessible using a plug-in that runs on top of Google Earth (see Figure 13-right). The Google Earth rendering engine shows a 3D representation of the floor plan and the position of the WTAG.

## 6 Conclusions

The research work produced within the Cortina project has demonstrated the viability of WSNs as a low-cost tool to reliably collect contextual information from a physical space. Most our work has been focusing on collecting RSS data to support target tracking, but the sensing infrastructure and the data collection mechanism can be easily expanded to collect additional information. The BMP085 sensor included on our nodes already measures temperature and barometric pressure, and other sensors such as humidity, ambient light can be integrated in our platform with minimal hardware changes.

Our preliminary tests have shown the soundness of the WSN approach in designing a self-configuring, self-calibrating RTLS capable of tracking people and assets with an average error of approximately two to three meters. This level of accuracy is sufficient to correctly detect the room or the corridor where the target is moving without requiring filtering algorithms that use knowledge of the floor plan to correct the position estimates.

During our tests with the system, we particularly appreciate the simplicity of deployment and maintenance. The RFID-enabled, wall-plugged sensors make it easy to extend the network or replace defective units. The ability to plug additional sensor nodes is also important to improve the accuracy of the system. During our preliminary tests we have discovered locations where the position estimates were consistently worse than others. The errors in these "problematic" areas are mostly caused by the specific configuration of our building. For example, WTAGs moving at the end of long corridors might be inaccurately located due to the different signal propagations that is not captured by the propagation model. In general, the errors in problematic zones can be greatly reduced by plugging a new sensor node in one of the closest available plugs. The limited cost of the units and the self-configuring nature of the system make this solution a viable option to quickly improve the localization performance.

The activity recognition algorithm running on the WTAGs is also particularly effective in improving the performance of the localization system, especially when the tags are stationary for some time. When a tag is not moving, location updates occurs every five minutes. If the tag remains in the same location, the system will average the incoming measurements until the tag moves again. As a consequence, the location of static tags is progressively refined as new measurements are transmitted. For a tag that remains motionless for one hour or more, we noticed that the error reduces to less than one meter in most cases. The improved accuracy achievable using inertial measurements is especially important when tracking small valuable assets that might be misplaced in some remote location.

### 6.1 *Future Work*

As part of our future research work we plan to perform a more exhaustive set of experiments to characterize the position error of our system and the accuracy of the activity recognition firmware running on the WTAG. We also plan to further investigate the following topics:

- Propagation Models and Positioning Algorithms. Currently we are using a simple propagation model based on polynomial interpolation of the available RSS data. We realize that many models have been proposed in the literature to estimate the path loss of indoor radio communication. Some of them take into account the number of walls between any two units and the attenuation introduced by specific building materials. We plan to evaluate some of these models and their applicability to our problem.
- 3D Localization. Our initial experiments with BMP085 have shown promising results in detecting the floor using barometric pressure readings. In the near future we plan to deploy our WSN on multiple floors of the QRC building and enable 3D target tracking. We will also update our interfaces to display real-time positions in 3D.
- Data Mining. Cortina is currently collecting a large amount of information about the location of people moving in our building. We plan to use the available data to extract knowledge about the users' behavior and their environment. One immediate application is the autonomous creation of floor plan maps using the location traces of the wireless tags. By monitoring the position of several people over time, the system can learn the location of corridors and offices when a floor plan is not available.
- Outdoor/Indoor Tracking. We are currently integrating some of our WTAGS with a CDMA/GPS mod-

ule. These tags will be able to track people and assets on a global scale using GPS and the cellular network. When one of this device is moved in a building equipped with a Cortina network, indoor positioning will be seamlessly supported using the solutions presented in this work.

## 7 Acknowledgements

We thank Martin Renschler, Gene Marsh, and Steve Dorner for their helpful comments. We also thank the staff from the Proto Lab at Qualcomm for their help in realizing the hardware prototypes, and David Fischer for his help in realizing some of the web visualization tools.

## References

Akyildiz, IF, W. Su, Y. Sankarasubramaniam and E. Cayirci. 2002. “Wireless sensor networks: a survey.” *Computer Networks* 38(4):393–422.

Bahl, P. and VN Padmanabhan. 2000. “RADAR: an in-building RF-based user location and tracking system.” In Proc. *IEEE INFOCOM.*

*Bosch BMP0085 Datasheet.* 2008. `http://www.bosch-sensortec.com/content/language1/downloads/BMP085_DataSheet_Rev.1.0_01July2008.pdf`.

Gilles Thonet et al. 2008. “ZigBee WiFi Coexistence, Test Report.” `http://www.zigbee.org/imwp/idms/popups/pop_download.asp?contentID=13184`.

Giorgetti, G., R. Farley, K. Chikkappa, J. Ellis and T. Kaleas. 2011. “Cortina: Collaborative Indoor Positioning Using Low-Power Sensor Networks.” In Proc. *IPIN* .

Giorgetti, G., S.K.S. Gupta and G. Manes. 2008. “Localization Using Signal Strength: To Range or Not To Range?” In Proc. *MELT* .

Jennic. 2008. “Asset Tracking: Proof of Concept Report. Internal document.” .

Jennic JN-AN-1079. 2008. “Co-existence of IEEE 802.15.4 at 2.4 GHz, Application Note JN-AN-1079 v1.0.” `http://www.jennic.com`.

Jennic JN-UG-3041. N.d. “JN-UG-3041 JenNet Stack User Guide.” `http://www.jennic.com`.

Jennic JN5139. N.d. “Jennic JN5139 Datasheet.” `http://www.jennic.com/jennic_support/datasheets/jn5139_module_datasheet`.

Krishnan, P., AS Krishnakumar, W.H. Ju, C. Mallows and SN Gamt. 2004. A system for LEASE: Location estimation assisted by stationary emitters for indoor RF wireless networks. In *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies.* Vol. 2 IEEE pp. 1001–1011.

Liu, H., H. Darabi, P. Banerjee and J. Liu. 2007. “Survey of Wireless Indoor Positioning Techniques and Systems.” *Systems Man And Cybernetics, IEEE Trans. on* 37(6):1067.

Mao, G., B. Fidan and B. Anderson. 2007. “Wireless sensor network localization techniques.” *Computer Networks* 51(10):2529–2553.

*NXP PN65K Datasheet.* N.d. `http://www.advanide.com/datasheets/sfs_pn65k_rev1_3.pdf`.

*OceanServer OS4000-T User Manual.* N.d. `http://www.ocean-server.com/download/OS4000_Compass_Manual.pdf`.

Youssef, M.A., A. Agrawala, U. Shankar et al. 2003. WLAN location determination via clustering and probability distributions. In Proc. *IEEE Percom.* IEEE pp. 143–150.

Zensys Inc. 2007. “WLAN Interference to IEEE802.15.4, White Paper.” `http://www.srw-magazine.com/images/WhitePapers/WLAN%20Interference_FINAL.pdf`.