Power Saving in Wireless LAN Localization Systems

Power Saving in Wireless LAN Localization Systems

BY

Rafal G. Rzeczkowski, B.Sc.

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

AND THE SCHOOL OF GRADUATE STUDIES

OF MCMASTER UNIVERSITY

IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

Master of Applied Science

Master of Applied Science (2009)
(Electrical & Computer Engineering)

McMaster University
Hamilton, Ontario, Canada

TITLE: Power Saving in Wireless LAN Localization Systems

AUTHOR: Rafal G. Rzeczkowski
B.Sc. (Hons), Molecular Biology — McMaster University

SUPERVISOR: Dr. Terence D. Todd

NUMBER OF PAGES: xiv, 98

```
@MastersThesis{Rzeczkowski:SPLAN,
author={Rafal G. Rzeczkowski},
title={Power Saving in Wireless LAN Localization Systems},
language={English},
school={McMaster University},
address={Hamilton, Ontario, Canada},
year=2009}
```

Abstract

A localization system based on the Institute of Electrical and Electronics Engineers 802.11$^{\text{TM}}$ wireless local area network standard enables cost-effective localization measurements. It also integrates data dissemination within the system, unlike the popular NAVSTAR global positioning system. This thesis addresses two significant deficiencies in such a system: potentially *low localization accuracy* and *high power usage* of the communication infrastructure. A design for a low-cost solar powered localization augmentation node (SPLAN) is proposed to provide accuracy-increasing localization assistance service to localization tags—small battery powered units equipped with an IEEE 802.11 transceiver. Power saving network protocols for the SPLAN/tag architecture are developed to extend the tag battery lifetime and allow the SPLAN to be cost-effectively supplied by renewable energy sources such as solar energy.

A comprehensive power consumption model is developed for two complementary communication systems: *plain* tag and *smart* tag. Through its use, significant SPLAN energy conservation for low tag densities was demonstrated in the *plain tag system*, while preserving legacy compatibility. In the *smart tag system* even greater energy gains were obtained for both the SPLAN and the smart tag, up to maximum supportable tag densities. This was possible through the use of a custom communication protocol designed specifically for smart tag system power saving. Two innovative aspects of the protocol include timing-assisted scanning of management beacon packets and an optimized procedure for dissemination of raw localization data.

Acknowledgements

I would like to thank my supervisor **Dr. Terence D. Todd** for his support throughout the development process of my thesis. He has helped me in the planning stage to select an appropriate focus from my excessively diverse areas of interest, has later provided suggestions for the system behaviours, and has given me feedback regarding the evaluation of system components. Dr. Todd was also instrumental in taming most of my unorthodox approaches to the written expression of research ideas including the organization of material, the typographic layout and the usage of the English language. However, whatever errors and imperfections still remain are entirely my own.

I would also like to thank the other members of my thesis defence supervisory committee: **Dr. Ted Szymanski** and **Dr. Steve Hranilovic**, for reviewing my thesis. In addition, Dr. Szymanski deserves my thanks for the valuable insights into the area of network performance evaluation that he has shared with me in the CoE:4DN4 and the ECE:731 courses that he taught. Dr. Hranilovic has my gratitude for showing confidence in my former professional abilities, by inviting me at the beginning of my graduate study at McMaster University to an informal discussion of free space optics hardware and their Ethernet interfaces.

I extend my thanks to my laboratory colleagues, Dres. **Mohammed Smadi** and **Amir Sayegh** for their moral support and encouragements. The discussions that we had regarding the thesis structure, the significance of the results, and topics for exploration are likewise appreciated. Also appreciated are the comments from other members of the Department of Electrical and Computer Engineering who have attended this thesis's in-progress work presentations.

A special thanks goes to **Dr. Vernor Vinge**[1], a science fiction author who has provided an inspiration for this thesis through his entertaining and insightful books. In the "Deepness in the Sky," Dr. Vinge introduces *localizers*—power-aware smartdust devices utilizing wireless mesh networks for localization and surveillance data dissemination; they represent a highly sophisticated evolvement of the smart tag featured in this thesis. These enticing extrapolations of current technologies in science fiction writing motivate and encourage current research such as this.

List of Acronyms

ACK	Acknowledgement (frame)
AoA	Angle of Arrival
API	Application Programming Interface
BSS	Basic Service Set
BSSID	Basic Service Set Identifier
CCA	Clear Channel Assessment
CCK	Complementary Code Keying (modulation)
CCX	*Cisco Compatible Extensions*
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
CW	Contention Window
DBPSK	Differential Binary Phase Shift Keying (modulation)
DCF	Distributed Coordination Function
DHCP	Dynamic Host Configuration Protocol
DIFS	DCF Interframe Space
DQPSK	Differential Quadrature Phase Shift Keying (modulation)
DS	Distribution System
DSSS	Direct-Sequence Spread Spectrum (modulation)
ESS	Extended Service Set
GPS	Global Positioning System
IBSS	Independent Basic Service Set
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
LAN	Local Area Network
LBS	Location-Based Service
LI	*Listening Interval*
LOCP	*Cisco Location Protocol*

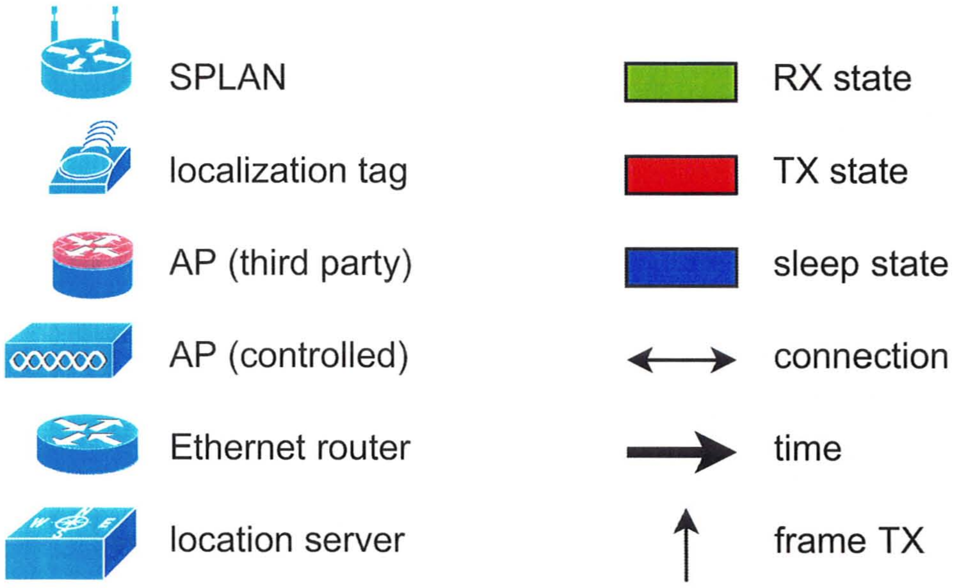
MIMO	Multiple-Input and Multiple-Output (antenna system)
NAM	*Network Allocation Map*
NAV	Network Allocation Vector
NIC	Network Interface Controller
NNSS	*Nearest Neighbour in Signal Space*
OFDM	Orthogonal Frequency-Division Multiplexing (modulation)
OSI	Open Systems Interconnection (network model layers)
PCF	Point Coordination Function
PDA	Personal Digital Assistant
PHY	Physical layer (OSI model)
PLCP	Physical Layer Convergence Protocol
PS	Power Saving (Energy Conservation)
PSAP	*Power Saving Access Point*
pTag	*plain-mode localization tag*
QoS	Quality of Service
RF	Radio Frequency
RFID	Radio-Frequency Identification
RSSI	Received Signal Strength Indication
RX	Receive/Reception
SBC	Single-Board Computer
SNMP	Simple Network Management Protocol
SPLAN	*Solar Powered Localization Augmentation Node*
SSID	Service Set Identifier
STA	Station (network node)
sTag	*smart-mode localization tag*
TCP	Transmission Control Protocol
ToA	Time of Arrival
TS	Time Slot
TX	Transmit/Transmission
UDP	User Datagram Protocol
VoIP	Voice over Internet Protocol
WDS	Wireless Distribution System
WLAN	Wireless LAN
WMN	Wireless Mesh Network

Contents

List of Tables

List of Figures

List of Figure Symbols



Chapter 1

Introduction

1.1 Localization Overview

Localization is the determination of the location (and implicitly the identity) of an entity. It is distinct from *mapping* in that the localized entity is expected to be mobile and hence the determination of its location is valid only transiently. Localization encompasses [→2.2.1] a variety of biological processes and computer technologies for a range of distance measurement scales. The focus of this thesis is small to medium-scale localization where the location data are electronically collected in a central repository for further processing.

Localization provides the foundation [8] for location-based services (LBS) which enable service users to find location-enabled objects. LBS applications [→A.1] which fit the focus of this thesis can be found in agriculture, robotics, health care, network security, physical security, emergency response, and public transport. The benefits obtained from localization in these applications centre on process automation in industrial systems and work flow optimization in commercial systems.

1.2 Localization Technologies

Among the variety of available localization technologies, the subset relevant to these applications [→A.1] is presented below. The overview is based on Reference [12] and includes a representative selection of systems that have received recent research attention.

Global Positioning System

The NAVSTAR GPS[13] is a mature and well known localization system with successful commercial implementations for navigation. A globally ubiquitous network of satellites orbiting Earth transmits telemetry to ground-based mobile receivers where location is computed by multilateration with accuracy of 3–15m.

The primary **limitation** of the GPS is its lack of inherent mechanism for dissemination of localization data to a central server for processing and visualization, while the addition of a GPS receiver to a localization device causes high energy drain due to the long satellite acquisition time per localization transaction. Reference [14] identifies a security problem with the GPS: no protection is offered against signal spoofing on the public (non-military) GPS data stream. Indoor operation of the GPS is limited because a GPS receiver must have a clear view of the sky; even outdoor operation may occasionally be erratic in cities due to occlusion and scattering of the GPS signals by tall buildings, as mentioned in Reference [15].

In fairness to recent advancements in GPS receiver technology, many of these problems have been mitigated to a large extent in the latest available chipsets. For example, the newest state of the art *u-blox 6* chipsets (by u-blox AG) announced[16] on September 2009 include excellent "-160dBmW tracking sensitivity" for reception of weak signals, and a new "Capture & Process technology" for instant location acquisition meant to enhance the host device's battery life in applications such as photo-geotagging. Featuring a "<0.2s" delay per position capture, the *Capture & Process* technology is also relevant for use in LBS-type localization systems where infrequent but quick location fixes are desirable.

Cellular Systems

While some mobile phones include a GPS receiver, the cellular network itself can be used for localization to minimize hardware costs. Localization is used to provide "Enhanced 9-1-1" service: the association of a user's phone number with the phone's physical location. It allows responders in *Public Safety Answering Points* to direct emergency response to the correct location.

The mobile phone's location can be determined by angle of arrival, time difference of arrival or pattern matching techniques based on the measurement of cellular signals. The main limitation of the cellular system is its relatively low accuracy of about 150–300m, making it unsuitable for many LBS applications.

Infrared Systems

The *Active Badge*[17] system from Olivetti Research Laboratory consists of small tags which transmit an infrared signal every 10s. A network of sensors installed throughout a building detects these transmissions and uses a proximity algorithm to estimate the user's location. Alternatively, a personal digital assistant (PDA) with an infrared transceiver may be used instead of a tag, as featured in the *ParcTab Ubiquitous Computing system*[18] from Xerox Palo Alto Research Center. Using the algorithms from the Active Badge system, a ParcTab user can be localized and can receive location-based notifications directly on the ParcTab device. Unfortunately, infrared-based systems such as the *Active Badge* or the *ParcTab* suffer from degraded performance due to direct sunlight and fluorescent lighting exposure, limiting their applicability to dark, indoor spaces. The need for deployment of special sensors increases the cost of such systems.

Ultrasonic Systems

The *Active Bat* system from AT&T uses a combination of radio frequency and ultrasound transmissions. A ceiling-mounted sensor emits an RF signal that activates a tag, prompting it to emit an ultrasonic chirp. Its time of flight is calculated in reference to the RF signal with a high accuracy of 9cm. However, the requirement for a large array of precisely positioned ceiling sensors limits the usefulness of this system. The *Cricket* system reverses transmitter and receiver locations compared to the *Active Bat*. Reception of a single beacon enables cell-proximity localization, while multiple beacons can be used in more precise lateration computation. The availability of location information is restricted to the tag, enabling privacy but excluding numerous LBS applications.

Pressure Sensing

The *Smart Floor* system from Georgia Tech uses cell-proximity algorithm to analyze data from floor-installed pressure sensors. The lack of electronic tagging precludes entity identification, while entities which do not interact with the floor can not be tracked at all. Combined with high installation and infrastructure costs, pressure sensing systems have limited practical uses.

Magnetic Tracking

Direct current magnetic-field pulses from a stationary antenna are captured by antennas mounted on the localized entity. The system offers excellent performance specifications of 1mm spatial resolution, 1ms time resolution, and 0.1° angular resolution, but a limited range of 1–3m. Such systems are useful for motion capture in computer animation, but scale poorly for use in LBS.

Radio-Frequency Identification

A radio-frequency identification [19] (RFID) system consists of RFID tags and RFID readers. The tags are small microchips with an antenna that transmit a unique serial number in response to interrogation by a RFID reader via RF. There are two main classes of RFID tags: passive and active. The passive tags obtain energy wirelessly from the reader and have an effective communication range from about half a meter to tens of meters, depending on their fixed communication frequency. The active tags are more bulky, and much more expensive. They contain a battery that extends their communication range to 100m or more.

The RFID system is promoted as a successor to the printed barcode system, or more generally as means to identify objects and people. While RFID systems are not designed for localization, some RFID-based localization systems have been proposed, such as the LANDMARC system featured in Reference [20]. In this system, a dense grid of reference tags (1 tag/m^2) is deployed in the localization area together with an array of readers supported by an IEEE 802.11 network for data dissemination. The tag's location is computed by a variation of a proximity algorithm, with an accuracy of about 1m. However, retrofitting an environment with such a high density of tags is impractical at best. Another problem is that passive RFID tags have a short interrogation range and require high RFID reader density, while active RFID tags face strong competition from IEEE 802.11 tags discussed next, due to the low cost of IEEE 802.11 commodity hardware.

IEEE 802.11 Network Localization

Localization using IEEE 802.11 networks—the focus of this thesis—is similar in principle to other systems. An IEEE 802.11 client device (e.g. a laptop or a PDA) or a dedicated IEEE 802.11 localization tag emits a beacon which is received and measured by infrastructure access points. An accuracy of several meters is achievable which makes IEEE 802.11 localization useful for many LBS applications, both indoor and outdoor.

As the IEEE 802.11 networking technology continues to gain popularity, the manufacturing economies of scale reduce the cost of client and infrastructure equipment. A localization system based on the IEEE 802.11 equipment can share in the cost savings, unlike other localization systems which require specialized and dedicated hardware. Yet additional hardware may be needed in other localization systems such as the GPS for dissemination of localization results, while the IEEE 802.11 networks accomplish both localization and dissemination through the same hardware. Reference [21] argues that "for large networks of very small, cheap, low power devices, practical considerations such as size, form factor, cost, and power constraints of the nodes preclude the use of GPS on all nodes" in favour of a "short-range radio frequency (RF) transceiver" such as the IEEE 802.11 transceiver in pTags and sTags considered in this thesis.

The first notable IEEE 802.11 localization system was developed in the year 2000 by Microsoft Research. The MS-RADAR[22] system based on RSSI measurement explored two approaches to localization computation: lateration (4.3m accuracy at 50% probability level) and pattern matching (3.0m accuracy at 50% probability level). Several commercial implementations were subsequently developed based on this research with limited communication protocol standardization. [→2.2.3]

1.3 Thesis Motivation: Issues in IEEE 802.11 Localization

Research issues in IEEE 802.11 localization addressed in this thesis are presented according to the framework developed in Reference [23] Sections V and VI.

System Cost

Lowering of costs increases the range of possible deployment scenarios for localization. *Equipment* cost which includes infrastructure and tag hardware costs is in part due to inefficiencies in data acquisition, communication and computation processes that lead to higher system requirements. System *installation* cost is incurred through routing of power and data, which is especially troublesome in outdoor installations. Ongoing *maintenance* cost, apart from general network administration, is due to the need for replacement of non-renewable tag batteries.

Localization Accuracy

To function correctly, a localization service must obtain accurate data from its localization system. While for a theft-prevention service it might be sufficient to know whether the localized device is still inside the premises, finding the same lost device requires at least room-level accuracy. The problem of low accuracy can be classified into *low average accuracy* (poor performance throughout the region of interest) and *low minimum accuracy* due to the existence of zones with no effective localization coverage.

Low average density of APs is the predominant cause of the low accuracy problem. However, even with high density, a suboptimal AP placement can induce errors in a localization solver as shown in Reference [24]. This placement, referred to in GPS terminology as *Geometric Dilution of Precision*, is exemplified by linear, non-enclosed AP arrangements. Such arrangements can arise in IEEE 802.11 localization due to data network coverage and localization services having different placement optimality, e.g. an indoor network optimized for data service but not localization. Outdoors, the necessary proximity to sources of electricity and wired Ethernet connections can cause suboptimal installation patterns. Finally, a high node mobility can cause localization computations to become quickly outdated and hence inaccurate.

A popular research direction is the development of more sophisticated estimation algorithms which increase accuracy within the limits of IEEE 802.11 hardware. Beyond these limits, the use a different measurement technique (e.g. ToA) on modified hardware or the use of an additional physical layer (e.g. a magnetic exciter chokepoint) can increase accuracy with the trade-off of increased system costs. Another trade-off is increased power usage versus an increase in the frequency of localization measurements, to minimize errors due to node movement.

Network Impact

IEEE 802.11 localization enjoys significant costs saving by *sharing* IEEE 802.11 infrastructure previously deployed for traditional uses such as Internet access and multimedia communications. If the localization communication system monopolizes network resources, (examples cited in Reference [23]) the cost benefit is reduced or disappears as more equipment has to be purchased to support the original services. Even if cost is not considered, the radio spectrum is a limited resource especially in the IEEE 802.11 2.4 GHz band.

Security and Privacy

In some LBS applications it is desirable to maintain the location or identity of the localized entity confidential. In a trusted environment this might be realized through encryption of localization communication packets. However, when the user does not trust the infrastructure, it might be preferable to complete the localization process on the client device. Another issue is the alteration of localization data; it should be difficult to engineer an attack that would cause the localization system to misreport the position of an entity. This is especially important for LBS uses such as theft prevention or location-based augmentation of network access security.

Environment Type

The most general and useful classification of the environment type for IEEE 802.11 localization is an indoor versus an outdoor environment. The traditional focus has been on indoor localization, due to the predominance of indoor IEEE 802.11 network deployments and strong competition with the GPS outdoors. However, as IEEE 802.11 networks continue to grow in popularity, their outdoor deployment increases accordingly. Hence, there is an increased interest into developing outdoor systems such as those in References [25, 26]. The target environment type influences the design choices for the localization system, in particular the choice of localization algorithm used (e.g. pattern matching versus environment simulation). Additional considerations must be given to the availability of power, which is generally more scarce outdoors.

Localization Protocol

The plain tag IEEE 802.11 localization protocol is inherently simple: a pTag periodically emits a chirp packet, a localization infrastructure AP receives it and forwards it to a localization server along with its measured received signal strength indication (RSSI). An RSSI measurement of any standard transmission between a STA and an AP can also be used.

The notable areas where localization protocol issues have been investigated are in location privacy and physical layer integration. Reference [27] introduces the concept of an *opportunistic silent period* to be used along with client identity (MAC/IP address) changes and transmit power control, to give users an option to opt-out of a localization system. Reference [28] proposes a localization protocol based on both PCF and DCF to enable the operation of ToA and AoA localization methods.

Cisco Systems has promoted the standardization of plain tag commercial systems via its "Cisco Compatible Extensions program for Wi-Fi tags" which defines the protocol that the tag must follow to be compatible with "Cisco Unified Wireless Network." Protocol standardization for smart tag systems is virtually non-existent though. Commercial systems define their own methods for smart tags to interface with a localization server, while relying on IEEE 802.11 standards for interfacing of tags with the network.

While localization tags implement rudimentary power saving (PS) schemes to permit on-battery operation, no such provisions are being considered for localization infrastructure components. Traditionally, these have been placed indoors where power is readily available, but outdoor localization systems may use costly renewable power sources. Having PS schemes available for localization infrastructure components would help to reduce these costs.

1.4 Thesis Scope and Contributions

The subject of this thesis is the communication protocol for IEEE 802.11 localization of both plain and smart tags. An AP-like device called a SPLAN is proposed that either integrates within an existing plain tag system to provide localization accuracy enhancement, or serves as a sole localization service provider in a *de novo* design of a smart tag system.

The main feature of the SPLAN and its communication protocol is PS which is realized by taking advantage of opportunistic network quiet periods in the plain tag system or explicitly scheduled PS intervals in the smart tag system. The general SPLAN design objectives are the improvement of localization *accuracy* through assistance functionality, lowering of power consumption to minimize photovoltaic component size, ensuring compatibility with existing systems, and ensuring system reliability despite the necessary increases in system and protocol complexities.

In the context of the current IEEE 802.11 localization research objectives [→1.3], the proposed SPLAN system makes advances in the areas described below.

System Cost

Unlike conventional commercial designs where localization is an added service to a fully featured and expensive AP, the SPLAN is dedicated to just localization: its hardware and software costs are smaller. Through the design of PS algorithms, the cost of providing the SPLAN with solar power can be reasonable too, while the ongoing system maintenance costs are addressed by the attention given to the tag-centric PS algorithms to increase the tag's battery lifetime.

Localization Accuracy

The primary goal of increasing localization accuracy is realized through the deployment of indoor/outdoor SPLANs which increase the localization infrastructure *density*. The *placement* of localization infrastructure components is traditionally constrained by power and data availabilities leading to geometrical arrangements unsuitable for localization (e.g. along city street light poles). A power-conserving solar-powered mesh-networked SPLAN has a greater installation freedom than a conventional AP, perfect for optimizing specific areas of low localization accuracy. Areas with existing third-party APs benefit from the proposed system too, through emphasis on smart tag development. Unlike a conventional plain tag, a smart tag can take advantage of any APs deployed in an area, not only those specifically prepared for localization. Finally, the localization *frequency* can be increased in the proposed system within the existing battery budget, leading to better accuracy in high mobility situations. Typical commercial designs in plain tag systems waste power on the transmission of multiple chirp packets (a primitive form of error correction) while non-standardized smart tags suffer power drain through constant AP association, address acquisition through DHCP and TCP/IP communication overhead. The proposed sTag system reduces these communication algorithm inefficiencies so there is more battery power left to schedule more frequent localization updates instead.

Network Impact

The SPLAN design minimizes the impact of a localization service on the underlying IEEE 802.11 network through a novel smart tag communication protocol design. Conventional pTags broadcast their chirp packets multiple times at the lowest data rate so they can be received by as many localization-aware APs as possible. Instead, in the proposed smart tag system, the sTag selects a SPLAN with the highest signal strength, allowing for transmission at higher, channel-conserving data rates. An explicitly set MAC address allows hardware-based packet filtering and an explicit acknowledgement (ACK) system enables a localization packet to be usually transmitted only once.

Security and Privacy

Neither security nor privacy issues related to localization are considered in this thesis.

Environment Type

The proposed SPLAN PS algorithms make the use of solar power feasible, providing the biggest benefit for outdoor installations. However, there is nothing inherent in the design that limits SPLAN use to the outdoors. The remaining benefits explained in other sections are still applicable to indoor use, where an indoor-type SPLAN could have a form factor of an over-sized power adaptor, making its deployment exceedingly easy and inexpensive. In fact, the proposed SPLAN system is ideally suited to provide localization in a mixed indoor/outdoor setting such as a university campus. There, a GPS-tag would not work indoors necessitating the use of an IEEE 802.11 tag and indoor-type SPLANs, while the same IEEE 802.11 tag would continue to work outdoors with outdoor-type SPLANs, saving on the cost and the energy requirements of a GPS receiver.

Protocol

The primary contribution to the *plain tag system* is the design of SPLAN PS behaviour compatible with the protocol used most commonly in existing deployments. Firmware modifications for a pTag are also proposed to allow for quicker pTag synchronization, resulting in higher localization assistance levels. In the *smart tag system* the proposed protocol design paves the way for standardization efforts to produce a universally compatible IEEE 802.11 sTag system. Through the use of DHCP option-like packet format, an ease of implementation combined with future extensibility was achieved.

1.5 Thesis Organization

This document is organized according to the following outline.

Chapter 1 introduces *localization* and discusses representative localization systems relevant to the scope of the thesis. Current research issues are highlighted and an analysis of the thesis's treatment of these issues is provided.

Chapter 2 opens with background information on IEEE 802.11 networks. The general system architecture is presented followed by a focus on detailed processes relevant to this thesis. They include *CSMA/CA* for optimized sTag data uploading, *scanning* for assisted sTag data collection and *infrastructure power saving*—a pervasive consideration throughout all the thesis sections.

In the second Background Section, localization systems are introduced. A new process-oriented localization taxonomy is developed that also explains the functioning of a generalized localization system. Foundations of localization algorithms are discussed for both the conversion of raw measured data to geometric constraints and the integration of measurements from multiple sensors. Legacy system implementations for both plain and smart tag systems are presented to demonstrate the existing lack of localization protocol advancement.

Chapter 3 presents two novel localization system designs for plain and smart tags. The systems' applicability to practical deployment scenarios is considered. System behaviour and protocol communication details are presented for the localization operational phase and first establishment of tag/SPLAN contact: *synchronization* of pTags and *registration* of sTags.

Chapter 4 develops system models for power consumption of SPLANs and tags. This entails analysis of representative hardware power consumption and switching latency data as well as durations of states in the proposed designs. The limitation of localization assistance in the plain tag system due to tag mobility is also modelled.

Chapter 5 uses the models from Chapter 4 to illustrate the effect of varying system parameters, most commonly n—the number of tags in the system—on SPLAN and tag power saving. Implications of the results for practical system performance are discussed.

Chapter 6 summarizes the thesis work and highlights fundamental conclusions.

Chapter 7 suggest future research directions and presents specific ideas extending the work presented in this thesis.

Appendix A contains additional technical data for existing tag systems and localization system applications.

Chapter 2

Background

2.1 IEEE 802.11 Networks

The most successful standard for implementation of wireless local area networks is the Institute of Electrical and Electronics Engineers (IEEE) 802.11$^{\text{TM}}$ standard[29]. It includes a family of standards (Table 2.1) which share a common MAC protocol but use different physical layers.

Table 2.1: IEEE 802.11 standards family comparison

IEEE standard	release year	operating frequency (GHz)	modulation	data rates (Mbit/s)
802.11-1997	1997	2.4	DBPSK, DQPSK	1, 2
802.11a	1999	5	OFDM	6–54
802.11b	1999	2.4	CCK	5.5, 11
802.11g	2003	2.4	OFDM	6–54
802.11n	2009[1]	2.4, 5	OFDM	6.5–600

The original IEEE 802.11-1997 standard introduced three physical layer (PHY) technologies: diffuse infrared (1Mbit/s), FHSS, and DSSS (both at 1 and 2Mbit/s). Only the DSSS PHY was continued in the IEEE 802.11b standard, which also added 5.5 and 11Mbit/s data rates. Even faster data rates were introduced in the IEEE 802.11g standard, but the need to maintain legacy compatibility prevented the realization of full theoretical performance. The IEEE 802.11a standard does not sacrifice performance, as it operates in the 5 GHz frequency band which is unused by devices compliant with the previous standards. The use of a different frequency also helped to reduce interference problems and allowed the use of more channels to share between co-located

[1] The IEEE 802.11n standard is expected to be finalized in December 2009

IEEE 802.11 networks. However, the IEEE 802.11a standard is primarily meant for indoor use due to greater RF signal attenuation at higher frequencies. The IEEE 802.11n standard is yet to be ratified, but already has significant product deployment based on draft standard versions. It uses a multiple-input and multiple-output antenna system (MIMO) to boost data rates and increase the effective network coverage area.

2.1.1 IEEE 802.11 Network Architecture

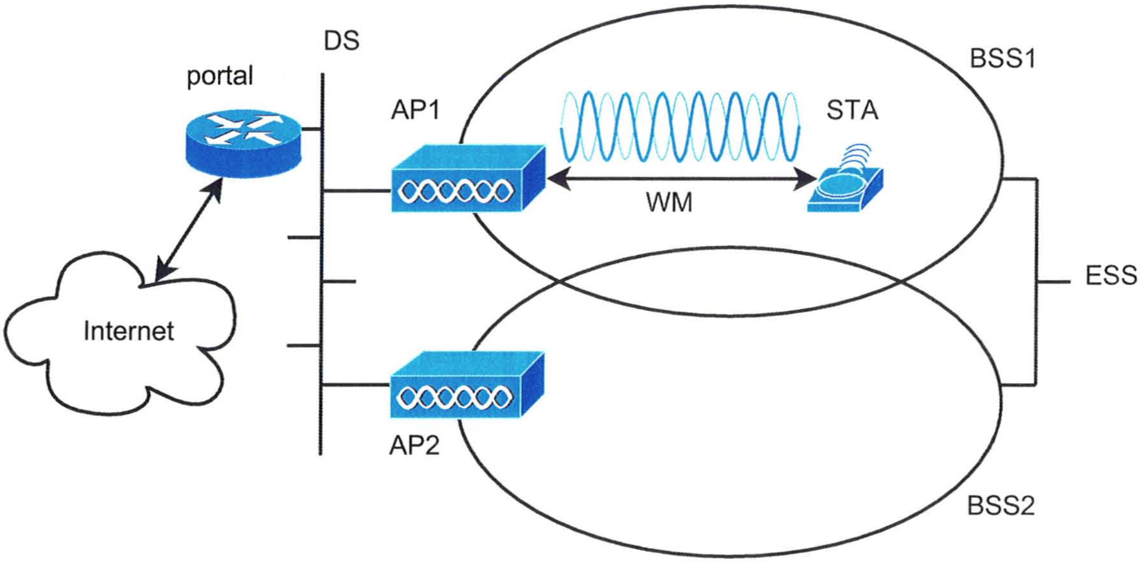


Figure 2.1: IEEE 802.11 network architecture and components

2.1.1.1 Network Components

The components of an IEEE 802.11 network are illustrated in Figure 2.1 and are discussed below.

Station

A STA is a device with two IEEE 802.11 compliant components: a PHY module ("the radio") which is responsible for modulating/demodulating data from/to the WM and a MAC module which implements the IEEE 802.11 rules for accessing the WM. The typical implementation of an IEEE 802.11 STA is a an IEEE 802.11 wireless network interface card found in computing devices such as laptop PCs, desktop PCs, PDAs, and IEEE 802.11 localization tags.

Access Point

APs are enhanced STAs that form the network infrastructure; they provide management functionality such as the broadcast of periodic management beacons that advertise the network and its capabilities. APs also bridge packets between the WM and the DS to provide STAs with access to other networks such as the Internet.

Wireless Medium

A WM refers to the physical means that the PHYs use to exchange data. Discounting the now obsolete infrared medium, radio frequency communication in the 2.4 GHz and 5 GHz bands through the free space and physical obstacles between PHYs is the WM of choice.

Distribution System

A DS is an AP component that enables the distribution of data packets throughout the network. In particular, direct radio communication between STAs in an infrastructure BSS is prohibited; instead, packets are sent to the AP, routed through the DS (which in this case is just a component of the AP's software) and sent to other STAs over the same radio interface. Packets to/from wired networks travel through the AP's *portal*-type DS which is usually implemented as an Ethernet connection to an Internet router. Communication between STAs associated with different APs (or a portal on a non-associated AP) is through the AP/AP-type DS, which might be implemented as an Ethernet LAN between two APs or as a wireless distribution system (WDS). A WDS provides DS-equivalent functionality via the WM. A dedicated radio may be used for this *back-haul* communication for performance reasons or a single radio may be time-shared between STA communication and back-haul services. WDS is the foundation for wireless mesh networks (WMNs).

2.1.1.2 BSS Types and ESS Mobility

A basic service set (BSS), as illustrated in Figure 2.1, is a group of STAs that form a wireless network. The STAs can be self-organized as an independent BSS (IBSS), commonly called an *ad hoc* network. In this network STAs communicate directly with each other without the help of an AP and without access external to the IBSS. Typically, this type of network is temporary and involves only a few STAs, as each must be within the radio range of each other. An IBSS is not discussed further in this thesis, since an *infrastructure BSS* is the most common configuration for an IEEE 802.11 network. In this network a central AP provides network services to STAs in its vicinity; to obtain service a STA must be *associated* with that AP. The association establishes a STA/AP mapping, e.g. to support an Open Systems Interconnection layer 2 (OSI L2) packet forwarding service based on the STA's MAC address.

Multiple BSSs may be organized into an extended service set (ESS) identified by a service set identifier (SSID) as shown in Figure 2.1. The ESS forms a single OSI L2 domain from the point of view of a STA; packets destined to STAs associated with another AP in the same ESS are transparently forwarded through the AP/AP-type DS. As a STA moves throughout the coverage area of the ESS, it changes its AP association (*roaming*) but retains its higher level network configuration such as its IP address. This allows the STA to maintain its network sessions such as real time voice over Internet protocol (VoIP) streams, provided that the associated AP/AP *hand-off* latency is acceptably low. [→2.1.3]

2.1.2 CSMA/CA

The IEEE 802.11 standard includes support for the *point coordination function* (PCF)—a contention free MAC service. The infrastructure AP's management beacon designates part of the channel time as a contention-free period during which the AP administers all traffic exclusively. The AP maintains a list of associated STAs participating in the PCF service. Each STA on the list receives a *CF-Poll* frame which gives it the right to transmit a data frame and a *CF-ACK* frame in response. The AP can control STA polling order and prioritize individual STAs according to their real-time requirements.

Since PCF is an optional IEEE 802.11 feature and is seldom implemented, it will not be discussed further in favour of the *distributed coordination function* (DCF) which is the primary method of frame transmission (TX) coordination in IEEE 802.11 networks (Reference [29] Section 9.2). DCF is an implementation of the generic carrier sense multiple access with collision avoidance (CSMA/CA) approach; a STA with one or more pending packets must perform a CCA [→2.1.2.1] prior to transmission. If the CCA finds the WM free for the duration of a DCF interframe space (DIFS), frame TX begins. If the WM is busy instead, the STA *defers* access and begins a *backoff* procedure which will allow it to TX later. Successful reception (RX) of a frame is confirmed by TX of an ACK packet by the recipient, whereas unacknowledged frames are retransmitted to provide OSI L2 reliability.

2.1.2.1 Clear Channel Assessment

Clear channel assessment (CCA) is the procedure used by a STA to confirm availability of the WM, so that frame TX can begin or the backoff timer [→2.1.2.2] may be decremented. In wired networks the CCA is simple since all transmissions can be reliably detected. The challenges faced by STAs in wireless networks include significant *attenuation* due to free space loss and non-line-of-sight conditions, *interference* from other devices, and *half-duplex* radio operation which precludes direct collision detection. To address these challenges, an IEEE 802.11 STA has several methods at its disposal to perform CCA; they can be classified into *physical* and *virtual* methods. Physical[10] CCA methods (summarized in Table 2.2) include *energy detection*—the use of a radio transceiver to measure energy readings centred at the expected carrier frequency. Energy detection requires little STA power but is unreliable since IEEE 802.11 transceivers use wideband signals that have a low spectral density, sometimes not much above the WM noise floor. To increase the effective signal to noise ratio in CCA detection, *carrier sensing*—a coherent detection technique—may be used. The typical approach is to focus on demodulating the frame's preamble which contains easy to detect repeating sequence of known symbols transmitted at a known speed and modulation. The disadvantages of carrier sensing are high power use for signal processing and a long decision delay—potentially the duration of an entire packet transmitted at the lowest data rate. While theoretically it is possible to eliminate this delay by scanning for symbols inside the packets, multiple simultaneous filters must be employed which consume even more power.

Table 2.2: Classification[10] of physical CCA methods

	Carrier sensing		Energy
	Preamble detection	Decorrelation based	detection
Complexity/power consumption	Moderate	High	Low
Reliability	High	Moderate	Low
Position in packet	Preamble	Anywhere	Anywhere

In some situations the physical CCA methods are unreliable, so the IEEE 802.11 standard includes a virtual CCA method implemented through the network allocation vector (NAV). To support NAV, the IEEE 802.11 frame header includes a 2-byte *duration* field, of which 15 bits are set by a transmitting STA to the number of microseconds it expects the WM to be busy. For a typical TX the NAV is set to the combined durations of the data frame, the ACK frame, and all the relevant inter frame spacings. Other STAs are expected to continuously monitor the WM and decode MAC headers from all packets to update the STA's internal counter—the NAV. A non-zero NAV value indicates a busy WM to the CCA procedure.

2.1.2.2 Random Backoff

The random backoff procedure (Reference [29] Section 9.2.5) is based on the manipulation of two variables: the CW_i (introduced here for convenience) and the *Backoff Timer*, expressed here in time slot (TS) units for convenience. The CW_i (Contention Window index) determines the size of the contention window (CW), computed as $CW \leftarrow 2^{CW_i} - 1$ and used as the maximum value of a distribution from which the *Backoff Timer* integer is randomly selected, i.e. *Backoff Timer* is drawn from the uniform distribution [0,CW]. The *Backoff Timer* indicates how many TSs are left before a TX can commence.

When the random backoff procedure begins (e.g. in response to a pending transmission encountering a busy channel), the CW_i is set such that $2^{CW_i} - 1 = CW_{min}$. The CW_{min} is a PHY-specific parameter that specifies the minimum allowed CW size. As long as the WM remains busy, the *Backoff Timer* value remains unchanged. However, after the WM has been idle for the DIFS period, the *Backoff Timer* is decremented by one for each TS duration that passes while the WM remains idle. When the *Backoff Timer* reaches zero, the STA initiates frame TX. Alternatively, if WM activity is detected, the *Backoff Timer* countdown is suspended.

If the *Backoff Timer* of two or more STAs reaches zero in the same TS, simultaneous TX occurs resulting in a collision which is ascertained by the lack of an ACK response frame. The CW_i variable is incremented by one, a new CW is calculated and a new random *Backoff Timer* is selected from an enlarged distribution which is designed to reduce the chance of a subsequent collision. Subsequent collisions cause further increases in CW_i until $2^{CW_i} - 1 = CW_{max}$ and CW_i stops increasing. The CW_{max} is another PHY-specific parameter analogous to the CW_{min} parameter. The standard values for CW_{min} and CW_{max} are 31 and 1023 respectively for both the DS and OFDM PHYs; these PHYs support the vast majority of IEEE 802.11 networks currently deployed.

2.1.2.3 Research Issues in DCF

Because the CSMA/CA algorithm is critical[30] to the performance of wireless networks, much research effort has been directed at improving its performance.

The authors of Reference [31] note that the IEEE 802.11 DCF exhibits unfair behaviour when STAs use different bit rates (e.g. due to signal-to-noise constraints). STAs with degraded bit rates cause significant performance losses for STAs operating at nominal rates. The proposed solution is adjustment of the CW_{min} parameter in accordance with the STA's bit rate to penalize slower STAs by means of a higher CW_{min}. This scheme results in an improved network throughput.

Reference [32] proposes that a STA should listen to all WM transmissions and count the number of idle TSs. That result is compared to the theoretically estimated value for the proposed scheme. If the number of idle TSs is too low, the CW is increased, otherwise CW is decreased. The *number of idle TSs* used in this reference as an indicator of current collision levels is superior to the DCF's collision estimation mechanism based on the lack of an ACK frame following a transmission. The later can occur not only due to a collision but also as a result of a failed transmission with the data bit rate set too high for the current WM conditions.

Reference [33] addresses the issue of poor DCF performance in conditions of heavy network loading. The CW is adjusted based on the estimated number of STAs competing for WM access, according to a proposed optimization function. Anther contribution is the proposal of two algorithms for estimating the number of STAs from the observed collisions.

These examples demonstrate the general framework used for DCF optimization. Either the CW-influencing fixed parameters are adjusted[31], the CW adjustment is performed via an alternate algorithm[32], or the CW is calculated directly to achieve optimality[33]. These adjustments are based on an enhanced knowledge of a network state not taken into account in the original DCF, e.g. the TX rate or observed collisions. This thesis adapts these ideas to the unique circumstances faced by sTags in the IEEE 802.11 localization. A simple CW_{max} adjustment is evaluated as in Reference [31], but the main contribution is a direct optimized CW calculation [→4.4.3.2], similar to that in Reference [33]. The source of additional data for the optimization procedure (number of STAs competing on the channel), is a SPLAN-provided estimate of currently registered number of sTags. [→3.2.6]

2.1.3 Beacon Scanning

2.1.3.1 Standard Scanning Procedure

Scanning is a process of enumerating accessible APs, used in IEEE 802.11 localization by smart tags to collect RSSI measurements for location computation [→3.2.5]. In IEEE 802.11 networks scanning is used by STAs *before initial association* to enumerate all available networks and present them to the STA's user, or to confirm the existence and discover operational parameters of a network identified by a pre-configured name. While associated, a STA may continue to scan to discover other compatible APs with better connectivity as part of a *roaming* process.

The IEEE 802.11 standard defines two scanning methods (Reference [29] Section 11.1.3): passive and active. In the *active scanning* process, a STA tunes in sequence to each channel selected for scanning and sends a *Probe Request* packet to the MAC broadcast address. The packet may contain either a wildcard or specific AP identifiers that determine if a response is sought from all or a specific AP. If there are no responses received for a duration of *MinChannelTime,* the STA concludes that the channel is empty and begins to scan the next one. Otherwise, *Probe Response* packets are collected for the duration of *MaxChannelTime.* In some scenarios, the active scanning mode is prohibited due to regulatory restrictions and *passive scanning* must be used instead. A STA conducting a passive scan tunes to each channel selected for scanning and listens for a maximum of *MaxChannelTime* for any AP management beacons.

Other details of the scanning procedure, such as the choices of the timer values, are left to implementations. The typical values for $MinChannelTime = 10$ms and $MaxChannelTime = 100$ms were obtained from Chapter 5 of Reference [34]. The choice of the roaming trigger used for scan initiation is also left to implementations. Simpler IEEE 802.11 implementations are content with performing a reactive scan in response to dropping RSSI levels, whereas more advanced algorithms may favour pro-active partial scans that gradually enhance a STA's knowledge of its RF environment without much impact on its communications, enabling faster subsequent roaming.

2.1.3.2 Research Issues in Active Scanning

Improvements to the IEEE 802.11 scanning process are proposed in literature as a means of handoff latency reduction which is desirable for real-time applications such as VoIP. Reference [35] claims that possibility of a loss of a *Probe Request* packet due to channel collision in a noisy environment introduces a significant unreliability in the active scanning process. In response, a *reliable active scanning* process is proposed where the STA performs traffic detection following the transmission of the *Probe Request,* and immediately retransmits if nothing is forthcoming after the expected time. Reference [36] proposes a standards-compliant optimization of the *MinChannelTime* and *MaxChannelTime* parameters of the active scanning process.

In general, enhancements to the *active scanning* mode are proposed, as passive scanning is considered too slow for purposes of a real-time hand-off. Passive scanning performance is explicitly evaluated in Reference [37] where scanning time per channel is equated with the AP beacon interval (usually 100ms). Reduction of the passive scan duration is proposed by the lowering of the beacon interval on all APs. Unfortunately, increasing beacon frequency reduces channel capacity; it was found that 60ms inter-beacon time was a reasonable compromise for channel usage, but was still unacceptable as a hand-off latency for real-time applications.

2.1.3.3 IEEE 802.11k Standard and Passive Scanning

While it may seem that active scanning is preferable for reasons of performance, and passive scanning is a dead-end for improvement, the passive scanning approach was chosen in this thesis nevertheless [→3.2.5]. A novel performance optimization specifically designed for use in localization allowed the passive scanning performance to approach that of optimized active scanning.

The featured optimization is an adaptation of the IEEE 802.11k standard[38]: "Radio Resource Measurement of Wireless LANs." The standard's goal is to "extend the capability, reliability, and maintainability of WLANs" by providing means for STAs and APs to enhance their understanding of their WM characteristics. This is achieved by defining various measurement types that a STA or an AP can perform, e.g. the "active channel scan" that was discussed previously, or a "Noise Histogram," which is a new feature of the standard. The standard's other component is the dissemination of these measurements: a STA can self-initiate and conduct a measurement locally (as in the plain IEEE 802.11 standard) but it can also ask another STA or an AP for measurements local to their environments. IEEE 802.11k measurements relevant to this thesis are shown below.

The **beacon measurement** (Reference [38] Section 5.2.7.1) instructs a STA to capture AP beacons (or equivalent *Probe Response* packets) on a particular channel. The scan type can be passive, active, or *null* where previously cached information is reported. The beacon report contains data that include the AP's basic service set identifier (BSSID) and the AP's operating channels. However, no specific beacon timing information is provided.

The **neighbour report measurement** (Reference [38] Section 5.2.7.9) instructs an AP to provide data about neighbouring APs, including information from the *dot11RRMNeighborReportTable* portion of the AP's *Management Information Base*. Some of the variables reported are the BSSID, the channel number, and the PHY type. Unlike the *beacon* report, the *neighbour* report does not define the acquisition method used to obtain the data (Reference [38] Section 11.10.9), but instead suggests that it might be obtained from associated STAs or directly from other APs via a DS.

The **location measurement** (Reference [38] Section 5.2.7.7) allows a STA to request its own location or ask another STA for its location. The response includes the latitude, the longitude, the altitude, and an optional azimuth. This measurement type is actually unrelated to IEEE 802.11 localization systems, as it assumes that the location is already available; a STA—such as a laptop equipped with a GPS receiver—may respond to a location measurement request with valid data. In the context of an IEEE 802.11 localization system, a STA that has been localized by a central server (through RSSI readings acquired separately), may ask an AP for its own location.

Novel uses of the IEEE 802.11k measurements are demonstrated in Reference [39] where the authors propose that an AP requests both the *location* and *beacon* measurements from its associated STAs. The resulting data are used to build a map of AP coverage useful for location-assisted handovers. Roaming can also be expedited by the use of *neighbour* reports. These provide a STA with ready-to-use database of APs in the vicinity without requiring time-consuming local scans. Reference [40] considers selecting the most optimal AP transition candidates from such reports.

In general, these examples show that significant benefit can be obtained from obtaining third-party scan reports. In this thesis, the sTags are assisted by SPLAN scan reports [→3.2.5]. However, the reports are not used directly, but to enhance the functioning of sTag's own scan. The sTag needs to measure RSSI at its own location by conducting its own scan which is the cornerstone of IEEE 802.11 localization. The SPLAN-provided report eliminates unused channels which is a technique suggested in Reference [37] Section IV. In addition, detailed beacon timing information (unavailable in the IEEE 802.11k standard) is used to schedule radio tuning to the appropriate channel at the exact moment when a beacon TX is expected.

2.1.4 Power Saving

The behaviour of STA PS is specified in the IEEE 802.11 standard (Reference [29] Section 11.2.1). A STA in a PS mode wakes up periodically to receive a traffic indicator map from its AP. The bitmap indicates whether the AP has any buffered packets for the STA. If so, the STA sends a *PS-Poll* packet to initiate transfer of a buffered packet. This PS procedure assumes that the AP remains continuously operational.

Reference [41] provides a comprehensive survey of the corresponding need for PS in the network infrastructure. An increase in adoption of solar power WMNs is noted, as this combination provides for tether-less operation that makes data and power connections unnecessary. In turn, tether-less operation can reduce installation costs or permit AP installation in an otherwise inaccessible location. The drawback of solar power is the high cost of solar panel and battery components; an undersized photovoltaic system can cause operational outages.

The outages can be reduced by correctly sizing the photovoltaic component to support a system with a given communication profile, as demonstrated in Reference [42] where meteorological data are used to simulate a node's energy input. Combined with the node's estimated energy drain, an optimal battery capacity and panel size curve is constructed for the desired outage probability. A control mechanism to deal with solar insolation fluctuations is also proposed.

Even a properly provisioned photovoltaic system may still cost too much, so energy conservation protocols are needed. Reference [43] introduces the concept of a power saving access point (PSAP), which extends PS capabilities to an IEEE 802.11 AP. The PSAP's function is to provide coverage service for nearby STAs. It does that by splitting the IEEE 802.11 beacon period into three intervals: a *contention* interval used for communication with nearby STAs on the PSAP's home channel, a *relay* interval to relay STA's data to upstream APs on the PSAP's relay channel, and a *sleep* interval to turn off the radio and conserve power. Compatibility with legacy STAs is maintained by the AP pretending to act as a PCF point coordinator for the duration of the sleep and the relay periods. As STAs update their NAVs to indicate service unavailability during PCF, they are prevented from transmitting via DCF when the PSAP is sleeping or relaying.

This thesis adopts the central idea of infrastructure PS from Reference [43] and applies it to the IEEE 802.11 localization system where the only traditional concern has been tag PS. An entity called a SPLAN analogous to a PSAP is introduced. The SPLAN's potential for PS is due to the use of sleep periods [→3.1.3] that maintain compatibility with the legacy pTag system. The notable differences lie in the scheduling scale: minutes between tag chirps versus 100ms between successive IEEE 802.11 beacons. The PSAP-type NAV blocking is not used to force tag transmissions into a suitable interval, as that would affect other non-tag STAs. Rather, the predictability of tag transmissions is exploited to create adaptive listening intervals (LIs).

Whereas Reference [43] provided compatibility with standard IEEE 802.11 STAs (just as the SPLAN is compatible with legacy pTags), further PS can be realized if STAs are explicitly aware of AP PS. Reference [44] introduces the concept of a network allocation map (NAM)—a generalization of a NAV—which specifies time periods when the PSAP is sleeping or relaying traffic. STAs recognize these NAMs and adjust their activities accordingly. The SPLAN in a smart tag system uses a similar but inverted encoding scheme which assumes that the SPLAN is sleeping all the time and its activity cycles are specified instead [→3.2.3.2]. Unlike the NAM specification, only the LI start time is specified, while its duration is sized as needed.

In Reference [44] the NAMs feature an *M-Boundary* that can be dynamically adjusted to meet current traffic demand. An *F-Boundary* is provided to preserve synchronization with STAs which were in PS mode and could not be notified of the changes. This optimization enables the STAs to bypass waiting for the next beacon. The key elements of this scheme are adopted in the SPLAN registration beacon [→3.2.3.2], established once and valid throughout the entire network area and for the lifetime of the network. It provides a guaranteed LI for sTags to support mobility transitions, which is analogous in function to the *F-Boundary*. Similarly, the SPLAN standard LIs are scheduled [→3.2.3.3] to support a dynamically changing tag population, just as the *M-Boundaries* are moved to support different traffic demands.

Quality of service (QoS) extensions to the IEEE 802.11 infrastructure PS are discussed in Reference [45]. The AP sleep patterns are scheduled to satisfy the requirements of currently serviced traffic flows. A connection admission control scheme is used by the AP to verify the availability of sufficient resources for supporting a new STA and to manage its power saving requirements.

QoS in the context of IEEE 802.11 localization is not well defined, so a clarification is provided presently that applies concepts from network QoS to localization QoS. The primary *quality* attributes of a localization service are the localization *frequency* and *accuracy*. The former is related to the interval between chirps (pTag) or localization beacons (smart SPLAN), while the later is primarily influenced by the number of APs that can hear a pTag TX or the number of SPLANs that emit a beacon.

There is no QoS support in the pTag system: SPLANs synchronize to pTags on a best-effort basis while service suffers as a result of pTag mobility [→5.1.3]. sTags in the smart system do support QoS: they send a registration request message [→3.2.3.2] with the required localization interval, which is a part of the localization QoS specification. The smart SPLAN employs a simple *connection admission control* scheme that limits admissions if capacity is exhausted. It also schedules the LIs to provide reasonable power usage on both sTags and the SPLAN [→5.2.3.3]. Some other interesting aspects of localization QoS not considered in this thesis are discussed in Section [→7.6].

2.2 Localization

2.2.1 Localization Taxonomy and Generic Localization Process

Reference [12] proposes a useful localization taxonomy, but fails to order the classifiers. This is done below in a novel process-oriented taxonomy for localization systems. Apart from providing taxonomic classifiers, this scheme also provides a discussion of a *generalized localization process*. In the provided framework, localization systems can be compared to each other to expose their functional limitations while cross-system integration and functionality adaptations can be proposed.

Origination

Localization is initiated by the introduction of a wave signal into an environment either *actively* (explicitly for localization) or *passively* (as part of a naturally occurring process). A sonar "ping" emitted by a submarine transducer or a "chirp" packet emitted by a pTag are examples of an *active* origination, while surface ship noise detected by passive sonar or a standard IEEE 802.11 AP management beacon demodulated by an sTag are examples of a *passive* origination.

Waves and the corresponding localization systems are classified based on their nature: mechanical or electromagnetic. *Mechanical* waves can be further subdivided based on the physical medium they travel through. *Liquid* waves are exemplified by sonar-based underwater object detection and ocean floor scanning systems. Bat biosonar used for aerial navigation and prey localization is an example of a famous biological localization system that uses *gas*-based waves; some indoor localization systems [→1.2] mimic biosonar through technology. The use of *solid* waves is the least common with earthquake localization being a notable example.

Electromagnetic localization is typically limited to aerial uses due to poor propagation of electromagnetic waves in liquids and solids. The *visible light* frequency range of electromagnetic waves is used in robotic/human vision and laser-based measurement systems such as the LIDAR (light detection and ranging) system. *Radio waves* are preferred in many other systems due to their lower attenuation and their ability to traverse solid objects. Radar and many other localization systems including IEEE 802.11 based localization uses RF signals.

Measurement

As the signal travels through an environment, the localized object affects its characteristics. The measurements of these characteristics become the input data to a localization algorithm. The simplest characteristic is the *presence/absence* of a wave; for example, the presence of a beacon frame from a known AP localizes an IEEE 802.11 tag to the general vicinity of that AP. Since higher accuracy is usually desired, other wave characteristics are measured. The wave's *amplitude* can be called "loudness" in mechanical systems or "received signal strength indication" (RSSI) in electromagnetic systems. By comparing the received amplitude with the origination energy, signal *attenuation* is derived.

The localization signal's *travel time* is designated as the *time of arrival* (ToA) when both the origination and the detection time instances are measured. When only the detection time is measured (in reference to absolute time), the measured quantity is referred to as the *time difference of arrival* (TDoA). The signal's *angle of arrival* (AoA) can be obtained *mechanically* through a rotating detector as in older radar systems or *electronically* through beamforming as in the MIMO system proposed in Reference [46]. A related measure is the *phase angle difference* use in the reference-phase and the rotating-phase signals of the very high frequency omni-directional radio range system (VOR) used in airplane navigation. By examining additional signal characteristics such as *frequency*, *polarization* or *embedded digital data*, the localized object can be identified.

Processing

Pattern matching is one of two approaches to measurement processing that yields an entity's relative location and identity. The process begins with an *off-line phase* where signal characteristics are recorded at multiple discrete points in the environment, and is followed by a *localization phase* where a current measurement is compared with the database; a database entry containing the closest recorded characteristics represents the entity's location. A major disadvantage of the *pattern matching* method is the requirement for the mapping step which may be time consuming or outright infeasible. An alternative *environment simulation* method does not require that step and instead assumes that the behaviour of an energy wave can be modelled accurately through mathematical correlation of wave characteristics to physical measurements of distance and direction.

Combining

The localization result obtained from a single detector may lack the desired number of dimensions, for example a distance from the detector without an indication of a direction. This problem may be remedied by correlating information from multiple detectors. Additionally, the accuracy of the answer may improve as more data are used. Common mathematical techniques for these calculations include *angulation* for AoA data, *lateration* for ToA data and *multilateration* (otherwise known as hyperbolic positioning) for TDoA data.

While an exact analytical solution can be provided for simple systems, most systems use multiple measurements. Measurement noise and the inability to model the environment exactly, results in an overdetermined systems of equations. Most generally the computation of location from multiple measurements is a complex optimization problem.

Output

While the pattern matching algorithm's output is already placed in the desired frame of reference, the environment simulation's output requires mapping of the result to a useful reference, e.g. the *distance and direction* relative to the recipient of localization data may be presented, as in a radar system. Other output options exemplified by a typical portable GPS device include the *geographic coordinate system* (latitude, longitude, altitude) or *graphical placement* on a map of familiar features.

2.2.2 Localization Algorithms

Algorithms for calculation of an entity's location from raw measurement data are presented here, with particular focus on those applicable to IEEE 802.11 localization.

2.2.2.1 Localization Result Evaluation Metric

An entity's location is expressed as a point with x, y (and possibly z) coordinates. Hence, the natural metric for performance evaluation is the error distance, i.e. the distance between the entity's true and computed location. For a series of computations, a statistical summary is useful. For example, the accuracy of MS-RADAR[22] was found to be 2.94m for 50% of measurements. A more complete summary may be provided as a cumulative distribution of error (Reference [22] Figure 3). Unfortunately results obtained from various published IEEE 802.11 localization algorithms (such as in Reference [47] Table 1) are difficult to compare because they are dependent on the test environment used. Reference [47] proposes the use of a standard for IEEE 802.11 localization algorithm testing, and describes various specifications of such a standard including the environment, the equipment, the data collection, and the data formatting specifications.

2.2.2.2 Raw Data Processing

Environment Simulation algorithms for IEEE 802.11 localization require that the raw measurement data be converted to the corresponding distance or angle constraints.

The RSSI is the most common characteristic measured that provides the foundation for the majority of IEEE 802.11 localization algorithms. RSSI measurement is a standard process in IEEE 802.11 hardware, designed to support mobility transitions. It is available in many IEEE 802.11 operating system interface drivers through an application programming interface (API), making it is easy to obtain by localization software.

The fundamental relationship that permits the estimation of distance from the observed RSSI is the Friis transmission equation[48]:

$$\frac{P_r}{P_t} = G_t G_r (\frac{\lambda}{4\pi R})^2$$

where P_r is the measured RSSI, P_t is the transmit power level, $G_{r/t}$ are the RX/TX antenna gains, λ is the wavelength, and R is the distance (to be estimated) between the transmitter and the receiver. A serious limitation of this equation is that it assumes unobstructed free space RF propagation.

Hence, a more common version used[49] is the *log-distance path loss model:*

$$PL(d) = PL(d_0) + 10nlog(\frac{d}{d_0}) + X_\sigma$$

where $PL(d)$ is the measured path loss in dB, $PL(d_0)$ is the path loss at reference distance d_0, n is a propagation constant, d is the distance to be computed, d_0 is a reference distance (typically 1m for IEEE 802.11 localization) and X_σ is a zero mean Gaussian random variable that represents the naturally occurring RSSI variations (in dB) for a given environment.

The n parameter in this equation provides a one-parameter summary of the propagation environment characteristics influenced by *slow fading*, such as the shadowing of the transmission path by solid obstacles. The use of n improves accuracy significantly over the original Friis model, as shown in Reference [50] where the *log-distance path loss model* was evaluated specifically for use in IEEE 802.11 systems both indoors and outdoors. The n values for some common environment types can be found in Reference [49], along with the corresponding σ values which represent the variability expected due to inability of the model to account for all slow-fade causes. A separate source of error is *fast fading*: degradation of the signal's amplitude over the time scale of a fraction of a typical packet's transmission time. A common cause of fast fading is a multipath reflection that causes destructive wave interference with the direct signal. Fast fading errors in localization can be partially remedied by using temporal diversity, such as the statistical analysis of RSSI measurements of several consecutive packets.

A well-known further refinement to the Friis equation is the incorporation of attenuation due to traversal of walls and floors, as shown in Reference [51]. The drawback of this approach is that the inverse distance computation now depends on the number of walls and floors in the model, which are unknown. As a result, multiple answers are possible making the refinement unsuitable for use in typical formulations, requiring a more refined approach [→2.2.2.3].

An electrical phase shift is a quantity measurable in a multiple-antenna system, such as in the MIMO system of the latest IEEE 802.11n standard. By considering the spacing between antenna elements and the measured phase shift of the signal, the AoA of the original signal can be estimated. This is a well-known problem of *direction of arrival estimation* and has several popular solutions beyond the scope of this discussion, for example MUSIC: Multiple Signal Classification.

Relatively little work has been done with AoA measurements in IEEE 802.11 networks. This might be attributed to poor practical performance of the method due to signal reflections, and the lack of easy access to AoA data in the still relatively immature IEEE 802.11n operating system interface drivers. Nevertheless, the authors of Reference [46] have performed some experiments using a 4-element non-IEEE 802.11 antenna and provided a formula for AoA calculation (Reference [46] Equation 1) from the array outputs.

The transit time of radio waves is used to estimate distance based on the propagation speed of electromagnetic radiation. The distance travelled in meters d is linearly dependent on transit time in seconds t through the speed of light constant c, i.e. $d \approx ct$.

Negligible speed differences exist due to the refractive index of air, only relevant to large scale systems such as the GPS. In IEEE 802.11 localization systems a much more serious problem is the indoor multipath condition. It is measured by *delay spread*: the time difference between the arrival of the *line-of-sight* and the *non-line-of-sight* signal components. Reference [52] evaluated indoor channel characteristics at 2.4 GHz and found the delay spread in the range of 12–43ns. Based on the equation above, the corresponding localization error would be 3.6—13m. Efforts to reduce these errors include those in Reference [53] which analyzed statistical properties of multiple ToA measurements to estimate the channel profile and account for the multipath components.

Further errors are expected due to IEEE 802.11 hardware clock inaccuracies. Reference [54] demonstrates that the IEEE 802.11 driver API provides timestamps with accuracies of only 1μs which result in an unacceptably large 300m error. However, through statistical processing of 1000 measurements the error is reduced to $<$ 8m.

Reference [55] provides localization performance estimates for the raw capabilities of IEEE 802.11 hardware, disregarding any available API. A direct interface with the IEEE 802.11 NIC's internal clock at 44 MHz provides a 22ns timing accuracy, with the resulting distance measurement accuracy of 7m. Five hundred measurement repetitions can provide a 0.81m accuracy under an idealized *line-of-sight* scenario.

Finally, Reference [56] looks beyond IEEE 802.11 hardware capabilities and considers a high rate signal sampling hardware to provide a higher signal resolution and a correspondingly higher accuracy. 0.21m is the expected theoretical performance of a single measurement in this system in ideal conditions; the actual performance achieved in a sample test deployment was 2.4m.

These efforts demonstrate that ToA-based localization is possible in IEEE 802.11 networks but either at a loss of network bandwidth to accommodate multiple measurements with standard IEEE 802.11 hardware, or a significant expense to add non-standard components that increase timing accuracy. These problems make the ToA systems unpopular compared to the RSSI systems.

2.2.2.3 Measurement Combining

The geometrical constraints obtained from the analysis of a single data source provide only the distance or the direction from the measurement origin and are insufficient for fully three-dimensional localization. Hence, combining of measurements from multiple detectors is discussed below.

A Cell/Proximity algorithm is the simplest to implement but is also the least accurate. The smallest distance is chosen from a the set of distance calculations from multiple detectors, and the unknown location is estimated to be the same as the location of the detector. This technique is more applicable to situations where the detector density is high (such as in the *Smart Floor* system[→1.2]) or in sensor networks where the cost of location computation is high. The cell/ proximity algorithm is seldom used in IEEE 802.11 localization.

Lateration methods determine the unknown location by considering an intersection of multiple spheres, as illustrated in Figure 2.2. Three detectors P_1, P_2 and P_3 have measured distances r_1, r_2 and r_3 to the unknown location X. A system of three quadratic equations with two unknowns can be used to solve for the coordinates of X, which is a special case of lateration called *trilateration*: $||X - P_1||_2 = r_1$, $||X - P_2||_2 = r_2$, $||X - P_3||_2 = r_3$.

A more general case involves n arbitrary number of measurements (≥ 3) and measurement errors $E = (e_1\ e_2\ e_3 \ldots e_n)$. The solution can be expressed as an optimization where the goal is to choose X such that a certain norm (e.g. a 2-norm) of the error vector E is minimized, as shown in Figure 2.2.

Refinements of this basic approach are presented in Reference [57]. In particular, an *iterative non-linear regression* method—an iterative variant of the plain *nonlinear regression* presented above—was evaluated. It was found to have higher accuracy due to its superior rejection ability of multipath-induced outliers, but at an increased computational cost.

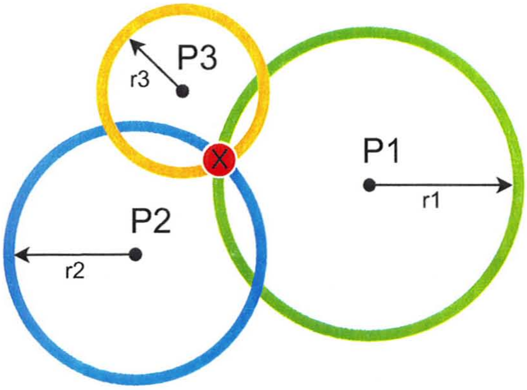


Figure 2.2: Trilateration calculation

$$\min_{X} ||E||$$

subject to

$$| \, ||X - P_1|| - r_1 | \leq e_1$$

$$| \, ||X - P_2|| - r_2 | \leq e_2$$

$$| \, ||X - P_3|| - r_3 | \leq e_3$$

$$\ldots$$

$$| \, ||X - P_n|| - r_n | \leq e_n$$

An angulation method computes the unknown location from measurements of two or more angles. The mathematical foundation of the process is illustrated in Figure 2.3 adapted along with the equations from Reference [58]. Angulation faces similar problems compared to lateration, in that multipath reflections can generate erroneous measurements.

$$S = \frac{Lsin(b)}{Msin(a)}$$

$$g = arctan\left[\frac{cos(b) - Ssin(a)}{Scos(a) - sin(b)}\right]$$

$$Y = L\frac{sin(g-a)}{sin(a)}$$

$$X_p = Ycos(g)$$

$$Y_p = M - Ysin(g)$$

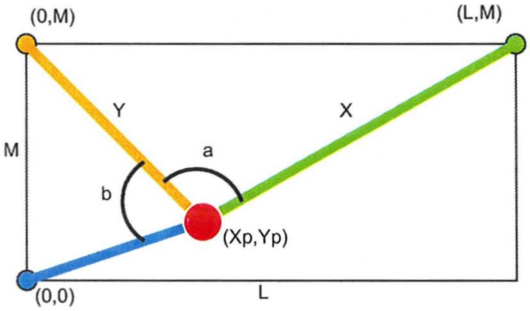


Figure 2.3: Triangulation calculation

Convex Optimization formulations for solving IEEE 802.11 localization problems were developed in a previously evaluated course project[11] that was a precursor to this thesis. A unique feature of these formulations was the incorporation of both lateration and angulation measurements from standard and MIMO-capable APs. In addition, the formulations automatically ignored unknown RSSI discontinuities (due to wall traversal), while still minimizing random errors and remaining convex for ease of computation. A quadratically constrained quadratic program (QCQP) formulation is illustrated in Figure 2.4.

$$\min_{\mathbf{x},\mathbf{v_k},\mathbf{e_k}} ||e||_F$$

subject to

$$||\mathbf{v_k}||_2 \leq 1 \quad k \subset S$$

$$\mathbf{p_k} + \mathbf{v_k} d_k + \mathbf{e_k} = \mathbf{x} \quad k \subset S$$

$$||\mathbf{e_k}||_2 \leq e_{max} \quad k \subset S$$

$$\tan(a_k + a_{max})\mathbf{v_{k,1}} \geq \mathbf{v_{k,2}} \quad k \subset M$$

$$\tan(a_k - a_{max})\mathbf{v_{k,1}} \leq \mathbf{v_{k,2}} \quad k \subset M$$

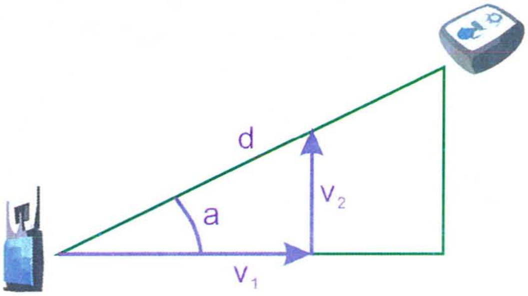


Figure 2.4: Convex Optimization formulation for IEEE 802.11 localization

The variables used were: *k*—an index into a set of all APs (*S*) or MIMO APs (*M*), *x*—the unknown location (Figure top right), *e*—a random measurement error, *d*—the distance measurement, *a*—the angle measurement and *p*—a known position of an AP (Figure bottom left).

The QCQP ("Spheres") algorithm's performance averaged across 32 iterations is illustrated in Figure 2.5 and is compared to a baseline pattern matching ("Fingerprint") method discussed in Section [→2.2.2.4]. A density of about 10 APs on a 60m×40m rectangular floor was necessary for the algorithm to achieve reasonable accuracy performance of about 8m.

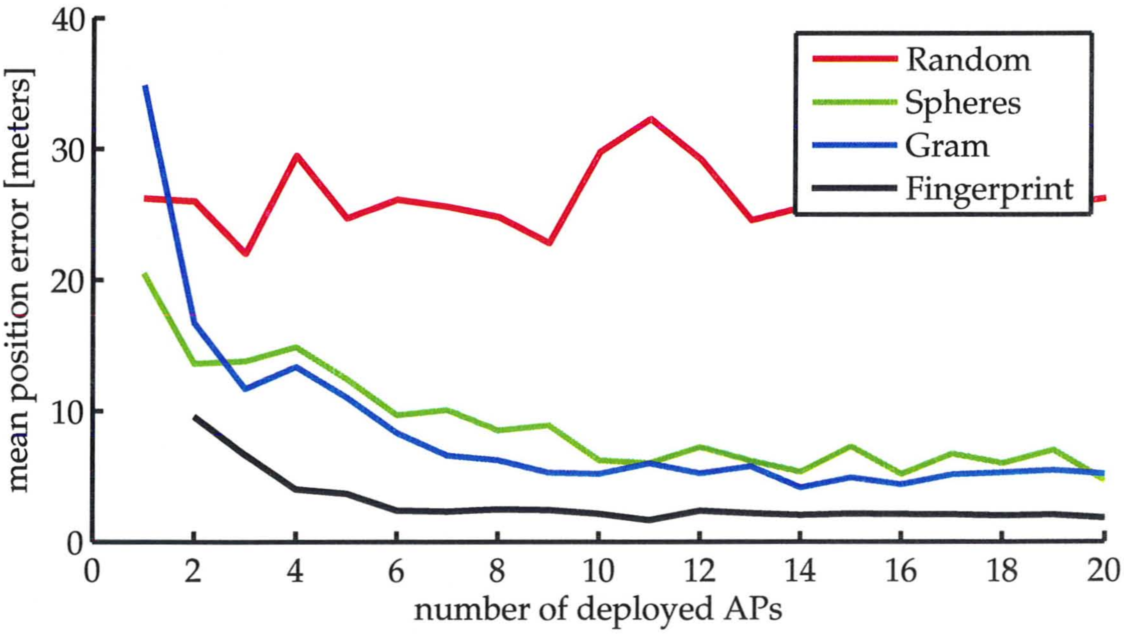


Figure 2.5: Performance comparison of localization algorithms developed in Reference [11].

2.2.2.4 Pattern Matching

The main drawback of *environment simulation* methods is their poor performance in the presence of signal propagation discontinuities. Indoor partitions, furniture, and people can cause unexpected signal attenuation. The *pattern matching* methods not only rectify these problems, but in fact thrive on abrupt signal changes which amplify the differences in the measured RSSIs, improving resolution. Unfortunately, a major disadvantage of the pattern matching method is the requirement [→2.2.1] for an off-line data collection phase which is time consuming and costly.

Pattern Matching Process Description

Data collection is the first step of the pattern matching process. Its objective is to construct a key/value database that matches an observed RSSI vector to its corresponding physical location. The RSSI vector is derived from RSSI measurements of AP management beacon frames received at a given location. Since receiver orientation also affects the RSSI vector, RSSI measurements at multiple orientations are taken, multiplying the vector's size by the number of orientations.

The collection of the location datum itself presents a challenging chicken-and-egg problem. A GPS receiver is generally ineffective indoors, so it can not be used to automate the process. In MS-RADAR (and many other IEEE 802.11 based localization systems) the location has to be provided by the system operator who inputs it by visually comparing his or her surroundings to a digital floor plan. Since this step is time consuming, inefficient, and error prone, it should be automated. A *self-mapping* approach is presented in Reference [59] that uses a graph-based algorithm with a small set of seed data. An outdoor experiment to evaluate the algorithm using sporadic GPS coverage (10%) as seed data achieved an accuracy of 56m with 84% coverage.

Another approach that can be used to construct the database is through sophisticated channel modelling such as ray tracing[60]. It can provide good results as long as the floor plan is accurate; it may change slightly over time, or the material specifications of floor partitions may not be exact. Reference [61] uses ray-tracing to generate a database of AoA and RSSI values for three-dimensional IEEE 802.11 localization.

The location estimation in pattern matching systems involves measuring an RSSI vector ("the pattern") at an unknown location and comparing it to the RSSI measurements stored in the database. The methods for comparing these vectors are diverse and are a subject of much current IEEE 802.11 localization research. An easy to implement approach is the *nearest neighbour* *in signal space* (NNSS) approach as demonstrated in Reference [22]. The comparison is based on Euclidean distance between the observed vector and the stored vectors; the stored vector with the smallest distance is deemed to correspond to the unknown location. A simple refinement is the consideration of k nearest neighbours instead (kNNSS); the authors of Reference [22] found that averaging over $k = 3$ nearest neighbours resulted in a performance slightly superior to the original approach that used $k = 1$. Additional sophisticated approaches were reviewed in Reference [62]; they include probabilistic methods, support vector machine methods and neural networks.

2.2.3 Localization Protocols

A localization protocol's RSSI measurement direction may used to classify IEEE 802.11 localization systems into smart and plain systems. *Smart tags* listen for standard IEEE 802.11 AP management beacons and upload their RSSI measurements to a localization server. Hence, sTags are able to use any standard AP infrastructure. More sophisticated client devices (PDAs, laptops) using smart localization mode can potentially complete the localization process locally without uploading any data, to preserve privacy. On the other hand, *plain tags* in the complementary plain tag system send out OSI L2 multicast frames ("chirps") instead of listening for signals. These chirps are detected and recognized by localization-aware APs and forwarded to a localization server.

2.2.3.1 Plain Tag System Localization

While there are no official standards for either localization mode, the pTag behaviour is partially unified by Cisco Systems through their *Cisco Compatible Extensions* (CCX) for WLAN infrastructure. Manufacturers of CCX tags include AeroScout, G2 Microsystems, and PanGo (InnerWireless).

Generic CCX System

An AP in a CCX compatible plain tag network[63] listens on a single assigned channel to support its normal IEEE 802.11 operations. To support the CCX localization service, it also recognizes pTag's WDS-type packets. Other IEEE 802.11 STAs such as laptops or PDAs can also be localized by CCX, since a STA in a disassociated state issues periodic probe requests on all usable channels to enumerate the APs present. The RSSI measurements of these probe requests are recorded by the CCX AP and made available to a localization server.

A complication arises once the STA authenticates to and associates with an AP. It still continues to scan for other APs to support roaming, but the exact scanning behaviour is not defined in the IEEE 802.11 standards and varies between manufacturers. The roaming-type scanning is usually less frequent than the association-type scanning. Additionally, certain implementations may use passive scanning where no probe requests are sent. Finally, a STA may spend little time scanning alternate channels to avoid disrupting ongoing communications on its primary channel. Hence, some STAs may not send probe requests often enough for reliable localization. Cisco Systems addresses this problem by requiring STAs to comply with the CCX: the STAs are required to perform an active scan in response to an *S36 Radio Measurement Request*. The measurements taken by the STAs are uploaded to the requesting AP using a *Radio Measurement Report*. While this data could be used for a smart-type localization system, in the CCX protocol the actual measured data are ignored and only the AP-measured RSSI of the triggered *Probe Request* is used.

While the generic CCX localization is useful for high power devices continuously connected to a WLAN, it has limited applicability to tag-based systems which are the focus of this thesis.

AeroScout pTags

The behaviour of a typical pTag is exemplified by the AeroScout CCX compliant pTag described in References [63, 64]. After the tag is configured and registered to the localization server, it enters normal operations mode. It waits for its chirp TX interval timer to expire and performs a *CCA* for 100μs as part of the standard DCF operation. When the channel becomes free, the tag transmits a 56-byte long OSI L2 multicast frame (chirp) using a 4-address WDS format [→A.2.1] on each configured channel. Following a successful TX the tag goes back to sleep with only a simple timer circuit and a motion sensor remaining active.

On the infrastructure side, multiple APs detect the tag's TX. Each AP stores the RSSI measurement of the chirp and any extra messages in a table indexed by the tag's MAC address. The APs are periodically polled by a localization server through the simple network management protocol (SNMP) or a Cisco Location Protocol (LOCP) described in Reference [63] Figure 3-26.

PanGo pTags

PanGo manufactures tags[65] similar to the AeroScout tags. After each chirp cycle interval timer expires, an OSI L2 multicast frame [→A.2.2] is sent on all configured channels. The 32-byte long, WDS-format frames are transmitted at 1Mbit/s using 19dBmW TX power setting. The transmission is repeated five times on each configured channel.

2.2.3.2 Smart Tag System Localization

Although the behaviour of sTags is not standardized, the underlying characteristics are common to all implementations. The RSSI measurements of AP beacons is done on the tag and the resulting data are uploaded through standard IEEE 802.11 communications to a localization server. More details of two representative systems are presented below.

PanGo

The same PanGo tag discussed above can also be used in smart mode to interact with a PanGo localization server via IP. The tag acts as an OSI L3 WLAN STA; as part of its normal operational cycle it *wakes up* after sleep timer expiry and joins an IP network. Network joining includes sending of a *Probe Request* frame to find an AP, *authentication* and *association* with an AP and obtaining of an IP address through the *DHCP*. Following that lengthy procedure, the tag scans RF channels for AP beacons and uploads the collected data to a PanGo server.

The network state information is not cached in the PanGo tag memory; a complete DHCP process has to be repeated for every wake-up cycle. There are also address resolution protocol queries necessary to find the address of the PanGo server. Because so many packets are needed for each cycle (4+ for authentication and association, 4 for DHCP, 2 for MAC address look-up, 2 for the actual localization message transmission and ACK), the process is highly inefficient.

Ekahau

While the Ekahau tags are CCX compatible, a smart mode is their primary operational mode[66]. The tag measures RSSIs of surrounding AP beacons and uploads them to a specified Ekahau localization server via IP/UDP on port 8552. The readings may be collected at pre-configured intervals, in response to motion, or in response to a tag's button press. Communication with the server is bidirectional; tags not only send RSSI readings but can also receive current operational parameters such buzzer (de)activation, light-emitting diode (de)activation, or a text message to be displayed on the tag's integrated liquid crystal display.

Skyhook Wireless

The Skyhook Wireless's XPS localization system is a commercial implementation of the research-oriented Place Lab[25, 59] localization system architecture studied by Intel Labs Seattle. The XPS system's design[67] is similar to a smart tag localization system design, but is meant for general-purpose portable computing devices such as cellular phones, PDAs, portable audio/video players, and laptops instead of localization-dedicated smart tags. To enable localization, the system's end-user downloads a 0.5–1.0MB XPS application and a 0–200MB XPS database to his or her portable. The application initiates a standard IEEE 802.11 scan of the WM through the portable's IEEE 802.11 interface, capturing MAC addresses and RSSI measurements of management beacons of nearby third-party APs. A related Patent Application [68] proposes several enhancements to this process: the inclusion of packets other than the management beacons, automatic frame restoration from nearly identical incomplete frames, and a hybrid passive/active scanning method. A different set of optimizations more appropriate to sTag use was proposed in this thesis, where a strong emphasis is likewise placed on the use of RSSI data from abundant third-party APs, even in systems where a partial localization infrastructure is deployed.

Once the RSSI data have been collected, further processing depends on the XPS deployment mode. In the "networked" deployment mode suitable for devices continuously connected to the Internet, the measurements are sent to a central localization server for processing, incurring a 1–3s delay. For intermittently connected devices with limited storage, the "tiled" mode is most appropriate, where parts of the localization database most relevant to the current location are downloaded; the localization processing takes place on the client. Where storage resources are plentiful (e.g. on an audio/video player), the whole database can be downloaded. In all cases, the localization is based on a proprietary variation of the *pattern matching* algorithm [→2.2.1]. The necessary database is constructed by Skyhook employees through wardriving (scanning of wireless networks from a moving vehicle) or through user submissions. The location computation is provided to localization-aware applications running on the client device though standard APIs such as the NMEA 0183 serial communication protocol commonly used in GPS devices.

While the XPS system emphasizes IEEE 802.11 localization, it is actually a hybrid system that integrates IEEE 802.11 beacon measurements, localization fixes from the GPS, and a cellular phone transmitter triangulation. Those three systems can be used jointly or separately depending on the availability of hardware in the client device, the environmental conditions, and the desired localization accuracy. For example, XPS installed on an IEEE 802.11 enabled cellular phone may fall back to 200–1000m accuracy of cellular transmitter triangulation in a rural area where IEEE 802.11 signals are scarce. However, even when a GPS receiver is available, the IEEE 802.11 localization option still provides a benefit through an improved time to first fix of 1s compared to assisted GPS's performance of 30s. The area availability of the hybrid system is also higher at 99.8% compared to 95% for assisted GPS alone. The popularity of the XPS system reaffirms the validity of this thesis's focus on IEEE 802.11 localization in mixed outdoor/indoor scenarios.

Chapter 3

System Design

System designs for both the *plain* and *smart* tag systems are proposed in this chapter. The primary components of both systems are tags and SPLANs. A localization **tag** (a pTag in the plain tag system or an sTag in the smart tag system) is a small, battery powered embedded system with an IEEE 802.11 radio; the tag attaches to the localized entity. The newly proposed **SPLAN** device is a medium-size embedded system (SBC) with an IEEE 802.11 NIC; the SPLAN is powered by a battery recharged by a solar panel. Its software allows it to automatically form a WMN with other SPLANs, communicate with tags, and act as an IEEE 802.11 STA to other IEEE 802.11 APs. A programmable sleep timer suspends the SPLAN for a specified duration.

3.1 Plain Tag System

A pTag (plain tag) [→2.2.3.1] includes a limited functionality IEEE 802.11 radio capable of performing carrier sense and transmitting chirp packets at the lowest IEEE 802.11-1997 speeds[→2.1]. The tag is programmed to transmit chirps at preset intervals and remain dormant in PS mode for the remaining time. Its chirps are received by nearby APs and SPLANs, where their RSSI is measured and uploaded to a localization server for location computation.

The advantage of the plain tag mode is that the pTag only needs to be activated briefly to transmit the chirp, keeping power consumption to a minimum. The disadvantage is that only localization-aware network nodes (special APs and all SPLANs) are able to collect the RSSI readings; the localization accuracy improvement potential of third-party APs is ignored.

3.1.1 pTag System Architecture

The IEEE 802.11 localization industry lacks formal standards, but Cisco Systems CCX-type tag is currently closest to a *de facto* industry standard. With such standardized deployments already present, the proposed SPLAN should inter-operate transparently with these tags and APs to preserve the investment in existing infrastructure. To this end, the hardware and software of these tags and APs shall be generally considered to be immutable.

A CCX system is illustrated in Figure 3.1. It consists of Cisco Systems APs supporting both standard data/multimedia services and a CCX-compatible pTag localization service. The APs are connected by Ethernet to a localization server. Ignored for simplicity is the existence of lightweight APs and their controller, as well as a localization appliance which provides an interface between the controller and the localization server.

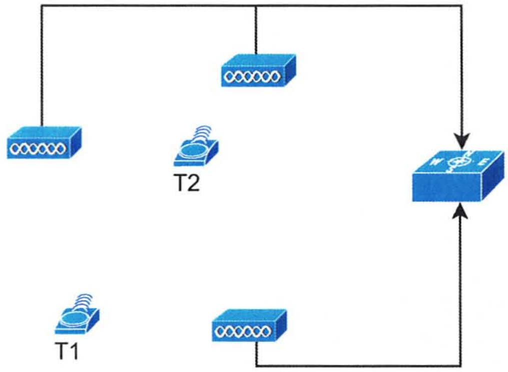


Figure 3.1: CCX network legacy architecture

A typical problem in a CCX system is a low density of APs due to their high cost. Additionally, the AP deployment pattern may be suboptimal for the localization service due to the difficulty of wiring certain locations, resulting in the existence of areas with poor localization accuracy. This is illustrated in Figure 3.1 by tag T1 which lacks sufficient AP coverage in its 100–340° arc.

The addition of SPLANs to a CCX network is illustrated in Figure 3.2. The SPLANs are standalone units (no power or data cabling) connected to the adjoining APs as standard IEEE 802.11 STAs to disseminate RSSI data to the localization server. Localization accuracy improves through elimination of poor geometrical arrangements (pTag T1) and increased density (pTag T2).

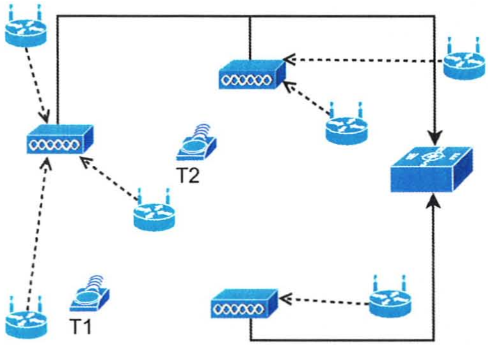


Figure 3.2: CCX network with SPLANs

3.1.2 pTag Synchronization

A significant complication in the operation of the pTag system is the synchronization of the SPLAN to pTags. This thesis defines *synchronization* as the process by which pTags and the SPLAN establish communication—analogous to STA association in IEEE 802.11 data network. While plain tags do not associate with standard APs, they need to be synchronized with SPLANs so that the SPLAN can perform PS. The synchronization process must be passive and unidirectional, i.e. the SPLAN synchronizes its reception schedule to the existing tag chirp schedule based on the timing of chirp transmissions. Note that in the pTag system, the behaviour of pTags is considered to be defined by the CCX standard and can not be arbitrarily changed to meet the PS requirements of the SPLAN; the SPLAN must adapt to existing system constraints.

The challenge for pTag synchronization lies in obtaining the tag's chirp *phase* offset and *interval* to create a suitable listening interval (LI). The SPLAN may be asleep when a tag is first turned on or enters the coverage zone of the SPLAN. In these cases the tag's TX is lost due to unknown chirp *phase*. Additionally, the pTag does not advertise its chirp *interval* so even if the SPLAN knows the phase of a single chirp, it does not know when to wake up for the next chirp.

A possible solution that does not require interaction with the existing system is the creation of periodic listening windows during which the SPLAN is continuously active for the duration of the maximum allowed chirp interval. The drawback is that the active duration will limit the maximum supported tag chirp interval while the sleep duration will extend the time to first pTag acquisition. For example, a 6min activity interval out of every 60min will incur a baseline power consumption of 10%, with a maximum supported chirp interval of only 6min and a maximum waiting time of 60min before localization assistance is provided. Since these trade-offs are unacceptable, the SPLAN requires assistance from APs which are continuously active already and do no incur additional power consumption penalties. In the proposed system the SPLAN uses the API of the CCX-compatible neighbouring APs to extract a list of registered pTags. It then compares this list to a one that it maintains to determines if there are any new pTags that require a new LI.

3.1.2.1 Legacy Compatible Synchronization (pSyncL)

pSyncL is the first proposed synchronization method compatible with legacy CCX systems; it is illustrated in Figure 3.3. To find the pTag chirp phase, the SPLAN polls the AP at time T1; the response does not contain any unknown tags. At time T2 a tag is first turned on or moves into the area; it emits a standard chirp received by the AP but not by the SPLAN which is asleep at

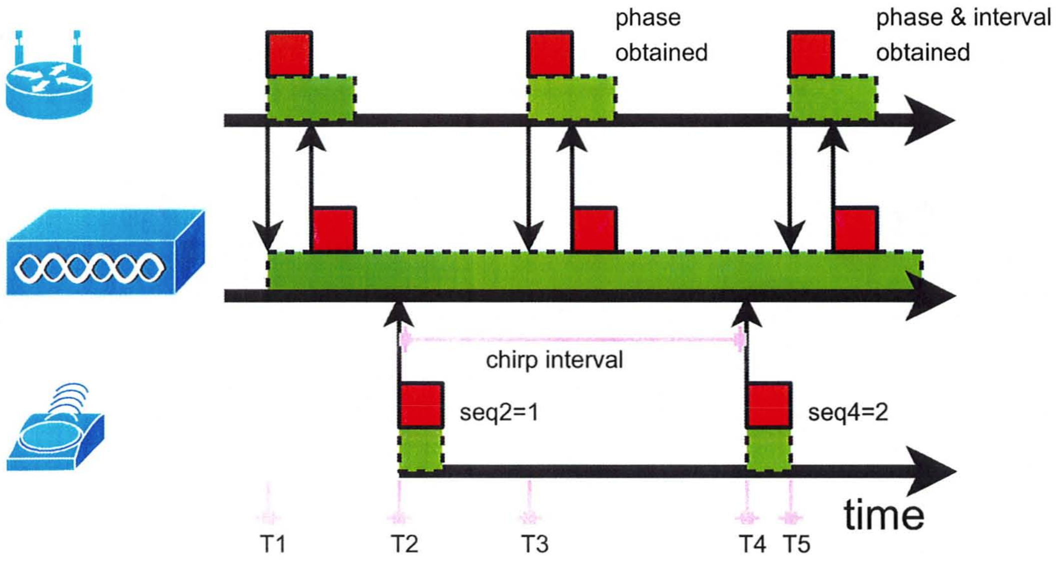


Figure 3.3: pSyncL legacy synchronization for pTags

that time. At time T3 the SPLAN issues another poll and the AP responds with data about the new tag with chirp phase T2. However, the chirp interval (i_l) is still unknown at time T3. It needs to be *measured* by comparing two timestamped transmissions, which requires double AP polling. The chirp interval is calculated as $i_l = (T4 - T2)/(seq4 - seq2)$ where T4 and T2 are the timestamps from Figure 3.3 and $seq4, seq2$ are the corresponding OSI L2 chirp sequence numbers. Consideration of chirp sequence numbers accounts for situations where the chirp interval is much smaller than the selected AP polling interval. When SPLAN polls the AP at time T5, it computes the chirp interval and starts actively participating in localization by providing a LI for the tag.

The analysis of Figure 3.3 demonstrates that reception of at least two pTag chirps is required for the pSyncL approach; if the tag's mobility is high, many localization assistance opportunities will be missed. Similarly, if the tag chirp interval is short compared to the SPLAN poll interval, many packets will be transmitted between polls, wasting assistance opportunities.

3.1.2.2 Firmware Assisted Synchronization (pSyncF)

An alternative pSyncF [→A.5] synchronization approach requires extending the functionality of the CCX system to provide further support for the SPLAN. The proposed modifications require an insignificant amount of new firmware code and fully preserve backwards compatibility. The modified tag transmits its current chirp interval value in every chirp packet while the modified AP informs SPLANs about new pTags through a new communication schedule.

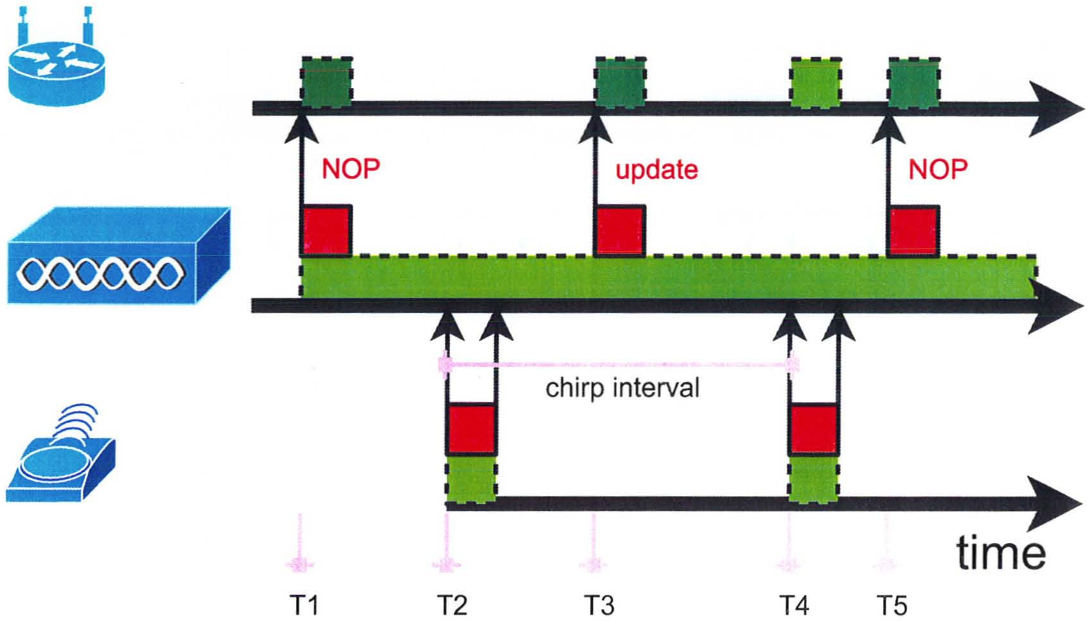


Figure 3.4: pSyncF assisted synchronization for pTags

The operation of the pSyncF synchronization method is illustrated in Figure 3.4. At time T1 the SPLAN begins its regularly scheduled AP communication interval and receives a response with no new pTag information. At time T2 a new pTag emits a chirp which includes the chirp interval value; it is received and recorded by the AP. The AP shares both the phase and interval of the new tag with the SPLAN at time T3. The SPLAN configures a new LI appropriately and receives the second pTag chirp at time T4.

3.1.3 pTag and SPLAN Operation in Localization

The SPLAN schedules a LI for a tag once it acquires the necessary timing information. To account for hardware clock errors, it uses the standard deviation of the observed pTag's sleep interval durations to decide how much earlier it needs to wake up to successfully receive the chirp with a given probability. For example to attain a 99.73% chirp reception success rate, it uses a guard time equal to 3 standard deviations of the pTag's sleep times. Upon each successful chirp reception, the SPLAN updates its timing information table with the chirp phase and sleep time standard deviation to account for clock drift and error changes over time.

The general pTag/SPLAN operation is illustrated in Figure 3.5, where two pTags with different chirp intervals of 5min and 8min are shown. The AP listens continuously, while the SPLAN creates LIs with the correct phase and interval to capture pTag's chirps.

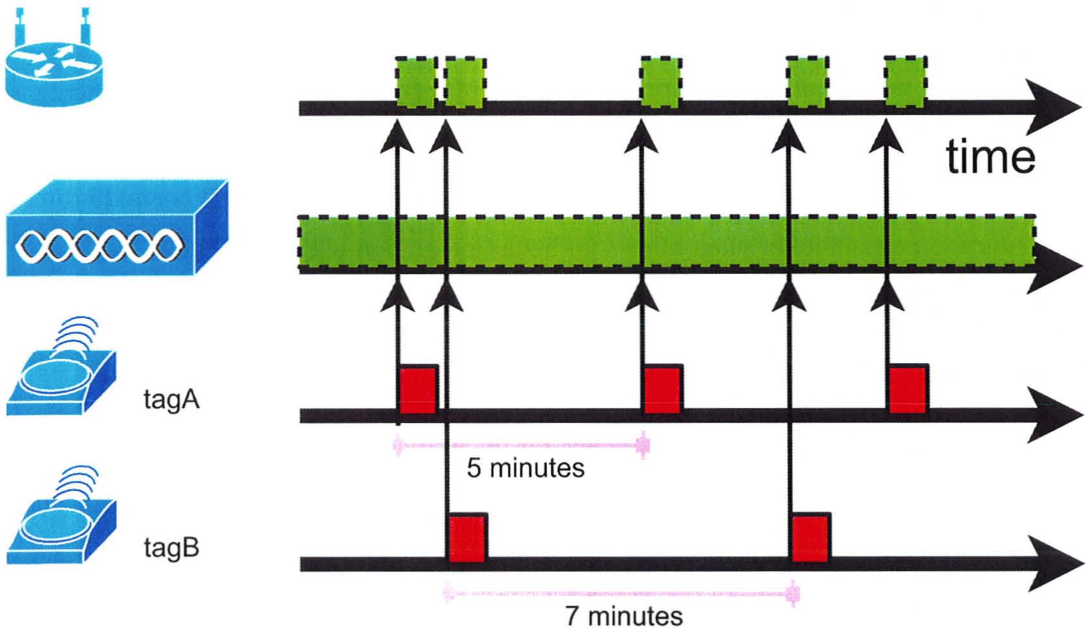


Figure 3.5: pTag localization overview

Figure 3.6 illustrates a single chirp event. The tag emits its first chirp packet following a CCA, while the SPLAN successfully receives it and goes back to sleep, providing that there are no collisions or interference. Irrespectively of the success or failure of the first transmission, the tag emits additional packets according to its configured settings, separated by an inter-packet wait interval. Here a single additional chirp packet is seen.

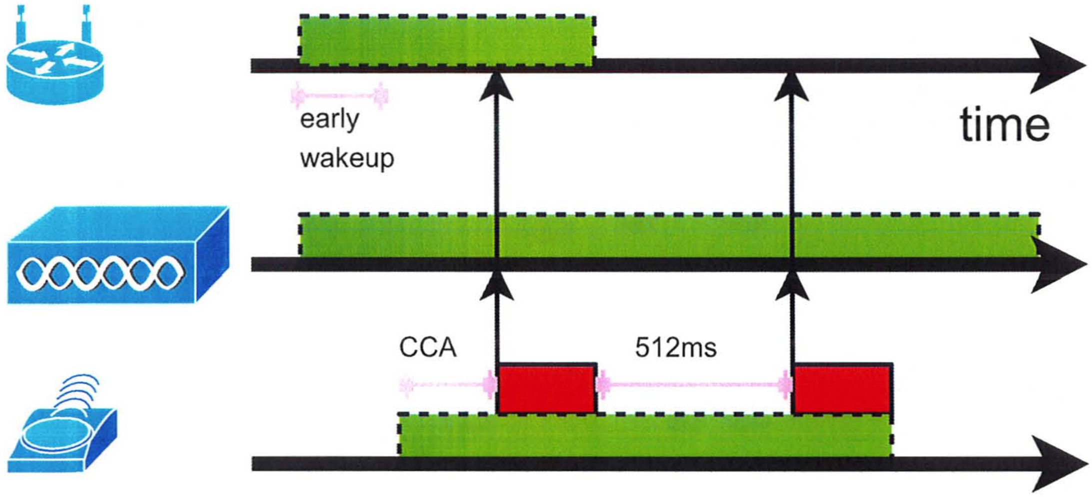


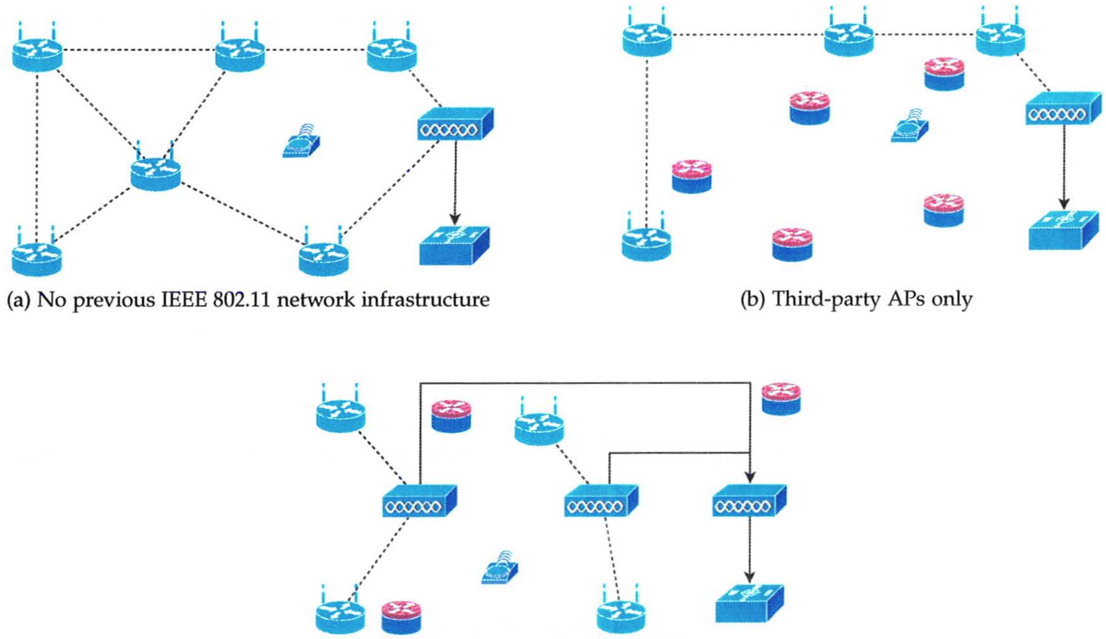
Figure 3.6: pTag single chirp event

3.2 Smart Tag System

Whereas plain tags require the support of the AP infrastructure for RSSI measurements, smart tags do not share this limitation as they take advantage [→2.2.3.2] of any deployed IEEE 802.11 APs which may not be easily upgradeable to support localization. Also, the inherent bi-directional communication capability of sTags allows for better integration with SPLAN PS.

3.2.1 Deployment Scenarios

A SPLAN operating in plain mode is suitable for integration with an already-deployed plain tag network, but a similar arrangement is not appropriate for sTags. There is no official standard or even an industry standard for smart tags, and all reviewed systems were proprietary, with no detailed functionality descriptions available [→2.2.3.2]. Even with sufficient information, any possible SPLAN integration design would only be applicable to that single propriety framework which would restrict the potential for SPLAN PS. Thus, a smart tag system designed specifically for SPLAN integration is going to be considered instead. There are three distinct initial starting scenarios that can share the same SPLAN front-end (sTag/SPLAN) signalling design, as shown in Figure 3.7a, Figure 3.7b, and Figure 3.7c.



(a) No previous IEEE 802.11 network infrastructure

(b) Third-party APs only

(c) A mixture of own and third-party APs

Figure 3.7: Smart tag network designs

No Network Infrastructure (Figure 3.7a)

A new farm animal localization application is unlikely to have an existing IEEE 802.11 APs. Therefore the SPLAN provides all the localization services including communication with sTags and dissemination of locations through a WMN.

Third-party AP Infrastructure Only (Figure 3.7b)

There are existing APs, but they are controlled by a third-party, i.e. an entity other than that implementing the localization service. An example is a city-wide localization network for public transit tracking; the city does not possess any IEEE 802.11 infrastructure but there exist APs owned by individuals and businesses, as in Reference [25]. In this scenario, the SPLAN is providing the localization dissemination back-end through a WMN, while the third-party APs provide the source for localization measurements: standard management beacons captured by smart tags.

Mixture of Third-party and Own APs (Figure 3.7c)

An example of this scenario is a university campus with significant quantity of "rogue" (personally administered) APs, but also a large network of university administered APs. The university APs are connected to a wired distribution system, while the third-party APs are standalone for the purposes of localization. SPLANs no longer require sufficient density to form a WMN among themselves, since they can act as STAs to the existing APs while cost-effectively filling in localization coverage gaps.

For all three of the above illustrated deployment scenarios, a design based on the plain tag system is unsuitable. In Figure 3.7c the self-owned APs may not be CCX compatible and thus may not accept plain tag chirps. Even if the self-owned APs are assumed to be compatible, there may be many other third-party APs already present that could not be utilized in a plain tag system, as illustrated in Figure 3.7b. Finally, pTag systems offer much less opportunity for SPLAN PS due to a lack of explicit pTag/SPLAN synchronization. For these reasons, SPLAN-based smart tag systems are more appropriate for scenarios without existing legacy plain tags.

3.2.2 sTag System Architecture

Infrastructure APs in the smart tag system are localization-unaware and immutable. They are deployed for a particular purpose (e.g. to provide data or VoIP services) with no consideration of localization. Some of these APs are third-party owned; their only—but important—purpose is to provide beacons for RSSI measurement by the sTag. Management beacons are part of the IEEE 802.11 standard, so their existence can be safely assumed in a typical network configuration.

The sTags proposed for the integration with the smart-mode SPLAN are assumed to be constructed from typical hardware of contemporary smart tags [→2.2.3.2], but use a custom designed firmware. While the design of the custom hardware is beyond the scope of this work, unnecessary, and costly, the custom firmware design is necessary to support advanced PS. There is no advantage of adapting the design to a specific existing sTag behaviour without existing standardization of smart tag protocols. A SPLAN designed to inter-operate with one particular proprietary system would not work with another. Also, forcing the SPLAN to inter-operate with non-power-aware smart tags of a particular existing design would result in loss of significant potential for PS, just as in the plain tag system.

The SPLAN provides accuracy augmentation services, as in the plain tag system. The difference is that the SPLAN LI activities include the transmission of beacons instead of just the reception of tag chirps. The SPLAN also takes on more back-end support roles, specifically the dissemination of collected RSSI data to a localization server. In the plain tag system each AP and each SPLAN was responsible for disseminating the RSSI measurements of a single pTag chirp to the localization server, whereas in the smart tag system the SPLAN alone disseminates a complete set of the RSSI readings collected by an sTag. This thesis proposes an enhanced protocol for localization data reporting from the sTag to the SPLAN, designed to improve both the network utilization and SPLAN/sTag power consumption over the corresponding attributes in a conventional sTag system that functions within the framework constraints of an unmodified IEEE 802.11 MAC and TCP/IP [→2.2.3.2].

A localization server design is necessary to accommodate SPLAN-mediated data collection, but the actual design is beyond the scope of this project, as are the back-end communications. Once the SPLAN acquires RSSI readings from sTags, further data dissemination (through the SPLAN WMN, to a receiving portal node, to the localization server) is not considered.

3.2.3 sTag Registration

pTags in the **plain tag system** make their presence known to always-on APs; this information is forwarded to SPLANs via pSyncL or pSyncF polling to establish pTag synchronization. In the **smart tag system** APs do not participate in localization actively so an sTag registers directly with SPLANs. If the SPLANs were listening continuously for new tag registrations, no PS could be realized. Instead, sTags share some registration burden and listen for periodic SPLAN beacons.

3.2.3.1 Registration Levels

An sTag which is turned on for the first time has no knowledge of network timings; the corresponding registration state is designated as the **R0: unregistered** state. Progressively higher registration states are the **R1: network registration** state, when an sTag knows the registration beacon timing used in the entire SPLAN network, and the **R2: SPLAN registration** state when an sTag knows the timing for an LI provided by a specific SPLAN.

Both the R1 and R2 states are obtained through the same registration process described subsequently; the difference lies in their loss. When an sTag moves to a service area of another SPLAN it looses the R2 but retains the R1 registration state. The R1 state permits quick registration with other SPLANs which all share the same registration beacon timing offset. A SPLAN treats sTags in R2 state as active and includes them in its internal algorithms for LI assignment. Unlike in the IEEE 802.11 *association*, an sTag may be registered to more than one SPLAN at the same time.

3.2.3.2 Registration Protocol Details

The most common R1→R2 registration sequence begins with the sTag waking up from sleep just before the expected time of SPLAN registration beacon, as illustrated in Figure 3.8. A guard time proportional to the duration spent unsynchronized compensates for accumulated timing errors. The sTag enters listening mode while the SPLAN carries out CCA followed by a transmission of registration beacon packet shown in Table 3.1. ($reg\_beacon\_size = F_{pm} + 3F_{pc} + 1 + 8 + 8$ bytes)

Table 3.1: Packet structure of an sTag registration beacon

item type	size (bytes)	argument
beacon	1	*registration* subtype
BSS timestamp	8	timestamp in μs
SPLAN timestamp	8	timestamp in μs

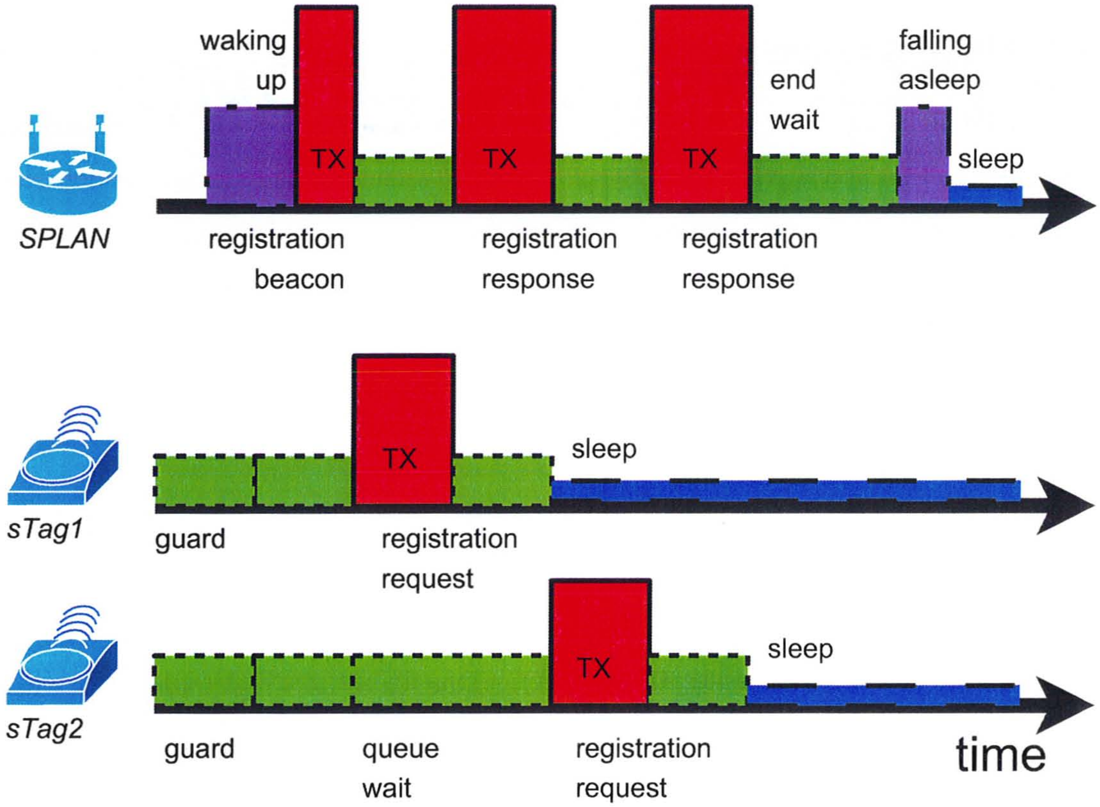


Figure 3.8: SPLAN and sTag activities during registration

The *beacon* datum designates the packet as a beacon-type while the argument selects the *registration* subtype. The *SPLAN timestamp* is a timestamp global to the entire SPLAN network, preferably expressed in real time if Internet connectivity is available. It is through this timestamp that sTags initially set their clocks and make clock corrections throughout their lifetime. Timing information encoded in other control packets shown subsequently is also based on this timestamp.

The other beacon timestamp is a *BSS timestamp* local to each SPLAN. Its use is necessary to preserve compatibility with standard IEEE 802.11 hardware, as some indeterminate amount of time will pass between the SPLAN control software's encoding of the SPLAN timestamp into the packet's data portion, and that packet's transmission. The delays are due to passing of the packet through various communication layers and the CCA procedure. The solution is the use of the supplementary *BSS timestamp* which is encoded into the packet's body during the packet's creation. At the moment of packet's transmission, an updated BSS timestamp is also encoded in the packet's MAC header by IEEE 802.11 hardware. The receiving sTag compares the MAC and data portion BSS timestamps to determine how much time has passed between the packet's encoding and transmission. This adjustment is added to correct the primary *SPLAN timestamp*.

Upon reception of the registration-type beacon, the sTag transmits a registration request message shown in Table 3.2. ($reg\_packet\_request\_size = F_{pm} + 2F_{pc} + 1 + 4$ bytes)

Table 3.2: Packet structure of an sTag registration request

item type	size (bytes)	argument
registration	1	*request* subtype
chirp interval	4	milliseconds

The *registration* datum identifies the packet as a registration request packet while the i_l *chirp interval* is the requested localization QoS specification; the sTag is informing the SPLAN about the frequency of the localization updates that it requires to meet the specifications of its localization role [→2.1.4]. While the specification could be much more elaborate[→7.6], in this model the QoS transaction is simple: the SPLAN can only accept or deny the QoS request. A denial (Table 3.3) occurs if the number of currently registered tags exceed the defined software limit n_{max}.

Table 3.3: Packet structure of an sTag registration response: n_{max} limit exceeded

item type	size (bytes)	argument
registration	1	*response* subtype
status	1	limit exceeded

To accept an sTag, the SPLAN checks if it already has a LI that matches the requested i_l and that the capacity of that LI has not been exceeded. The associated *interval_tag_max* parameter is dictated by the efficiency of multiple sTags competing for the WM as discussed in Section [→4.4.3.3]. If these conditions are met, the sTag is accepted into an existing schedule and sent a success response message shown in Table 3.4. ($reg\_packet\_response\_size = F_{pm} + 4F_{pc} + (1 + 1 + 4 + 4)$ bytes)

Table 3.4: Packet structure of an sTag registration response: success

item type	size (bytes)	argument
registration	1	*response* subtype
status	1	ok
phase	4	milliseconds
interval	4	milliseconds

The *phase* and the *interval* are the LI specifications; the *interval* expresses the time duration between successive localization beacons while the *phase* provides the beacon's offset.

At this point in time, the registration of a single sTag is complete; if there are more tags awaiting registration they follow the same procedure. The SPLAN detects the lack of further pending registrations by applying CCA for several normal durations ($cca\_multiplier = 5$), and initiates sleep. An sTag can initiate sleep immediately after receiving the registration response packet without waiting for the SPLAN.

Note the inherent optimization specific to sTags featured in this design: multiple tags are aggregated within a single LI to keep SPLAN's active/sleep transition overhead to the minimum. This is not possible in the pTag system because the SPLAN has no control over pTag phases and must create a separate LI for every pTag unless they coincide by chance.

3.2.3.3 Listening Interval Scheduling

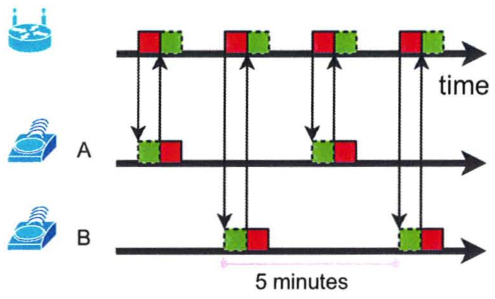


Figure 3.9: sTag basic LI scheduling

Listening interval (LI) scheduling is the cornerstone of PS in a smart tag system. Figure 3.9 illustrates a *naïve* scheduling approach reminiscent of the plain tag system; a separate LI is created for each sTag registering with the SPLAN.

When several sTags share the same i_l QoS specification, they are all serviced in a single, larger LI. This approach is superior because it eliminates the following overheads of separate LIs: extra processing to track individual LIs, SPLAN wake-up time (consumes energy and reduces capacity), SPLAN beacon TX (wastes WM capacity), and registration guard time. The SPLAN creates only a single *shared* LI when a first tag is initially registered. When a second tag is registered, it is assigned to the first tag's LI. As more tags are registered, the original LI becomes full and a new LI with a different phase but identical i_l is created to accommodate more tags.

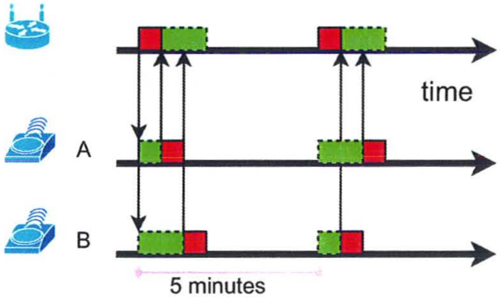


Figure 3.10: sTag combined LI scheduling

Figure 3.10 illustrates the optimization; both tagA and tagB registered in the same LI wake up at the same time and receive a SPLAN beacon. Subsequently, tagA transmits RSSI information to SPLAN, followed by tagB. This process is repeated $i_l = 5$min later, but with the TX order reversed.

3.2.4 sTag and SPLAN Operation in Localization

The registration process described in Section [→3.2.3] created LIs necessary for the sTag and the SPLAN to temporally converge their active states to perform the localization process. Localization consists of three phases: localization beacon transmission, RSSI data collection and RSSI data uploading. Figure 3.11 provides an overview of the process, illustrating the operation of two SPLANs, two APs, and two sTags.

The localization cycle begins at time T1 with SPLANs and sTags waking up from sleep according to the LI schedule. SPLANs transmit localization beacons and enter sleep, while sTags receive and measure them. At time T2, sTags measure AP management beacons through a legacy or an assisted scan. At time T3 the SPLANs wake up after a fixed time. Both tags upload all collected RSSI measurements according to WM access priorities of standard or assisted DCF. Note that these transmissions are directed to and acknowledged by different SPLANs.

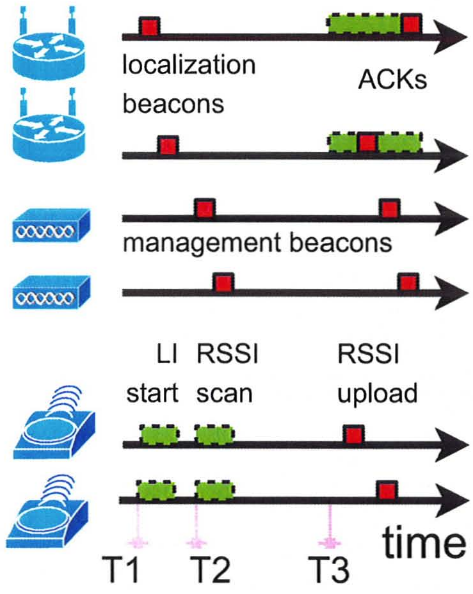


Figure 3.11: sTag localization transaction

Localization Beacon Transmission

To capture a localization beacon an sTag wakes up from sleep, activates its radio, and tunes to the SPLAN's channel. Concurrently, the SPLAN also wakes up and sends the localization beacon shown in Table 3.5. ($loc\_beacon\_size = F_{pm} + 14F_{pc} + (1+8+8+2+10(1+4+4))$ bytes)

Table 3.5: Packet structure of an sTag localization beacon

item type	size (bytes)	argument
beacon	1	*localization* subtype
BSS timestamp	8	timestamp in μs
SPLAN timestamp	8	timestamp in μs
DCF count	2	number of registered tags
AP beacon 1	1+4+4	channel, offset, phase
AP beacon 2	1+4+4	channel, offset, phase
...		
AP beacon *j*	1+4+4	channel, offset, phase

The purposes of the *DCF count* and the *AP beacon j* items are explained in Sections [→3.2.6] and [→3.2.5], respectively. Since localization performance stops improving[11] with more than 10 APs, *j*=10 is used to model the packet size.

RSSI Data Collection

The sTag uses data contained in the localization beacon to perform a passive RSSI scan [→3.2.5], while the SPLAN sleeps for the duration of this phase. Since the scanning duration varies depending on the actual calculated schedule, the SPLAN uses a fixed sleep time guaranteed to be larger than the sTag's maximum scanning time.

RSSI Upload

The RSSI collection process leaves all sTags synchronized; collisions would result from immediate transmissions. Hence, an assisted DCF algorithm [→3.2.6] is used to mitigate this problem.

Multiple SPLANs may received the RSSI measurement data packet. If multiple SPLANs ACK them too, it there would be too many unnecessary ACKs and too many unnecessary messages sent by SPLANs to the back-end system. To resolve this problem the tags direct their packets to a particular *primary* SPLAN based on an algorithm that favours the highest SPLAN beacon RSSI.

Hence, only the tag's primary SPLAN sends an ACK; if the tag does not receive the ACK, it retransmits the RSSI data. Further failures prompt the tag to direct its packet to another SPLAN. For comparison, consider that pTags do not use ACKs for their chirps which are instead repeated several times (3–5) to increase the chance of their transmission success. This consumes network resources needlessly and still does not guarantee correct operation. Additionally, an ACK-less protocol would function poorly in an sTag/SPLAN scenario with tightly synchronized transmission intervals and the resulting high probability of collisions.

3.2.5 RSSI Data Collection: Enhancements

Beacon collection is a standard IEEE 802.11 procedure [→2.1.3], not an exclusive sTag function. The disadvantage of the **passive** collection method for use in localization use is its long duration; a tag has to spend at least 100ms listening on each channel in its regulatory domain channel set[37]. Unlike in the pTag network where there is a tight control over the AP infrastructure and only a subset of channels may be used (e.g. 1/6/11), beacons may be present on any channel in an sTag network. While the **active** scanning approach reduces the 100ms waiting time on unused channels, it is undesirable for sTag use. The transmission of a *Probe Request* packet necessitates the use of a high power radio mode, while the capacity wasted by the *Probe Request* and the *Probe Response* packets can be considerable in a busy localization system. The elicitation of a *Probe Request* packet from a third-party AP is also undesirable for political reasons; passive beacon collection is a neutral activity, while the engagement in an active communication with third-party APs uses their resources without an explicit permission. Therefore, only the *passive* beacon collection method shall be considered for the sTag system.

Figure 3.12 shows the beacon collection strategy of an sTag using the IEEE 802.11 legacy approach [→2.1.3]. There are three APs on three different channels. In the first 100ms interval, a legacy sTag tunes to channel 1 and captures the CH1 beacon from the first AP; in the second 100ms interval, it captures the CH2 beacon, etc. The total duration required is for the scan is $(channel\_count)(100\text{ms}/channel)$.

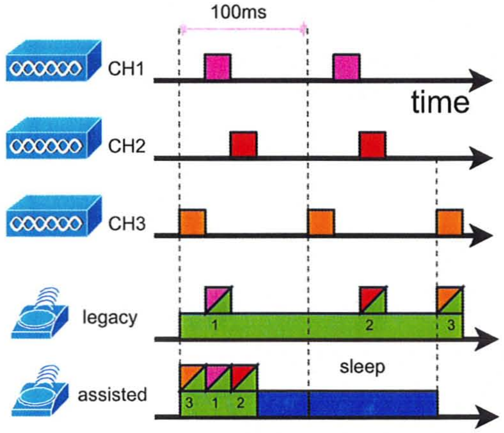


Figure 3.12: sTag RSSI collection of non-overlapping beacons

SPLAN Assisted Passive Scan

To provide scanning assistance, the SPLAN periodically performs an exhaustive site survey; the phase and the interval for all AP beacons in the SPLAN's vicinity are recorded and stored. This process is similar to that used by CCX-compatible APs which conduct periodic off-channel scans used for rogue AP and STA detection; the scans last 500ms out of every 180s of AP operation.

At the beginning of a LI the SPLAN broadcasts the collected information in the data portion of its beacon packet shown in Table 3.5. sTags aggregate this information from all the SPLAN beacons, generating a beacon map of all APs in their vicinity. The beacon map consists of the AP's *MAC address*, the *channel* that the AP currently uses, the *phase* of the beacon relative to absolute or network time and the *interval* of the beacon (usually 100ms).

An assisted sTag using the SPLAN-provided timing information is also illustrated in Figure 3.12; the tag knows that APs are only present on channels 1, 2 and 3. Using the beacon phase offsets, it tunes to the correct channel immediately before the expected beacon transmission. The scan takes less time and uses less energy: the sTag is free to sleep after a successful collection.

Scan Scheduling Algorithm

The SPLAN-provided timing information is used by an sTag to calculate an optimal scanning schedule: channel numbers and time offsets necessary to receive the beacons. The solution's optimality is characterized by minimization of the active time. If beacon density is high, some beacons may overlap and more than one inter-beacon interval may be needed to capture them.

For simplicity of analysis [→7.7] it is assumed that the tag remains active throughout the scanning process, that channel switching is instantaneous, and that the SPLAN-provided timing data are exact. The simple algorithm used is "tune to the next beacon in sequence" unless there is an overlap resolved by "postponing reception of an overlapping beacon to the next cycle."

An sTag can not upload the RSSI measurements immediately after capture, since the SPLAN is sleeping. The SPLAN's sleep duration is fixed to account for scan interval variability due to each tag having a different superset of all beacon timings from all nearby SPLANs. For simplicity of SPLAN energy analysis, the maximum capture interval length is fixed as **two** standard IEEE 802.11 beacon durations.

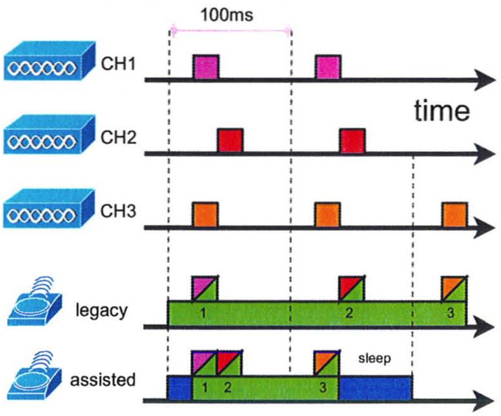


Figure 3.13: sTag RSSI collection of some overlapping beacons

The second interval is needed since some beacons may overlap temporally on different channels, as shown in Figure 3.13. The assisted sTag can not receive beacons from AP2 and AP3 simultaneously with a single radio; reception of the CH3 beacon is postponed until the second beacon interval. Note that a temporal beacon overlap on a single channel is resolved automatically by CCA and does not need to be considered separately.

3.2.6 RSSI Upload: Enhancements

Once an sTag has gathered beacons from all SPLANs and APs in its vicinity, it must upload these RSSI measurements to a localization server. A legacy sTag communicates through an AP owned by the organization providing the localization service. This communication includes AP association and authentication, DHCP transaction and a multi-packet IP-based data exchange [→2.2.3.2]. In the proposed system, the sTag communicates with the SPLAN at OSI L2 to avoid the overhead of establishing IP communications each time. The use of a standard IP network might not even be possible in some scenarios due to lack of any self-owned APs.

To upload RSSI measurements, the sTag selects the *primary* SPLAN (a SPLAN with the highest RSSI) and sends it a RSSI data packet shown in Table 3.6 at 11Mbit/s. ($rssi\_send\_size = F_{pm} + 12F_{pc} + (1 + 8 + 10(1 + 6 + 1))$ bytes). The SPLAN replies with an ACK packet; if no ACK is forthcoming immediately, the sTag assumes that its transmission was corrupted and retransmits immediately. No delays are permitted since the SPLAN uses absence of transmissions as a trigger for termination of the LI. Further error recovery is also possible by changing the transmission destination to the SPLAN with the next highest RSSI.

Table 3.6: Packet structure of an sTag RSSI data

item type	size (bytes)	argument
RSSI	1	*data* subtype
SPLAN timestamp	8	timestamp in μs
AP measurement	1 + 6 + 1	1, AP SSID, RSSI
...		
AP measurement	1 + 6 + 1	*j*, AP SSID, RSSI

The *RSSI* datum identifies the packet as part of RSSI upload transaction, while *data* subtype indicates that it is providing RSSI data. The *SPLAN timestamp* enables the localization server to determine when the RSSI reading was taken, as indeterminate amount of time may pass between RSSI measurement and its arrival at the localization server. The actual RSSI measurements are provided as one or more *AP measurement* items which identify the AP by its IEEE 802.11 SSID and provide RSSI as 1-byte value which has sufficient resolution for RSSI measurement accuracies typically obtained from IEEE 802.11 hardware.

Collision Avoidance

After sTags complete the RSSI collection cycle, they attempt to upload the data to SPLANs all at the same time. The IEEE 802.11 standard allows immediate transmission of a new packet on a clear channel which is usually fine because packets are normally generated by unsynchronized processes. Here, the processes are synchronized, so collisions inevitably occur.

A simple correction is for the sTags to assume that the WM was busy and immediately proceed with the back-off phase of the IEEE 802.11 MAC. A more serious problem still remains: the CW window sizing may not be optimal if done according to the IEEE 802.11 MAC procedure. If CW is too small, excessive collisions will occur; if it is too large, network capacity will be wasted. It might take several cycles of collisions for the back-off algorithm to adapt to a large tag count. Later, as the number of sTags awaiting TX decreases, unnecessarily large CW will be maintained.

The proposed solution is for a SPLAN to inform the sTag of the current number of registered tags, so that an optimal CW can be chosen. The details are illustrated in Figure 3.14. Both sTags are synchronized at the beginning of the timeline and the SPLAN begins to listen for RSSI data. Tags do not start transmission immediately despite sensing a clear channel in slot 1, but instead begin with a back-off procedure.

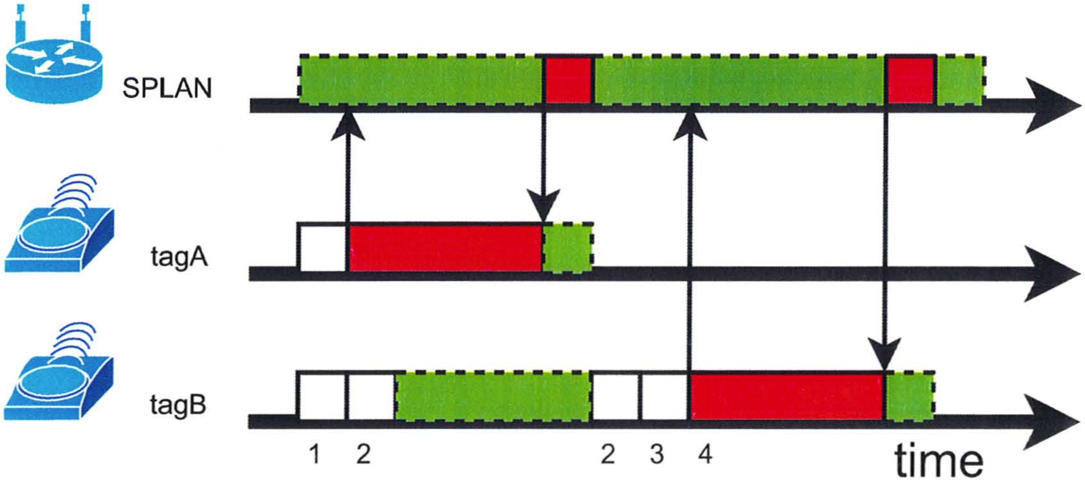


Figure 3.14: sTag RSSI upload contention

To further optimize the process, the SPLAN communicates the current number of registered tags (n) in each beacon packet as *DCF count* datum from Table 3.5. An optimal expression [→4.4.3.2] transforms n into CW_i which is used in accordance with DCF rules [→2.1.2.2] to set the CW. The example illustrated in Figure 3.14 assumes that $CW_i = 3$ is the best choice for $n = 2$, so the CW is set to $2^3 - 1 = 7$. From this set, tagA selects the second TS (of 7) while tagB selects the fourth TS (of 7). Thus, tagA transmits first in the second TS and receives an ACK from the SPLAN. Concurrently, tagB asserts a busy channel through the NAV, so it suspends backoff countdown. It resumes after tagA has finished transmitting, and tagB begins TX in TS 4. Note that SPLAN waits a while longer for more transmissions, as there may be more sTags present than the n count from the last cycle.

Chapter 4

System Models

Analytical expressions and simulation designs to approximate component power consumption as a function of model parameters and variables are developed in this chapter. In all deployment scenarios considered there are three network components for which power consumption can be evaluated: the AP, the SPLAN and the IEEE 802.11 localization tag (pTag or sTag).

The AP's behaviour is assumed to be unmodified or modified only slightly, hence AP PS opportunities are limited. Since the APs also handle standard IEEE 802.11 communications such as VoIP or data traffic, any PS scheme would also have to be compatible with these services. Jointly this is difficult to realize, while separately these issues has been researched before. Finally, the APs are assumed to be connected to wired data and power infrastructure where the cost and convenience values associated with PS are slight. Therefore, plain AP PS is not considered in this thesis.

The SPLAN's PS is important due to its high photovoltaic system cost. The ability of SPLAN to synchronize to pTag chirp cycles and the ability of sTags to register in SPLAN's LIs makes the PS considerations worthwhile.

The Tag's PS is also important since tags use a primary battery that needs to be replaced when depleted. Replacement labour is costly, so the batteries should last as long as possible. Hence, the proposed algorithms consider the tag's energy to be the most costly, but a sacrifice of a small amount of tag's energy for a significant gain in SPLAN's energy may be acceptable.

4.1 General Notation and Variables

In this analysis the device power consumption is modelled by a family of parameters with the general notation X_{state}^{device}. The notation's X is one of: **D**—the *duration* of the state in seconds calculated using analytical expressions or simulations from the model's parameters, **P**—the *power* usage of state in watts obtained or approximated from hardware specification datasheets, and **E**—the *energy* usage in joules for a specific event such as sTag registration. The notation's superscript indicates the type of device being modelled: either **T** for *tag* (sTag of pTag) or **S** for *SPLAN* (smart or plain mode). The notation's subscript indicates the state being modelled: **alphabetic** designations refer to normal states (e.g. *RX* is reception), while **numeric** designations subdivide similar state categories (e.g. *s1*/*s2* are light/deep sleep states). Transition states, such as that between light sleep and idle states, are indicated by an arrow, e.g. **s1** → **i**. The subscript **A** indicates an *average* state, i.e. the calculation's end result for a particular event.

Other frequently used parameters include n—the *number* of tags in the system, e.g. pTags synchronized to a SPLAN or sTags registered to a SPLAN. A related constant is n_{max}—the maximum number of tags that can be supported by the localization system. Various *intervals* are indicated by the letter i and include the *localization* (i_l), the *synchronization* (i_s) and the *registration* (i_r) intervals. The *localization interval* has dual meaning: for a plain tag system it refers to the interval between tag chirps while for a smart tag system it is the interval between successive localization beacons emitted by the SPLAN. The *synchronization interval* applicable to the pTag system refers to the interval between SPLAN's queries to an AP to obtain a new tag list. An equivalent interval in the sTag system is the *registration interval* which is the interval between transmission of successive registration beacons. Finally, c is the *chirp* packet count for a pTag, i.e. the number of packets sent in a single chirp event, while m represents *mobility* defined as the average number of seconds a tag remains in the vicinity of a SPLAN.

4.2 Packet Transmission Notation

Many approximations require estimating the packet's transmission duration ($T_{rate,framing}^{size}$), to calculate the amount of energy used during a transmission. Rather than developing a complete expression each time it is needed, the following framework provides the answer based on the following input parameters: the *size* of the user data being transmitted in bytes, the *rate* of TX in Mbit/s, and the *framing* type used: *raw* (OSI L2) or *cooked* (UDP encapsulated).

Table 4.1 summarizes the required framing object sizes. In IEEE 802.11 DS or HR/DSSS networks the physical layer convergence protocol (PLCP) header is always transmitted at $R =$ 1Mbit/s using DBPSK modulation for all packets with data portion rates (r) of 1, 2, 5.5 and 11Mbit/s, as shown in Reference [69] Figure 12-14 and Figure 12-17.

Table 4.1: IEEE 802, IEEE 802.11b, and IP/UDP packet framing

constant name		length	max speed	description	Reference
short	long (MATLAB)	(bytes)	(Mbit/s)		
F_{p1}	$fr\_plcp\_preamble$	18	1	PLCP preamble	[69] Figure 12-14
F_{p2}	fr_plcp_header	6	1	PLCP header	[69] Figure 12-14
F_{802}	$fr\_ieee80211\_header$	24	11	IBSS data frame	[69] Figure 4-8
F_S	fr_snap_header	6	11	SNAP	[69] Figure 3-13
F_T	fr_type_header	2	11	IP/ARP type	[69] Figure 3-13
F_I	fr_ip_header	20	11	IP header	[70]
F_U	fr_udp_header	8	11	UDP header	[71]
F_d	$fr\_ieee80211\_data$	variable	11	user data	
F_c	$fr\_ieee80211\_fcs$	4	11	frame checksum	[69] Figure 4-8

The maximum size of the packet's data portion (F_d) is $fr\_ieee80211\_data\_max = 2312$ bytes, as shown in Reference [69] Figure 4-8. Usually, it is further restricted to 1500 bytes for compatibility with Ethernet networks (e.g. RSSI data uploading to a localization server), but the full size can be used for communication between two IEEE 802.11 STAs (e.g. SPLAN/AP).

User data in some packets such as a legacy pTag chirp is encoded directly into the packet's data section. This is indicated by the subscript r (raw) in the model and the corresponding F_F parameter (framing size) is set to 0. For sTags which use UDP encapsulation, $F_F = F_S + F_T + F_I + F_U = 6 + 2 + 20 + 8 = 36$ bytes, and is indicated by the c (cooked) subscript.

A packet's transmit duration is:

$$T^{s}_{r,f} = \frac{F_{p1} + F_{p2}}{R} + \frac{F_{802} + F_F + s + F_c}{r}$$

For example, the duration of a 25-byte UDP packet transmitted at 11Mbit/s is $T^{25}_{11,c}$.

In the pTag system, only the raw chirp packets are used for pTag/SPLAN communication; the tag's status data are encoded directly into the packet's $fr\_ieee80211\_data$ portion without further framing. The newly proposed design for the sTag system could adapt that approach in the interest of efficiency (reduced packet size), however IP/UDP encapsulation will be used instead, in the interest of standardization. While IP/UDP headers are not strictly necessary for this application since the packets are non-routeable, having unformatted packets on the network is contrary to best network engineering practises. The most serious issue is that raw packets lack identification. A network diagnostic tool is unable to present them in a sensible way; even the target system has no reliable way of identifying such packets for localization processing.

UDP encapsulation solves this problem by specifying a target port number which can be globally registered for a specific purpose. For example, DHCP[72] uses officially registered UDP ports 67 and 68. A network administrator wishing to observe only DHCP transactions uses a filter which specifies these port numbers. Additionally, a DHCP server application can use the standard operating system interface (sockets) to bind port 67. To process raw packets instead, a DHCP server would have to bind to the raw network interface and inefficiently filter all packet by itself. For sTag localization transactions, UDP source port number *0x4C43* and UDP destination port number *0x4C53* are proposed in the interest of standardization. The data portion of the packet is modelled after *DHCP options* parameters, and is shown in Table 4.2.

Table 4.2: Localization packet's data portion structure

item	length [bytes]	description
A	8	magic number confirming the packet as IEEE 802.11 tag communication
B	1	command code identifier
C	1	command argument length [bytes]
D	variable	command argument (variable length)
E	[zero or more repetitions of item sequence B–C–D]	

The fundamental data length is $F_{pm} = 8$ bytes, while each command is composed of $F_{pc} = 1 + 1$ bytes and the size of argument (if any) up to 255 bytes.

Apart from the magic number, there are no fixed parts in the packet; traditionally they are included for accelerated processing in hardware. However, there is no need to develop specialized hardware just to decode localization packets given their relatively low volume. Thus, variable command sections are better suited for this purpose providing greater flexibility and future extensibility as additional command sections can be added while being ignored by current decoders.

4.3 Plain Tag System Model

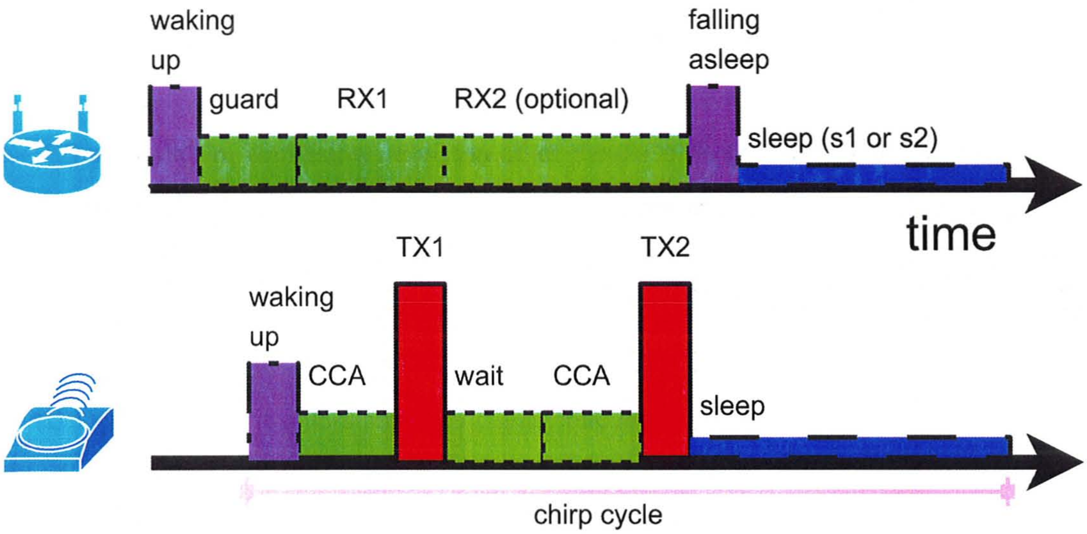


Figure 4.1: Power states of pTag and SPLAN in plain tag system localization

4.3.1 pTag Power Consumption in Localization

Four pTag activity states are illustrated in Figure 4.1: waking, receiving, transmitting, and sleeping. In the sleep state, the tag's circuits are powered off except for the crystal timer counting down to the next wake-up event. In the receive mode, the tag's radio is turned on and packets encoded within the demodulation capabilities of the tag (IEEE 802.11 1, 2 Mbit/s) are decoded to update the NAV, while packets with unsupported modulation are detected as power spikes during a CCA. The transmission of a chirp packet at 1, 2 Mbit/s is the state with the highest power.

Practical values of power consumption were obtained from G2 Microsystems tag G2C501[73] shown in Table 4.3 and Tag4M[74] shown in Table 4.4. These demonstrate a good correspondence between the states implemented in typical hardware and those needed for evaluation in this thesis.

Table 4.3: Power states of the G2C501 tag

state	power usage	duration
transmitting	700mA	4.7ms
receiving	60mA	90ms
deep sleep	100μW	1–500s
waking	25mA	52ms

Table 4.4: Power states of the Tag4M tag

state	power usage
boot-up	25mA
sleep	4–10μA
IEEE 802.11 RX	60mA
IEEE 802.11 TX	700mA

4.3.1.1 pTag State Durations

The durations given in Table 4.3 are applicable to the G2C501 tag acting as a wireless sensor node. These durations provide a useful starting point, but exact timings specific to IEEE 802.11 pTag localization are needed instead. They are derived in this section.

The tag's waking-up transition is shown to be nearly instantaneous in Reference [73] so the duration of the waking-up state is set to zero, and is not considered further. The first state that does need to be considered is the CCA state with the corresponding duration designated D^T_{CCA}. The CCA's duration varies depending on network activity level: with no network activity, the CCA procedure terminates after a set duration ($l = 100\mu s$ for AeroScout) and a clear channel status is declared. The probability of encountering a clear channel is approximately (1-U) where U is the network utilization, provided that the sampling interval is much smaller than the typical packet size. A typical 500 byte packet transmitted at 11Mbit/s takes $T^{500}_{11,c} \approx 284\mu s$. This is the same order of magnitude as the CCA interval, so the resulting approximation is of moderately fair quality.

If the CCA is unsuccessful (with probability U), the initial CCA interval is 0 length, as the interfering packet was most likely already being transmitted when the CCA was initiated. The next phase is the random back-off which takes from 0 to $2^i - 1$ TSs where $i = 5, 6, 7, 8, 9, 10, 10, 10, \ldots$ (for each iteration) and the TS for a 1–2Mbit/s DS PHY lasts $slot\_time = 20\mu s$ (Reference [69] Table 12-4). The average time spent in back-off is $b_i = (slot\_time)\frac{2^i-1}{2}$. The CCA duration can be approximated by: $D^T_{CCA} = (1-U)l + U(b_5 + (1-U)l + U(b_6 + \ldots$

Replacing $b_i + (1-U)l$ by A_i for clarity:

$$\begin{aligned} D^T_{CCA} &= (1-U)l + U(A_5 + U(A_6 + U(A_7 + U(A_8 + U(A_9 + U(A_{10} + U(A_{10} + \ldots \\ &= (1-U)l + UA_5 + U^2A_6 + U^3A_7 + U^4A_8 + U^5A_9 + U^6A_{10} + U^7A_{10} + \ldots \\ &= (1-U)l + UA_5 + U^2A_6 + U^3A_7 + U^4A_8 + U^5A_9 + A_{10}(U^6 + U^7 + \ldots) \end{aligned}$$

Simplifying using the sum of infinite geometric series identity, ($1 + r + r^2 + r^3 + \ldots = \frac{1}{1-r}$) the CCA's duration is:

$$\begin{aligned} D^T_{CCA} &= (1-U)l + UA_5 + U^2A_6 + U^3A_7 + U^4A_8 + U^5A_9 + \ldots \\ &+ A_{10}(\frac{1}{1-U} - (1 + U + U^2 + U^3 + U^4 + U^5)) \end{aligned}$$

A representative value of background network utilization U_b was chosen to be 0.30 for further analysis. Since the tags themselves also contribute to network utilization, the complete U is obtained by adding the tag chirp transmission time for all tags in the system, considering all packets within a chirp: $U = U_b + D^T_{TX}(c)(n)(1/i)$.

The chirp's transmission time depends on the packet length and the data rate. Typical lengths of the packet's data portion are 0 bytes for the PanGo tag and 34 bytes for the AeroScout tag. The supported data rates are 1 or 2Mbit/s; higher data rates are not used in plain tag systems to ensure maximum transmission range. The packet transmission time is calculated based on AeroScout defaults where $chirp\_size = 34$ bytes and $chirp_data_rate = 1Mbit/$s.

If the tag is configured to emit multiple packets per chip cycle a wait interval ($D^T_{s1} = 512$ms) follows each transmission except the last. In c repetitions of a single chirp packet transmission sequence there are c TX intervals, c CCA intervals and $(c-1)$ wait intervals. Hence the pTag's complete active interval is $D^T_{chirp} = c(D^T_{CCA} + D^T_{TX}) + (c-1)D^T_{s1}$. Following the activity intervals, the pTag enters the sleep mode which lasts $D^T_{s2} = i - D^T_{chirp}$. The aggregate average power consumption for a pTag can thus be expressed as:

$$P^T_A = \frac{c(P^T_r D^T_{cca} + P^T_t D^T_{tx}) + (c-1)P^T_{s1}D^T_{s1} + P^T_{s2}D^T_{s2}}{i_l}$$

While the average power consumption is a reasonable metric for the SPLAN (for photovoltaic resource allocation), a more practical metric for a tag is its expected lifetime. The typically battery model used[73] in tags is CR123A, a 3V Lithium primary cell with 1300mAh typical capacity[75]. Due to its long shelf life, the battery's self-discharge characteristics can be ignored for an expected maximum tag lifetime of 5 years, in relation to its 10 year shelf life. Therefore, $battery\_life = battery\_capacity/P^T_A$ where $battery\_capacity$ is in watt seconds (joules) and P^T_A is in watts.

4.3.2 SPLAN Power Consumption in Localization

The SPLAN hardware is assumed to be similar to a typical IEEE 802.11 AP hardware, with a trivial addition of a low power quartz crystal timer to support wake-up triggering. Thus, the three SPLAN components to consider are: the quartz crystal timer, the IEEE 802.11 network interface controller (NIC), and an x86/ARM/MIPS single-board computer (SBC) consisting of a CPU, random access memory and flash memory. While the choice of actual available hardware is irrelevant for this thesis, it is important to understand power management capabilities of such hardware to obtain an accurate analysis.

Quartz Crystal Timer

A representative example of a quartz crystal timer is the Maxim DS3231[76], "a low-cost, extremely accurate I^{2}C real-time clock with an integrated temperature-compensated crystal oscillator and crystal." Power consumption of this unit is $P^{St} = 726\mu\text{W}$, calculated from the provided specifications of $I_{CCA} = 200\mu\text{A}$ at $V_{CC} = 3.63\text{V}$. The clock frequency stability is $c_e = 3.5\mu$ (i.e. parts per million) over an industrial temperature range of -40°C to +85°C.

IEEE 802.11 Network Interface Controller

The authors of Reference [77] propose the use of a Bluetooth NIC as a wake-up mechanism for an IEEE 802.11 primary NIC—a *Cisco Systems PCM 350* NIC (100mW, 11Mbit/s, 3.3V, PC Card form factor). The NIC's activity matrix for its SS1 and SS2 sleep states is shown in Table 4.7. Both the power consumption (shown in Table 4.5) and the corresponding transition latency (shown in Table 4.6) were measured for this card. The sleep mode transition latencies are important for the analysis in this thesis because they participate in a trade-off between reduced power consumption achieved by entering a sleep state and the corresponding latency cost. On the other hand, the transition latency between TX/RX modes of a typical IEEE 802.11 NIC is 6–30μs (as seen in Reference [78] Section 3.1) and thus negligible for this analysis.

Table 4.5: IEEE 802.11 NIC power states

state	constant	power [mW]
TX	P_t^{Sw}	1800
RX	P_r^{Sw}	1520
Idle	P_i^{Sw}	1310
PSP		390
SS1	P_{s1}^{Sw}	230
SS2	P_{s2}^{Sw}	20
off	P_{off}^{Sw}	0

Table 4.6: IEEE 802.11 NIC transition latency matrix [ms]

↓from to→	Idle	SS1	SS2
Idle	0	20	1710
SS1	20	0	N/A
SS2	810	N/A	0

$D_{i\to s1}^{Sw} = 20\text{ms}$, $D_{s1\to i}^{Sw} = 20\text{ms}$,

$D_{i\to s2}^{Sw} = 1710\text{ms}$, $D_{s2\to i}^{Sw} = 810\text{ms}$.

Table 4.7: IEEE 802.11 NIC sleep states

state	active component	
	radio	electronics
idle	on	on
SS1	off	on
SS2	off	off

Single Board Computer (SBC)

The power consumption and latency of a typical system used for an IEEE 802.11 AP (and hence SPLAN too) are enumerated in References [79, 80] where Compaq Itsy, a system with a "206MHz StrongARM Processor (SA-1100), 32M bytes main memory, 32M bytes NOR flash memory and 320x200 resolution 16-grey level LCD panel" is examined. The relevant power consumption values are shown in Table 4.8. Note that these values are much lower than those of the NIC; the idle power for Itsy is 101mW while the idle power for the PCM-350 NIC is 1310mW.

Table 4.8: SBC power consumption

state		power
constant	description	[mW]
P_{s2}^{Sb}	deep sleep	4.58
P_{s1}^{Sb}	normal sleep	7.40
P_{i}^{Sb}	idle	101
P_{b}^{Sb}	busy	826

Power (Table 4.9) and latency (Table 4.10) data for the StrongARM SA-1100 CPU used in the Itsy SBC were obtained from Reference [79]. The power consumption values are in agreement with those of the whole system using the SA-1100 CPU, e.g. in the SBC IDLE state 50mW is used by the CPU and the remaining 51mW by other components. In the SBC sleep state, the CPU contribution is negligible and 7.40mW is used for memory refresh.

The CPU state transitions are all very short except for SLEEP→RUN transition, i.e. waking-up. The corresponding MATLAB constants used in the analysis are $D_{i\rightarrow s1}^{Sb}$ and $D_{s1\rightarrow i}^{Sb}$ for duration of SPLAN's transition from idle→sleep1 and sleep1→idle states, respectively. Since the SA-1100 does not offer transition from SLEEP to IDLE, the transition was estimated as $D_{s1\rightarrow i}^{Sb} = D_{s1\rightarrow r}^{Sb} + D_{r\rightarrow i}^{Sb}$.

Table 4.9: CPU power consumption

state	power [mW]
SLEEP	0.16
IDLE	50
RUN	400

Table 4.10: CPU state switching latency

↓ from to→	SLEEP	IDLE	RUN
SLEEP	0	N/A	160ms
IDLE	90μs	0	10μs
RUN	90μs	10μs	0

For an even greater PS, it is possible[80] to use a deep sleep state designated S2, distinct from the light sleep state S1 discussed above. The use of the S2 state requires storage of the operating system's memory content to flash memory, with the aid of run-length encoding compression. While the latency is an order of magnitude larger as shown in Table 4.11, the SBC's power consumption decreases from 7.40mW to 4.58mW: a 38% reduction. The corresponding MATLAB analysis constants for these transitions are: $D^{Sb}_{i \to s2} = 3900$ms and $D^{Sb}_{s2 \to i} = 1200$ms.

Table 4.11: SBC state switching latencies for S2 deep sleep state [seconds]

↓from to→	idle	deep sleep
idle	0	3.9
deep sleep	1.2	0

4.3.2.1 SPLAN Power States

Table 4.12: SPLAN power states

constant	SBC	NIC	timer	duration [ms]
P^S_{TX}	idle	TX	on	variable
P^S_{RX}	idle	RX	on	variable
P^S_{idle}	idle	idle	on	variable
$P^S_{i \to s1}$	busy	idle	on	20
P^S_{s1}	light sleep	SS1	active	variable
$P^S_{s1 \to i}$	busy	SS1	on	160
$P^S_{i \to s2}$	busy	idle	on	3900
P^S_{s2}	deep sleep	SS2	active	variable
$P^S_{s2 \to i}$	busy	SS2	on	1200

Based on the analysis of SBC and IEEE 802.11 NIC power states, the SPLAN power states (including two types of power saving states) are shown in Table 4.12. The constants are calculated as the sum of power usage of individual components in their appropriate power state, e.g. $P^S_{RX} = P^{St} + P^{Sw}_r + P^{Sb}_i$, etc. While it would be possible to control the power state of the IEEE 802.11 NIC separately from the SBC, there is no benefit of doing so for the SPLAN application. Since the SBC can enter a PS mode concurrently with the IEEE 802.11 NIC, the actual SPLAN transition duration is the maximum of the SBC's and the NIC's transition latencies: $D^S_{i \to s1} = max(D^{Sw}_{i \to s1}, D^{Sb}_{i \to s1})$, etc.

4.3.2.2 SPLAN State Durations

Please refer to Figure 4.1 showing SPLAN activities in chronological relation to those of the pTag.

The SPLAN initializes its hardware components in the waking-up state; its duration is fixed at either $D^S_{s1\rightarrow i}$ or $D^S_{s2\rightarrow i}$ with corresponding power usage of $P^S_{s1\rightarrow i}$ or $P^S_{s2\rightarrow i}$, respectively. The waking-up state must finish earlier than the projected beginning of sTag activity interval to account for clock inaccuracies. It is assumed that the systematic clock drift is already corrected [→A.4] and all that remains is Gaussian distributed error. To provide sufficient reliability, the guard interval needs to be several times larger ($c_m = 3$) than the standard deviation of the sleep time calculated from the clock error constant of $c_e = 3.5\mu$. Thus, the guard interval duration is $D^S_{guard} = (c_m)(c_e)(D^T_{s2})$, while its power consumption is P^S_{RX}.

The SPLAN's first chirp RX interval duration is equivalent to the pTag's CCA and TX intervals combined. If the reception is successful, the SPLAN can go back to sleep, while the pTag continues to transmit more chirp packets. However, there may be some cases where SPLAN does not observe any valid packets in the primary RX interval. In that case it attempts to receive a secondary or tertiary chirp packet.

To calculate the probability that one of more of these RX backup intervals is necessary, the causes of failure of reception of the primary chirp packet must be considered. The *late synchronization* problem where a chirp packet appears before the guard interval starts should be of no concern with the correct choice of c_m. Signal fading (due to changes in environment conditions) and interference factors (such as microwave ovens) are difficult to quantify, but generally should be at a negligibly low level due to the high reliability of low data rate packets. A more serious problem is *simultaneous TX by a remote node*, i.e. the classic hidden terminal problem. In this case not all external transmission will overpower the tag's packet; the interferer's distance has to be close enough to SPLAN but far away from the tag. As a reasonable worst case approximation half of all network transmission will be assumed to be interfering: $U = \frac{1}{2}U_b$. The interval duration is thus iteratively calculated as $D^S_{rx2} \leftarrow D^S_{rx2} + U^{j-1}(D^T_{s1} + D^T_{cca} + D^T_{tx})$ where j is the iteration count beginning at 2, corresponding to the second chirp packet.

After successfully receiving a packet on the first or subsequent tries, the SPLAN enters the falling-asleep transition state; its duration is fixed at either $D^S_{i\rightarrow s1}$ or $D^S_{i\rightarrow s2}$ with corresponding power usages of $P^S_{i\rightarrow s1}$ or $P^S_{i\rightarrow s2}$, respectively. The time left to SPLAN for sleeping in a given chirp interval is the difference between i_l and the sum all the durations above. The presence of multiple pTags is modelled by assuming that each one is allocated a separate SPLAN LI.

$$D^S_{sleep} = i_l - n(D^S_{s\to i} + D^S_{guard} + D^S_{rx1} + D^S_{rx2} + D^S_{i\to s})$$

This is a reasonable approximation for low temporal density of tags, but as more tags are added, some of their chips may randomly fall so close together that they might be more optimally served by an extended SPLAN listening cycle rather than two cycles. This optimization is not considered in the analytical approximation of the plain tag system, so the above approximation represents a lower bound on performance. Finally, the aggregate power consumption of the SPLAN is calculated by aggregating the durations and power consumptions of each of the intervals:

$$P^S_A = \frac{n\left(P^S_{s\to i}(D^S_{s\to i}) + P^S_{RX}(D^S_{guard} + D^S_{rx1} + D^S_{rx2}) + P^S_{i\to s}(D^S_{i\to s})\right) + P^S_s(D^S_{sleep})}{i_l}$$

SPLAN Sleep Mode Switching Thresholds Considerations

While the SPLAN could use a specific PS mode exclusively, switching of modes based on the observed load (n) is more efficient, as shown in Figure 5.4. The applied load is more fundamentally related to the interval i_e between subsequently scheduled tag chirps. When i_e is long, the SPLAN can use either the S2 or the S1 PS mode; when it is short, the S0 mode (RX) should be continued. To determine the appropriate switching thresholds, the power usage for these three modes is calculated using the expressions shown in Table 4.13.

Table 4.13: Expressions for SPLAN sleep mode switch threshold analysis

mode	expression for power usage in joules
S0 PS	$E_{s0} = P^S_{RX}(i_e)$
S1 PS	$E_{s1} = P^S_{i\to s1}D^S_{i\to s1} + P^S_{s1}(i_e - D^S_{i\to s1} - D^S_{s1\to i} - D^S_{guard}) + P^S_{s1\to i}D^S_{s1\to i} + P^S_{RX}D^S_{RX}$
S2 PS	$E_{s2} = P^S_{i\to s2}D^S_{i\to s2} + P^S_{s2}(i_e - D^S_{i\to s2} - D^S_{s2\to i} - D^S_{guard}) + P^S_{s2\to i}D^S_{s2\to i} + P^S_{RX}D^S_{RX}$

4.3.3 pTag Synchronization

The localization performance of the two proposed [→3.1.2] pTag synchronization methods considered in this section. In the legacy-AP compatible **pSyncL** method, the AP-resident tag database is queried by a SPLAN and the chirp timestamps for all the tags are obtained. At least two polling cycles are required to obtain i_l, calculated as the difference between two timestamps; more cycles are required if $i_l > i_s$. On the other hand, the **pSyncF** method requires *firmware* modification of the AP and the pTag. The AP timestamps a received packet which contains i_l, and forwards it to the SPLAN during the next SPLAN/AP-type LI.

The primary purpose of a SPLAN in a pTag environment is localization accuracy enhancement which is possible when the SPLAN successfully receives a pTag chirp packet; in that case the localization server is able to use a RSSI vector augmented by the SPLAN measurement. The degree of location accuracy improvement can be expressed as the reduction of the difference between the pTag's true and computed locations. It might be interesting to analyze localization assistance using this metric, as it would give direct and realistic performance improvement values. However, it would require the use of a specific localization algorithm simulated for a specific SPLAN/AP/ pTag geometry, and thus the fundamental performance metric would be obscured and influenced by unrelated layers. A more effective approach is to quantify localization improvement from the perspective of the communication algorithm. A successful localization assistance event shall be defined as the SPLAN's successful reception of at least one packet from the pTag's chirp.

There are three effects that can cause the localization assistance metric to be less than unity. **Interference** can cause the loss of a single chirp packet but is expected to cause negligible deterioration in localization assistance due to the use of multiple packets per chirp. For the same reason, SPLAN waking up from sleep **too late** is also of no concern. Unless i_l is very long, the SPLAN's guard interval will be significantly shorter than the inter-packet delay, so the SPLAN will receive the next chirp in sequence. Even missing the first chirp packet would be a rare occurrence since c_m is selected such that statistically only a small fraction of chirps are missed.

It is the **lack of synchronization** which is the primary source of localization assistance degradation. A pTag which is not synchronized with a SPLAN emits chirps during the time the SPLAN is sleeping, so there is no opportunity for the SPLAN to provide assistance. Lack of synchronization may occur when a **pTag is first turned on** or reset and a new chirp phase is generated. However, with the typical tag lifetime of a few months or years, that *initial lack* of synchronization is of little importance because it represents only a few missed localization opportunities out of potentially hundreds of thousands. As long as the pTag does not move, or moves within the confines of a SPLAN's reception area, the synchronization state will persist. Only when the pTag moves to a **new service area** it be unassisted until synchronization is reestablished. It is this effect that will be quantified in this section.

The number of missed tag chirps before the pTag is synchronized is dependent on the relative durations of the pTag chirp interval (i_l) and SPLAN synchronization polling interval (i_s). For specific situations their phase offset also plays a role, however for simplicity the worst case scenario (representing performance's lower bound) will be considered, where the first tag chirp occurs just after a SPLAN synchronization poll.

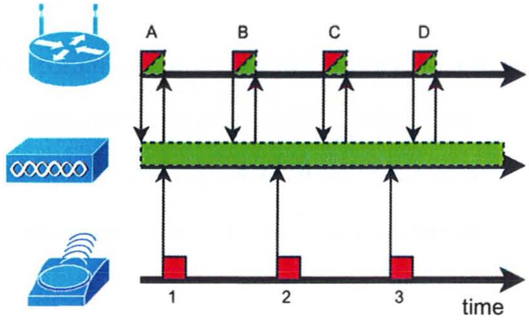


Figure 4.2: pTag synchronization missed chirp analysis—fast polling

Figure 4.2 illustrates a situation where $i_s < i_l$. The SPLAN starts with poll-A to find new tags: there are none. Shortly after, a pTag new to this service area emits its first chirp-1. Because $i_s < i_l$ the next poll occurs before tag's chirp-2 at time B. Under the *pSyncF* approach, chirp-1 contained i_l so the SPLAN obtained the required synchronization data pair at poll-B. At that point the tag is synchronized and further chirps are received directly by the SPLAN. However, with the *pSyncL* approach, the reception of pTag's chirp-2 is also necessary to obtain i_s. Once again, because $i_s < i_l$, at least one SPLAN poll (poll-C) is guaranteed to occur in between each chirp. Hence, the SPLAN is guaranteed to be in a synchronized state before pTag's chirp-3. In summary, there will be 1 missed chirp under *pSyncF* and 2 missed chirps under *pSyncL*.

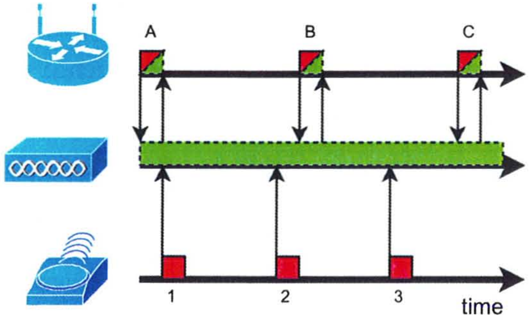


Figure 4.3: pTag synchronization missed chirp analysis—slow polling

Figure 4.3 illustrates a situation where $i_s > i_l$. Worst-case timing is again assumed in that SPLAN poll-1 happens just before pTag's chirp-1. The synchronization will be completed at poll-B for *pSyncF* or poll-C for *pSyncL*, just as in Figure 4.2.

The number of pTag chirp intervals that fit into the required SPLAN poll interval is $\lfloor ji_s/i_l \rfloor$ where $j = 1$ for *pSyncL* and $j = 2$ for *pSyncF*. The number of intervals is related to the number of endpoints (pTag chirps) as $endpoints = intervals + 1$, e.g. 2 intervals shown have 3 endpoints: start, middle and end. Hence, the number of missed chirps is $chirps\_missed = \lfloor ji_s/i_l \rfloor + 1$. The total number of chirps emitted by a pTag in single visit to a specific SPLAN coverage area is $chirps\_total = \lfloor m/i_l \rfloor$. The localization assistance metric expression is therefore:

$$loc\_assistance = 1 - \frac{\lfloor ji_s/i_l \rfloor + 1}{\lfloor m/i_l \rfloor}$$

4.4 Smart Tag System Model

The sTag and SPLAN energy requirements and transition latencies are modelled using values from the pTag analysis, as it is assumed that the same hardware is used for both. The functionality differences (which affect state durations) are implemented purely in the firmware.

4.4.1 Smart Registration Power Analysis

sTags that have been turned on for the first time or are re-entering a SPLAN service area after a long absence have no knowledge of the beacon phase (registration state R0). Since an sTag commences reception at a random phase, half of the SPLAN synchronization beacon interval will pass on average before the sTag receives the beacon. In the worst case, the entire interval will pass: $E^T_{r\_initial} = (P^T_r)(i_r)$.

SPLAN Energy Usage

In Table 4.14, n^S_r is the number of sTags registering with SPLAN during a single event. The n^S_r value can be approximated by considering the m parameter which specifies the average dwell time of a single tag at a specific SPLAN. The average number of registrations generated by a population of n tags is therefore $n^S_r = ni_r/m$.

Table 4.14: SPLAN power states during an sTag registration event

ID	count per event	power state	duration	description of SPLAN's action
A	1	$P^S_{s1\rightarrow i}$	$D^S_{s1\rightarrow i}$	wake-up from sleep
B	1	P^S_{TX}	$D^T_{CCA} + T^{sync\_beacon\_size}_{1Mbit/s}$	TX of registration beacon at 1Mbit/s
C	n^S_r	P^S_{RX}	$D^T_{CCA} + T^{sync\_packet\_request\_size}_{11Mbit/s}$	RX of registration request
D	n^S_r	P^S_{TX}	$D^T_{CCA} + T^{sync\_packet\_response\_size}_{11Mbit/s}$	TX of registration response
E	1	P^S_{RX}	$(cca\_multiplier)D^T_{CCA}$	ensure no pending registrations
F	1	$P^S_{i\rightarrow s1}$	$D^S_{i\rightarrow s1}$	enter sleep

The SPLAN energy usage per registration event is $E_r = \sum(Table\ 4.14)$, while the *average* SPLAN power usage attributed to registration alone is $P^S_r = \sum(Table\ 4.14)/i_r$. For the purpose of analysis in this document a *table sum* is defined as the sum of products of all items per row in a Table, i.e. $\sum^F_A$(count per event)(power state)(duration).

sTag Energy Usage

The power states of a registering sTag are illustrated by *sTag2* in Figure 3.8 and are summarized in Table 4.15. The average power usage attributed to sTag registration is $P_r^T = \sum(\mathit{Table}\ 4.15)/m$.

Table 4.15: sTag registration power states

ID	power state	duration
A	P_{RX}^T	D_{guard}^T
B	P_{RX}^T	$D_{CCA}^T + T_{1Mbit/s}^{sync\_beacon\_size}$
C	P_{RX}^T	$(n_r^T/2)(D_D + D_E)$
D	P_{TX}^T	$D_{CCA}^T + T_{11Mbit/s}^{sync\_packet\_request\_size}$
E	P_{RX}^T	$D_{CCA}^T + T_{11Mbit/s}^{sync\_packet\_response\_size}$

(A) The sTag guard time is calculated as for the plain SPLAN: $D_{guard}^T = (c_m)(c_e)(base\_duration)$. In the previous case, *base_duration* was the SPLAN's sleep time. Here it is the duration of an sTag's travel time from the previous SPLAN to the current one. Assuming continuous coverage, the maximum amount of time that an sTag will be out of contact is $base\_duration = i_l + i_r$.

(B) A registration-type beacon is received from a SPLAN.

(C) If more than one sTag is attempting to register, a DCF queue will form. The length of the entire queue will be n_r^T, similar to the SPLAN's n_r^S variable with a slight difference in that only an sTag *other* than the one currently considered need to be counted, so $n_r^T = (n-1)i_r/m$. While this is the average length of the queue, a specific tag can be serviced in any position of the queue, and on average at the half-point of the queue. The service duration is equivalent to the sum of durations of the registration request and registration response transactions (items D and E). Therefore, $D_C = (n_r^T/2)(D_D + D_E)$ as shown.

(D) The sTag transmits a registration request.

(E) The sTag receives a registration response.

4.4.2 Smart Localization Power Analysis

The smart localization transaction illustrated in Figure 3.11 is presented and discussed in three phases: localization beacon transmission, RSSI collection and RSSI upload. Phase durations from points of view of the SPLAN and the sTag are considered.

4.4.2.1 SPLAN Localization Power Usage

The beacon TX power usage is enumerated in Table 4.16.

Table 4.16: SPLAN power states in smart localization during *beacon TX*

ID	#	power state	duration
A	1	$P^S_{s1\rightarrow i}$	$D^S_{s1\rightarrow i}$
B	1	P^S_{TX}	$D^T_{CCA} + T^{loc\_beacon\_size}_{1Mbit/s}$

The RSSI collection duration depends on the scanning mode used by the sTag. There are two modes available as illustrated in Figure 3.12.

An sTag using the **legacy collection mode** tunes to each of the channels in its regulatory domain channel set (11 channels in the IEEE 802.11 2.4 GHz band) and receives beacons for the standard IEEE 802.11 beacon duration of 100ms which takes up to a total of $D_{Cl} = 1.10$s. During this time, the SPLAN could either remain in receive mode or switch to a sleep mode. According to Figure 5.5, the threshold at which sleeping becomes more efficient is 190ms which is $\ll 1.10$s. Hence, the SPLAN shall sleep when waiting for sTag legacy scanning as shown in Table 4.17.

An sTag using the **assisted collection mode** obtains additional information from the SPLAN's localization beacon to tune its radio to the correct channel at the correct time to capture AP beacons more efficiently. One beacon interval is generally sufficient for a full capture, but in the unlikely case that there is a temporal overlap between two beacons, a second interval is required. It is unlikely that beacons corresponding to all 11 channel are overlapping, so the use of two intervals can be considered a worst-case scenario. Hence SPLAN will need to wait $D_{Ca} = 2(100\text{ms}) = 0.200$s. Since this is only slightly longer than the sleep efficiency threshold, the SPLAN shall remain in RX mode during this time as shown in Table 4.18.

Table 4.17: SPLAN power states in smart localization during *RSSI legacy collection*

#	state	duration
1	$P^S_{i\rightarrow s1}$	$D^S_{i\rightarrow s1}$
1	P^S_{s1}	$D_{Cl} - D^S_{i\rightarrow s1} - D^S_{s1\rightarrow i}$
1	$P^S_{s1\rightarrow i}$	$D^S_{s1\rightarrow i}$

Table 4.18: SPLAN power states in smart localization during *RSSI assisted collection*

#	state	duration
1	P^S_{RX}	D_{Ca}

The RSSI upload stage's power usage is shown in Table 4.19. Due to shared WM, there are n data uploading transmissions per LI seen by a SPLAN, regardless of the number specifically directed to that primary SPLAN. The sTag's RSSI data upload duration is modelled by a single instance of the $csma\_ca[n]$ function [→4.4.3] instead of n repetitions of the $(D^T_{CCA} + T^{rssi\_send\_size}_{11Mbit/s})$ expression used previously. The function computing D^T_{CCA} was appropriate for modelling a single unsynchronized TX with a certain amount of unrelated background traffic. In this case however there are n simultaneously pending related transmissions, so a new model was needed.

Table 4.19: SPLAN power states in smart localization during *RSSI upload*

id	#	power state	duration
A	1	P^S_{RX}	$csma\_ca[n]$
B	n	P^S_{TX}	$T^{rssi\_ack\_size}_{11Mbit/s}$
C	1	P^S_{RX}	$(cca\_multiplier)D^T_{CCA}$
D	1	$P^S_{i\to s1}$	$D^S_{i\to s1}$

The overall average power usage for smart SPLAN localization is expressed as a MATLAB vector where the two columns correspond to an sTag using legacy and assisted collection methods.

$$P^S_l = \frac{\Sigma(Table\ 4.16) + [\Sigma(Table\ 4.17)\ \ \Sigma(Table\ 4.18)] + \Sigma(Table\ 4.19) + P^S_{s1}(i_l)}{i_l}$$

4.4.2.2 sTag Localization Power Usage

Reducing power consumption of an sTag is an important objective because battery replacement is a labour intensive (hence costly) procedure. Figure 3.11 demonstrates that sTag activity throughout the localization cycle mirrors that of the SPLAN and also consists of three phases described below.

Beacon RX power usage is enumerated in Table 4.20. The D^T_{guard} is the required SPLAN/sTag coordination interval, analogous to the D^S_{guard} interval in the pTag system. Its length is proportional to the sTag's sleep time: $D^T_{guard} = (c_m)(c_e)(D^T_s)$.

Table 4.20: sTag power states in smart localization during *Beacon RX*

ID	#	power state	duration
A	1	P^T_{RX}	D^T_{guard}
B	1	P^T_{RX}	$D^T_{CCA} + T^{loc\_beacon\_size}_{1Mbit/s}$

RSSI Collection

As illustrated in Table 4.17 and Table 4.18 there are two scanning modes available to an sTag. While performance analysis of the assisted algorithms will be discussed subsequently [→4.4.4], for the sTag overall power evaluation it is sufficient to consider the energy usage of fixed intervals mirroring those of the SPLAN: $P^T_{RX}D_{Cl}$ for the *legacy mode* and $P^T_{RX}D_{Ca}$ for the *assisted mode*.

RSSI Upload

To upload RSSI data, the sTag has to first participate in CSMA/CA with its radio in the P^T_{RX} power state. The analysis of the duration of this interval is presented subsequently [→4.4.3] where the $csma\_ca[n, mode]$ function is developed that returns the total time spent by all the uploading sTags. The *mode* parameter is one of the three DCF access control modes considered: standard, extended or assisted.

An individual average sTag is serviced in the middle of the DCF queue, so $median[1:n]-1$ is the average number of prior services, while $(1/n)csma\_ca[n, mode]$ is the average duration a single sTag spends in service. Therefore, the average waiting time before service is:

$$csma\_ca\_tag\_wait[mode] = (median[1:n]-1)\left(\frac{csma\_ca[n, mode]}{n}\right)$$

The total power usage for *RSSI upload* is summarized in Table 4.21.

Table 4.21: sTag power states in smart localization during *RSSI upload*

id	#	state	duration
A	1	P^T_{RX}	$csma\_ca\_tag\_wait[mode]$
B	1	P^T_{TX}	$T^{rssi\_send\_size}_{11Mbit/s}$
C	1	P^T_{RX}	$T^{rssi\_ack\_size}_{11Mbit/s}$

The overall average energy usage for sTag localization is shown below, where the resulting matrix enumerates all four possible combinations of legacy and assisted RSSI collection combined with standard and assisted RSSI upload. The corresponding average power usage is $P^T_A = E^T_A / i_l$.

$$E^T_A = \sum Table\ 4.16 + \begin{pmatrix} (P^T_{RX})D_{Cl} & (P^T_{RX})D_{Cl} \\ (P^T_{RX})D_{Ca} & (P^T_{RX})D_{Ca} \end{pmatrix} + \begin{pmatrix} \sum Table\ 4.21[s] & \sum Table\ 4.21[a] \\ \sum Table\ 4.21[s] & \sum Table\ 4.21[a] \end{pmatrix} + P^T_{s2}(i_l)$$

4.4.3 RSSI Upload

4.4.3.1 Simulation Design

The upload of RSSI data presents an unusual challenge for DCF, as discussed in Section [→3.2.6]. The previously developed analytical CCA model (D^T_{CCA}) is not adequate for this situation, as it only takes background traffic into consideration. It also assumes a single tag transmission suitable for use in the pTag system where the tags are not synchronized with each other. Instead, the total time spent by all sTag transmitting RSSI data in a single upload cycle is needed.

A simulation was written in MATLAB to model the system as an array of sTags with two attributes: **int timer**—the number of TSs the sTag has left until its next transmission and **float retry**—the contention window sizing index (CW_i) which determines time upper bound on CW used to select TSs. There are three types of events handled in each simulation cycle.

A collision occurs when ≥ 2 sTags become ready to TX in the same TS. A time duration equal to packet TX time passes. For colliding tags, the **retry** counter is adjusted according to DCF, and a new random value for **timer** is generated based on the recalculated CW.

A transmission event causes time duration equal to packet TX time to pass. The transmitting tag is deleted from the system.

A time decrement occurs when no tags are ready to transmit. A tag with the smallest timer is found and timers on all tags are decremented by min_timer, while a time duration equal to $(min\_timer)(slot\_time)$ passes.

Following the standard DCF rules, the **retry** counter is adjusted after a collision. When an sTag experiences a collision for the first time, the **retry** counter is set to i_{min}. If the **retry** counter has already reached i_{max}, then it remains unchanged. In any other case, the **retry** counter is incremented by 1. Note that the **retry** counter never decreases in this simulation. Normally this happens when a packet is successfully transmitted, but here a successful TX causes the sTag to leave the simulation, so there is no need to update its counter.

In addition to the baseline DCF, two efficiency-enhancing modifications are proposed. They are incorporated in the simulation as conditional execution blocks. The simplest modification **extends** the maximum possible window size; i_{max} is increased to a new fixed value which accommodates a larger number of sTags more efficiently. A more refined approach is a dynamically **assisted** retry counter adjustment. Unlike a standard IEEE 802.11 STA, an sTag has a good knowledge of the current network conditions which are dominated by sTag RSSI uploads. An approximate number of sTags competing for the WM is broadcast in the SPLAN beacon packet. As sTags wait for an opportunity to transmit, they are aware of successful transmissions by other tags and can estimate the current total of competing sTags. Based on this data, CW_i is adjusted for maximum efficiency through a function used at the beginning of the simulation and after every successful packet transmission: $CW_i = (1/log2)(log[dcf\_opt[tag\_count\_current]])$.

4.4.3.2 Optimization

The necessary $dcf\_opt[]$ function was constructed through a convex optimization formulation with the objective to minimize the time spent (T) at each DCF cycle. T is composed of:

(A) w wasted TSs followed by a successful packet TX and

(B) w wasted TSs followed by an unsuccessful packet TX resulting in a collision.

During each TX cycle (A) always occurs, while (B) occurs with probability p, which should be as a low as possible. This is achieved by increasing CW_i which in turn increases the number of available TSs: $CW = 2^{CW_i} - 1$. However the number of wasted TSs w is also increased. The proposed optimization formulation finds the value of CW_i which balances both factors.

$$\begin{aligned} T &= (w(slot\_time) + D) + p(w(slot\_time) + D) \\ &= (1 + p)(w(slot\_time) + D) \end{aligned}$$

In this expression for T, the $slot\_time$ is an IEEE 802.11 MAC quantity fixed for a specific PHY. D is also a fixed quantity in the RSSI upload process and is calculated as $D = T_{11,c}^{rssi\_send\_size}$. Expressions for both w and p are needed. The expression for calculating the probability of a collision

$$p = 1 - (1 - 1/W)^{n-1}$$

was obtained from Reference [81] Equation 3 where W is the slot count and n is the node (here the sTag) count. In this expression, $1/W$ represents the probability that a STA will pick a specific TS for transmission, out of W TSs. The corresponding probability that it *does not* pick that TS is the complement of $1/W$ i.e. $1 - 1/W$. To avoid a collision all stations other than the first $(n - 1)$ must avoid that TS. The probability that a series of events occurs is calculated as the product of the individual event probabilities, i.e. $(1 - 1/W)(1 - 1/W)(\dots) = (1 - 1/W)^{n-1}$, which gives the complete probability that a collision will not occur for a set for n STAs. The complementary probability that at least one collision will occur is $1 - (1 - 1/W)^{n-1}$, which completes the derivation of the original expression. Note that it would have been arduous to consider the probabilities of a collision directly (i.e. that STAs *do* pick a certain TS), because a collision can be generated by a variable number (2–n) of STAs colliding simultaneously; these events would have to have been enumerated separately with consideration given to possible overlap.

The expression $w = P_i/(1 - P_i)$ was obtained from Reference [32] Equation 4 where $P_i = (1 - P_e)^n$ and $P_e = \frac{2}{W+1}$. Combining all the expressions yields the objective function of W:

$$T = (1 + 1 - (1 - 1/W)^{n-1}) \left(\frac{(1 - \frac{2}{W+1})^n}{1 - (1 - \frac{2}{W+1})^n} (slot\_time) + D \right)$$

The corresponding constraints on W are $W \in \mathbb{Z}$ and $W \geq 0$, because W represents the number of TSs. Since integral optimizations are difficult to solve, the $W \in \mathbb{Z}$ constraint was dropped. Instead, W is quantized after an optimal floating-point version is found; for a one-dimensional optimization with high resolution of W, the expected accuracy loss is minimal. To find the optimal value, the MATLAB's standard $fminbnd[]$ solver was used with an interval bound of $[0, 10^5]$ and a default value ($TolX = 10^{-4}$) for the termination tolerance parameter.

4.4.3.3 Limiting sTag Count per Listening Interval

The sTag's energy consumption can be reduced even further by limiting the number of sTags registered in a single LI. Instead of allowing all n_{max} sTags with the same QoS into a single LI, only $interval\_max$ sTags are allowed, where $interval\_max < n_{max}$. Any sTags registering with the SPLAN when LI is full trigger automatic creation of an additional LI. Note that this optimization only applies to sTags with the same QoS i_l; sTags with different i_l values are assigned to different LIs by necessity. For simplicity, no provisions are made to actively re-balance the number of tags across different compatible LIs, but any new tags are assigned to a LI with the lowest tag population.

To determine a suitable value of $interval\_max$, a worst-case simulation with fixed n_{max} tags and increasing $split\_interval\_count$ (the number of LIs the sTags are distributed between) was constructed with assisted-type RSSI acquisition and uploading methods used for both the SPLAN and sTag. As a verification, an additional simulation was done to compare the best results obtained in Figure 5.10 to the proposed limit of $interval\_max = 250$.

4.4.4 RSSI Collection

To determine the typical averages and variability of RSSI collection duration, a simulation was written in MATLAB. The system is modelled as a matrix *AP_count* rows high, consisting of randomly generated beacon phases and channels. The matrix is sorted by phase and consecutive entries are examined for overlap. If the beacons are on the same channel, they are deemed to be non-overlapping because the CCA procedure prevents overlap. Consecutive beacons on different channels are overlapping if the difference of their start times is less than the beacon TX time T_1^0. An overlapping beacon is shifted by *beacon_period* into the future (to simulate beacon acquisition in the next *beacon_period*) and the analysis is restarted. When the entire matrix is traversed the beacon overlap is eliminated. The computed result is the time for an sTag to carry out an assisted scan, calculated as the difference between the first and last the beacons times plus the beacon duration. The simulation was repeated 32 times to obtain a statistical sample for the specific input parameters *AP_count* and *channel_count*.

Chapter 5

Performance Results

In this chapter the expressions developed in Chapter 4 are used to study the behaviour of tags and the SPLAN under varying model conditions. The objective is to demonstrate that the proposed PS schemes are effective for the SPLAN in the plain tag system and SPLAN/sTags in the smart tag system, while providing the required localization accuracy augmentation.

5.1 Plain Tag System

The primary challenge in the plain tag system is for the SPLAN to realize PS by adapting to the pTag chirps. The behaviour of the legacy pTags themselves is studied to provide a baseline comparison for subsequent evaluation of a *de novo* sTag design. The effect of pTag mobility on the pTag/SPLAN synchronization is quantified to determine the chirp interval necessary to maintain reasonable levels of SPLAN localization assistance at various levels of pTag mobility.

5.1.1 pTag Battery Life

Figure 5.1 demonstrates the effect of varying the number of packets (c) within a pTag chirp from 1 (minimum possible) to 5 (maximum recommended). Multiple packets are sent as a primitive form of forward error correction to ensure that at least one packet is received despite collisions and interference. Lightly loaded networks may benefit from lowered packet count which increases tag lifetime from $\approx$400d at a default configuration of 3 packets to $\approx$1600d at 1 packet.

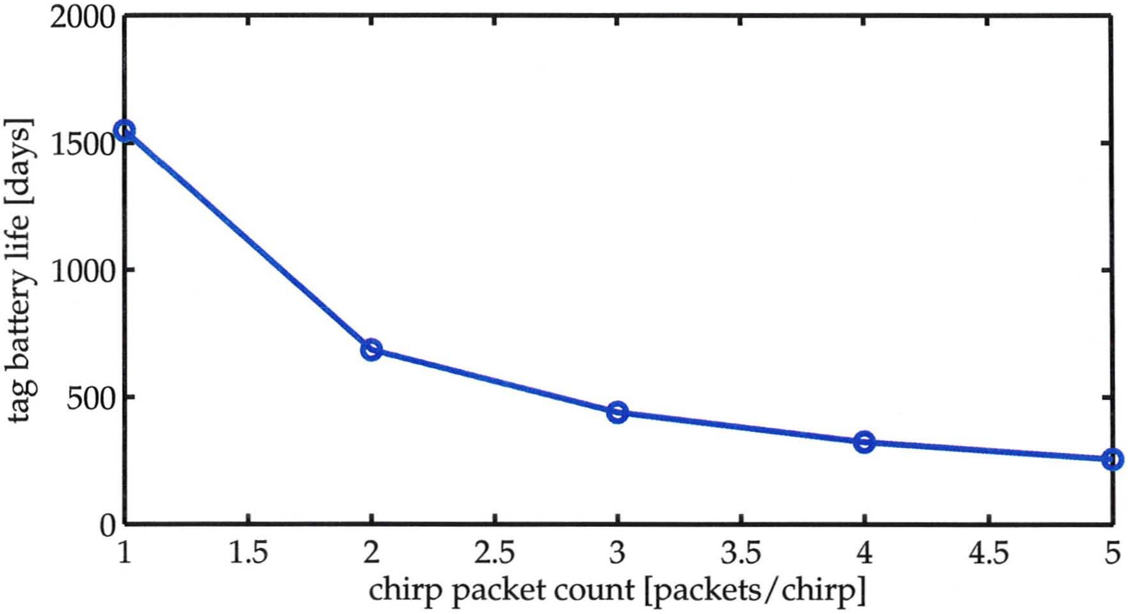


Figure 5.1: pTag energy consumption in relation to packet count per chirp

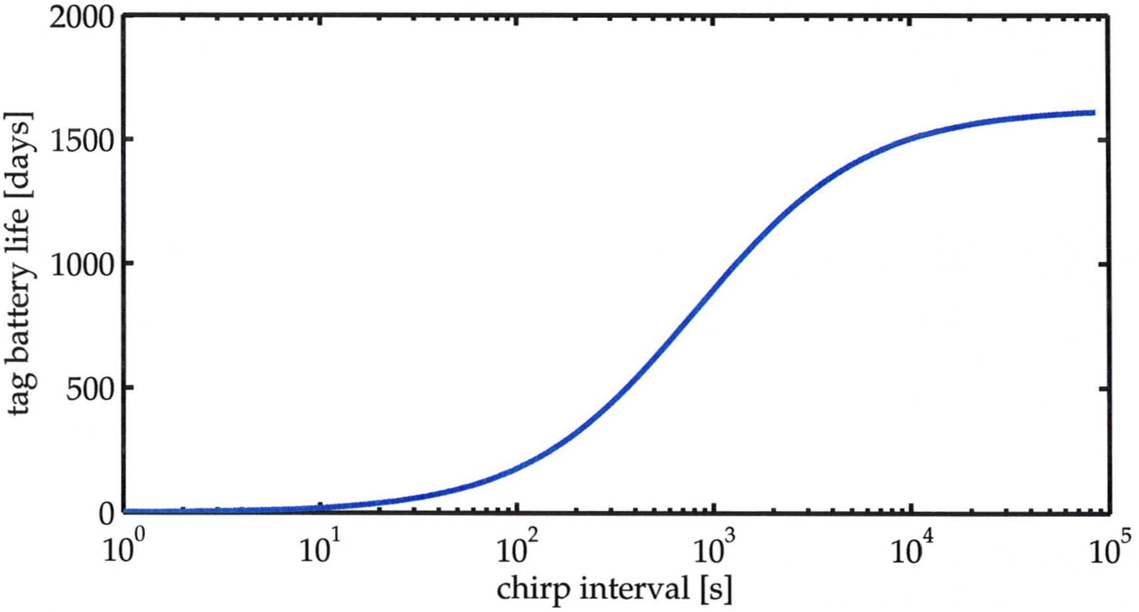


Figure 5.2: pTag energy consumption in relation to tag's chirp interval

The chirp interval (i_l) also has a profound effect on tag lifetime, as shown in Figure 5.2. The i_l parameters represents the IEEE 802.11 localization system's fundamental optimization balance between high frequency localization updates versus low battery lifetime. The Figure shows that an $i_l = 10^2$s yields the lowest reasonable tag lifetime of $\approx$200d. This is consistent with results obtained in Reference [73] for the G2C501 tag: 1 year lifetime at the 100s chirp interval. The correspondence suggests that the chosen model parameters accurately model actual systems.

Extending the chirp interval to 10^4s increases the pTag's lifetime to $\approx$1500d. Further i_l increases bring about little more performance improvement. Also, at such long lifetimes the battery's non-linear self-discharge effect ignored in the calculations becomes a significant source of error.

The effect of background WM utilization (U) is shown in Figure 5.3. An increase in U from 0% (WM dedicated to localization) to 80%, has a small effect on pTag lifetime—a decrease from $\approx$440 to $\approx$400d. While the pTag has to wait longer for WM to become clear, the corresponding radio RX mode uses much less power than the TX mode for chirp transmission (Figure 5.1). Utilization >80% decreases the tag's lifetime significantly, but such a high utilization is not realistic. Furthermore, the tag's transmissions will become frequently corrupted, necessitating the increase of c which would reduce the tag's lifetime even further.

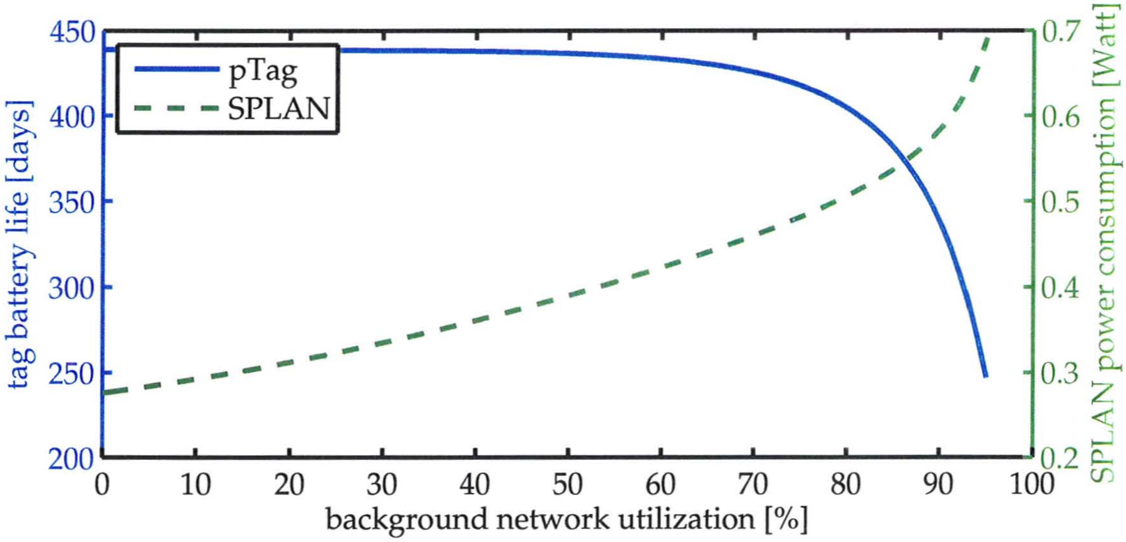


Figure 5.3: pTag energy consumption in relation to network utilization

5.1.2 SPLAN Power Consumption

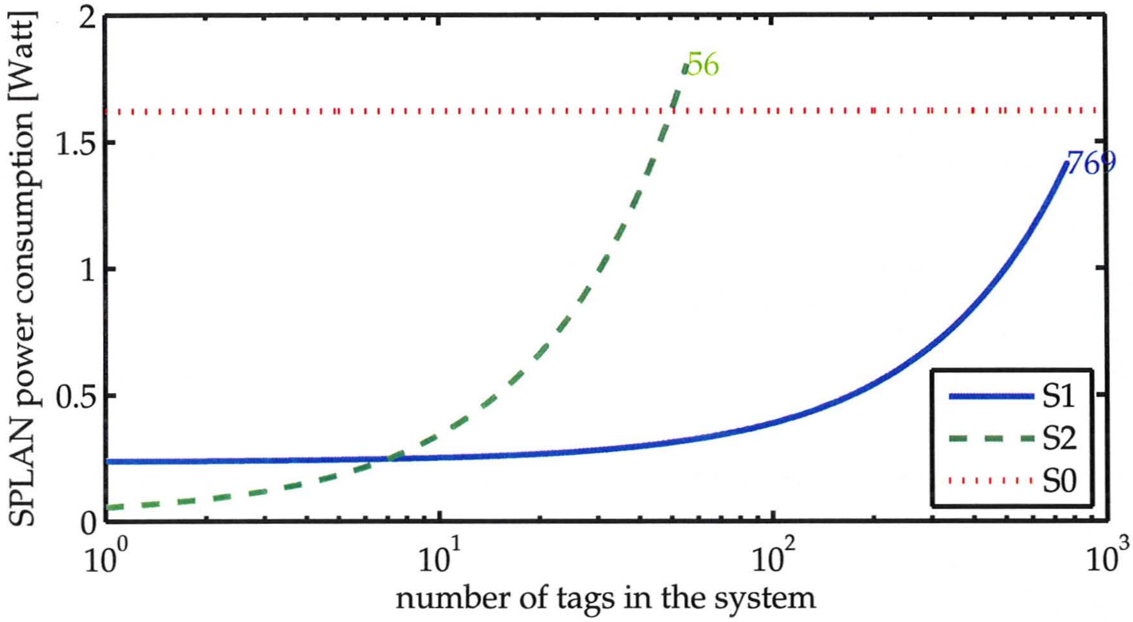


Figure 5.4: Plain SPLAN energy consumption in relation to tag density

Figure 5.4 compares SPLAN's three operational modes. The S0 mode represents the standard IEEE 802.11 node behaviour of continuously listening to pTag chirps, to provide a baseline comparison. In the S1 mode, the SPLAN uses low-latency light sleep in between expected pTag transmissions, while in the S2 mode the SPLAN uses high-latency deep sleep.

A rough estimate of the number of tags supported under the S0 mode is $\frac{(1-U)i_l}{D_{tx}^T(c)}$ which assumes an impossibly perfect scheduling of continuous transmissions without interframe gaps or frame collisions. Using standard model parameters, the value of this expression was found to be $\approx 1.08 \times 10^5$ tags. Apart from the infeasibility of this value due to perfect media access scheduling via a contention-based method, this value is also impractical. A SPLAN would have to maintain large state tables in memory and incur data processing overhead too large for an embedded system. The back-end traffic to upload that amount data to a localization server would be significant too. Even the localization server itself with its high-power hardware would not be able to handle such a load: the quoted maximum capacity for Cisco Location Appliance (Reference [63] Figure 5-24) is 2500 tags. Hence, it is unnecessary to be concerned about supporting 10^5 pTags in a PS mode; a more practical goal is that of the Cisco Location Appliance's maximum limit $n_{max} = 2500$.

The S1 mode reduces power consumption significantly to the 0.2–0.3W level (by >80% compared to baseline) while still supporting a small but practical population of 1–100 pTags. In the moderate range of 100–769 pTags, the power usage remains below the S0 mode baseline. As expected, the S2 mode reduces power consumption even further to 0.1W when only a few tags are present. However, power usage increases considerably for > 10 pTags due to high CPU utilization during memory compression/decompression that accompanies power mode switches. The sleep modes' performance cross-over occurs at 7 pTags when the S2 mode becomes less efficient than the S1 mode. For > 50 pTags the S2 mode actually takes more power than the S0 mode.

Given these findings, the SPLAN should switch between these three modes of operation depending on the number of tags present in the system, to realize maximum power saving gains possible under each offered density, while supporting the full range of 0–2500 pTags. The results are shown in Figure 5.5: if the time to the next scheduled tag chirp (i_e) is $<$ 190ms the only possible option for the SPLAN is to remain on-line in the S0 mode, as the sleep transition latencies incurred would cause it to miss the chirp. Starting at about 190ms, progressively higher PS can be realized using the S1 mode. At 42s and longer i_e intervals, the S2 mode is best.

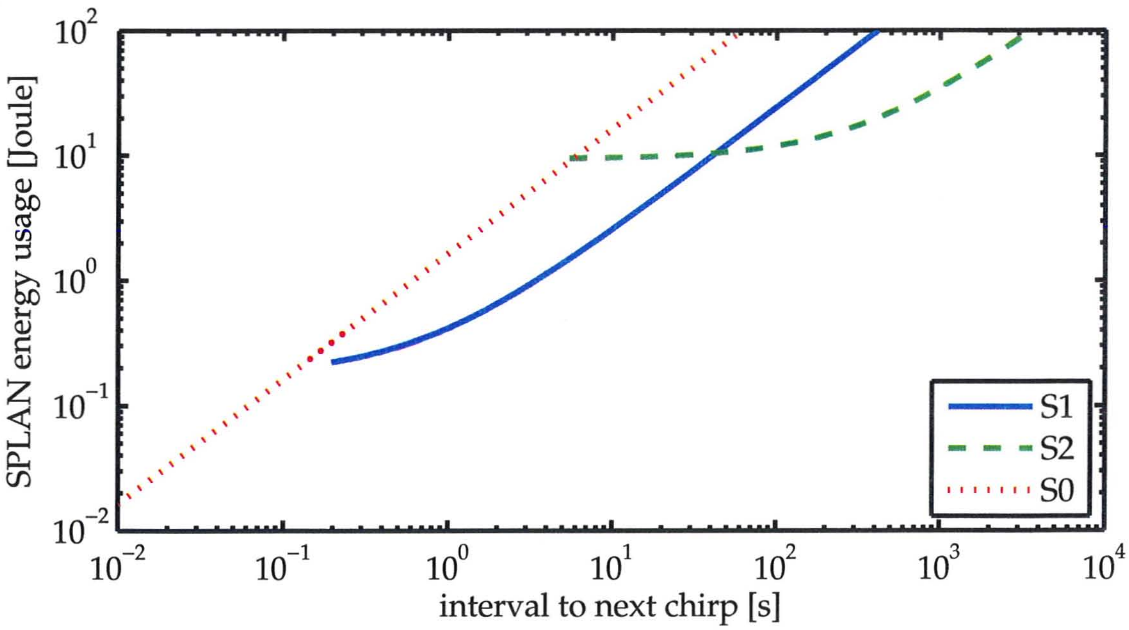


Figure 5.5: Plain SPLAN thresholds for PS method selection

SPLAN Sensitivity to Network Utilization

Figure 5.3 also shows that the SPLAN is much more sensitive to moderate increases in network utilization than the tag is. As U increases from 0% to 50%, the SPLAN's power consumption almost doubles from about 0.27W to 0.50W. For the pTag, a similar U increase only causes a decrease in lifetime from 440 to 400d. The difference can be explained by noting that the increase in U increases the probability of collisions which is irrelevant for the pTag (it will transmit the prescribed number of packets per chirp regardless of the network state). However when the SPLAN fails to receive the primary chirp packet it must use a secondary listening window which uses significantly more additional energy.

5.1.3 Localization Assistance

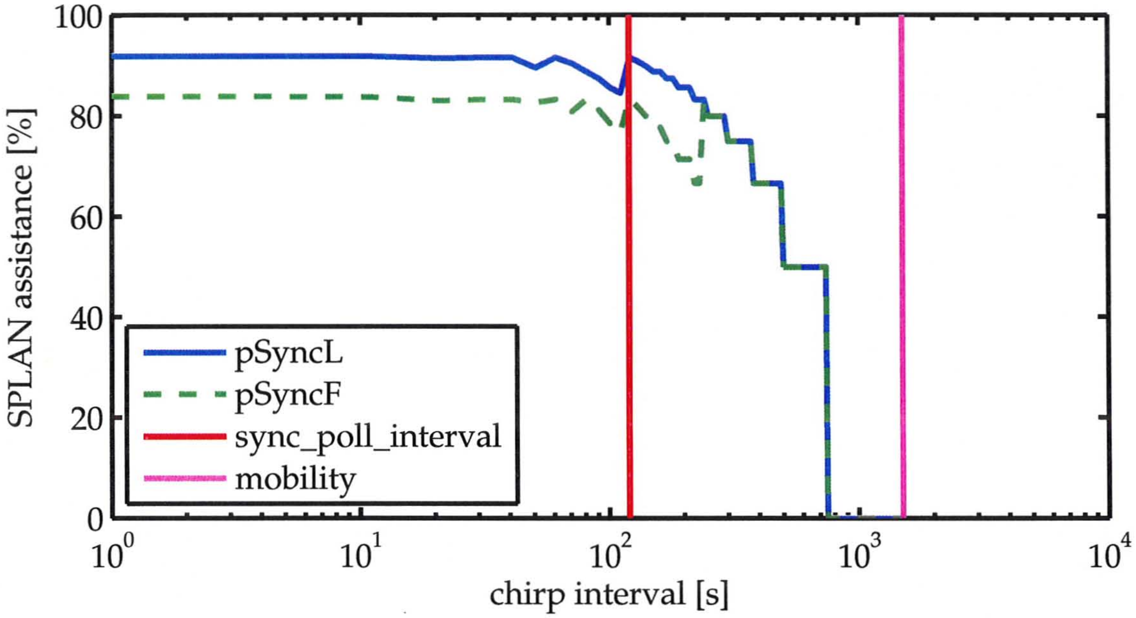


Figure 5.6: pTag localization assistance degraded by synchronization

Figure 5.6 shows the localization assistance levels for the default pTag mobility level of $m =$ 25min. Vertical lines represent the settings of the i_s and m parameters. Under the legacy pSyncL scheme the assistance level is reasonable at around 80% up to the point where $i_l > i_s$. Beyond, it starts to degrade quickly to zero at about 40% of the mobility parameter. This is expected: when i_l is at half of the m value, the pTag will emit only two chirps on average before moving to a different SPLAN. These two chirps are required for the SPLAN to synchronize, giving an effective assistance level of zero. The modified pSyncF scheme shows a similar curve shape with better performance throughout the entire range. This is expected because under the pSyncF scheme, a successful forwarding of a single pTag packet is sufficient to establish synchronization.

To demonstrate the relationship between the i_l and the m parameters on the localization assistance metric, a contour plot was prepared in Figure 5.7. It shows that even a 50% assistance level is unachievable if the pTag mobility is faster than 300s. For slower moving ($\approx$ 700s) tags a reasonable 70% assistance level is achievable provided that the chirp interval is frequent enough ($\approx$100s). When i_l is increased to 500s to conserve the pTag's energy, the 70% assistance level can only be maintained for a population of tags which move even slower (2000s average dwell time).

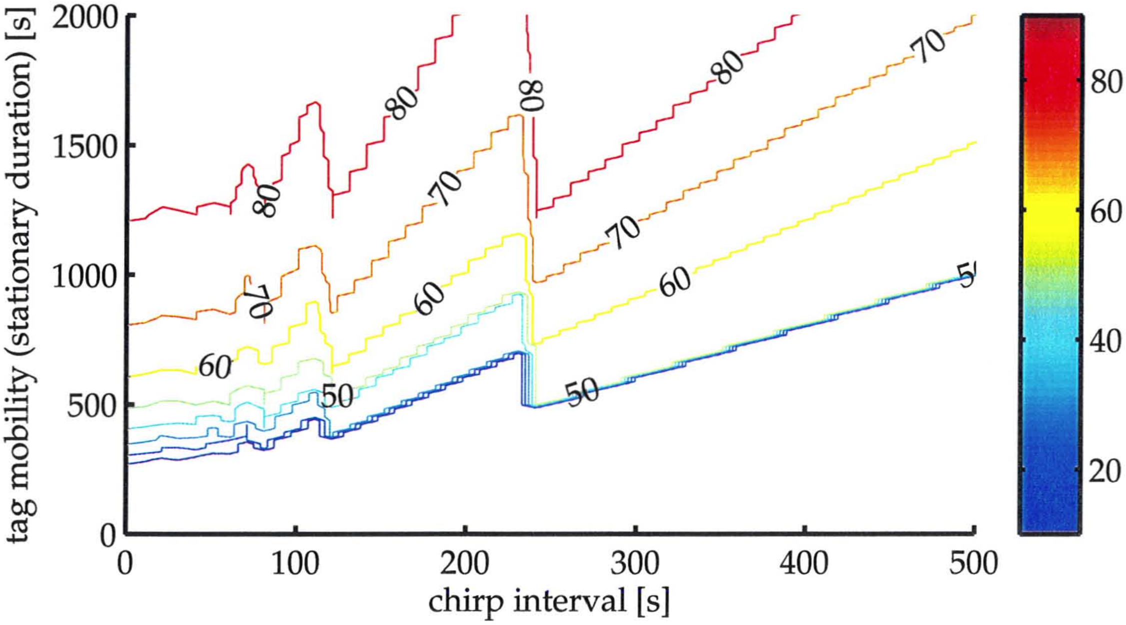


Figure 5.7: Localization assistance level contours in a plain tag network

5.2 Smart Tag System

The primary challenge in the smart tag system is for the SPLAN to achieve PS superior to that possible in the plain tag system, through the use of the proposed SPLAN/sTag localization protocol. In particular, it is expected that the proposed assisted RSSI collection and upload algorithm enhancement will reduce the required sTag active time leading to power saving for both the sTag and the SPLAN.

5.2.1 Registration

For a typical $i_r = 5\text{min}$, an initial registration attempt (R0→R1) was calculated to use a significant **0.38%** of the total sTag battery capacity. However, initial registration happens only a few times during the tag's lifetime. This is because once an sTag archives the R1 state once, the timing offsets for the registration beacon are valid anywhere on the SPLAN network and are accessible to the sTag through timing synchronization based on its always-on quartz timer chip. Considering the rarity of full registration events, there is no need to evaluate their energy requirements further.

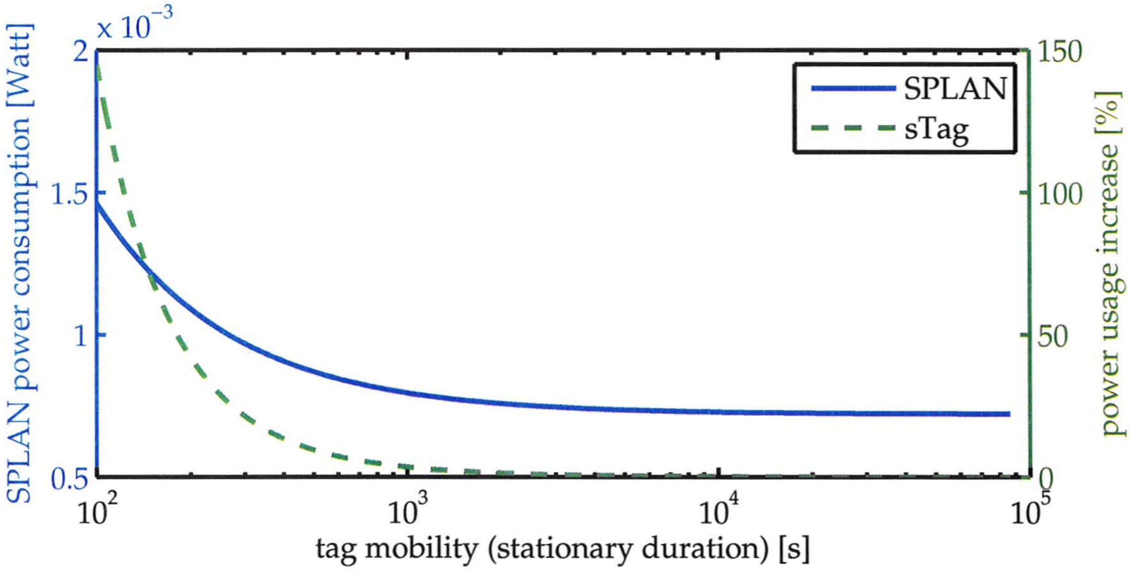


Figure 5.8: Energy consumption for registration in a smart tag system

The sTag power usage for **subsequent** registrations (i.e. R1→R2) expressed as an increase over baseline is shown in Figure 5.8. That baseline was chosen conservatively as the power usage that results in a 1000d sTag lifetime. The Figure shows that high mobility of 10^2s more than doubles the sTag's power usage. However, with a moderate mobility of 10^3s, the registration overhead decreases to 5%, and becomes negligible at longer mobilities. Overall, the registration's effect on sTag power consumption is quite reasonable under the typical mobility levels.

The corresponding registration results for the SPLAN are also shown in Figure 5.8. Even at a high sTag mobility of 10^2s, the average SPLAN power usage for registration is less than 1.5mW which is negligible in the overall SPLAN power budget. The far right of the graph with minimal tag mobility represents a baseline power usage of ≈0.7mW corresponding to just maintaining the registration beacon schedule.

5.2.2 Localization

SPLAN power

The results for SPLAN power consumption in the smart localization system using the proposed algorithm are shown in Figure 5.9. Servicing of sTags is inexpensive; the maximum supported number of sTags (n_{max} = 2500) can be serviced using just 0.273Wwhich is only 14% higher than the no-service baseline power usage of 0.240W. This demonstrates the significant SPLAN energy conservation advantage of the sTag system over the pTag system (Figure 5.4) which was able to support 769 pTags in PS mode at ≈ 1.4W power usage.

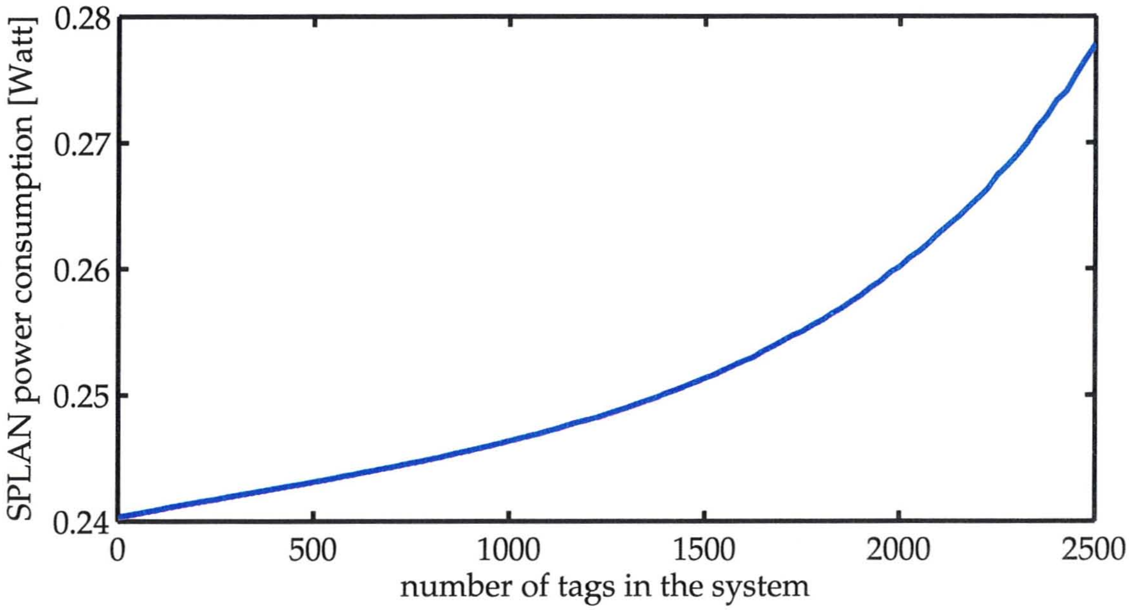


Figure 5.9: SPLAN power usage during localization in the smart tag system

sTag power

All four combinations of the two proposed communication algorithm enhancements were evaluated for the sTag, and are shown in Figure 5.10. With the standard DCF and RSSI scanning algorithms (DCFs/SCANs) the maximum possible sTag lifetime is 200d which is about half of a pTag's typical lifetime (Figure 5.3). This is reasonable, since a legacy sTag has more work (RSSI beacon scanning and data uploading) compared to pTag's work of just sending chirp packets. However, as sTag density increases, the sTag's lifetime degrades to about 80d at n_{max}, indicating that usage of DCFs has a significant effect on power consumption. Note the complete reversals of the SPLAN/tag sensitivities to the value of the n parameter in the plain/smart systems.

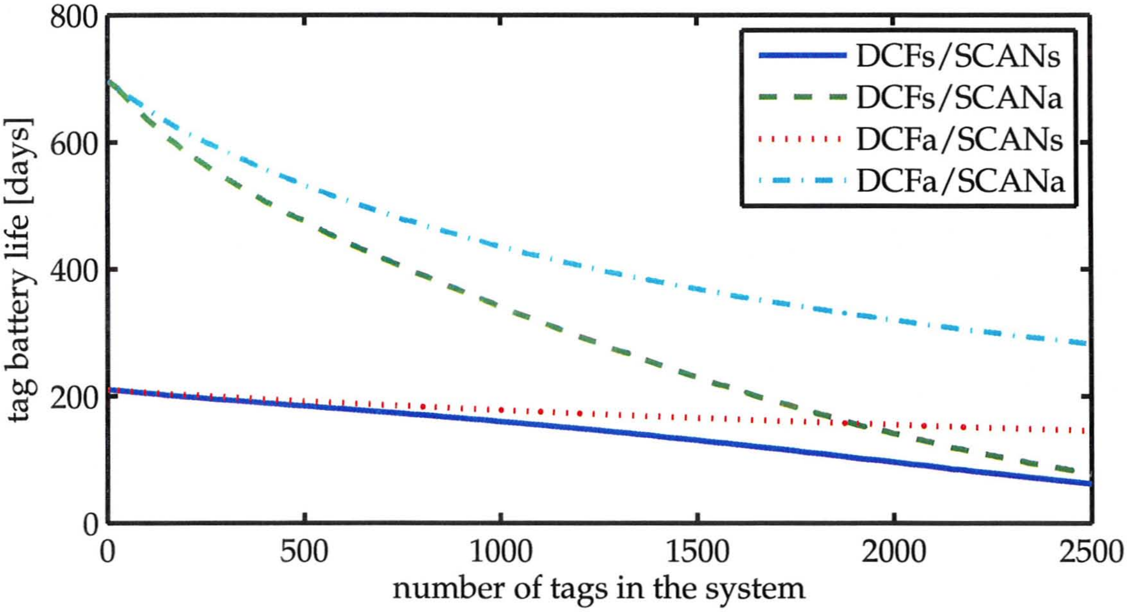


Figure 5.10: sTag power usage during localization in the smart tag system

When the assisted variation of the DCF algorithm (DCFa) is used instead, lifetime degradation of the sTag is reduced. Lifetime of 150d is achieved at maximum sTag density which is nearly double of the lifetime under the DCFs approach. However, the most immediate gains come from the SCANa method—assisted RSSI scanning; sTag's lifetime increases to more than 650d which is superior to the pTag's standard lifetime. This is an important result: despite increased complexity of the smart tag system, low sTag densities can compete in maintenance costs with pTags, provided that SPLAN assistance is used. Simultaneously, sTag specific advantages such as easy integration with an existing network, reliable communications, and enhanced accuracy due to the use of third-party AP beacons, are maintained.

The remaining issue is the performance degradation of the DCFs/SCANa algorithm with increasing number of sTags. Plain tags do not suffer from this problem since they transmit independently—their influence on each other is so slight that it was not worthwhile to model. However, the lifetime of SCANa sTag degrades to about 100d at n_{max} which is almost identical to the standard sTag and unacceptable. Hence the use of DCFa RSSI upload optimization combined with SCANa. While at n_{max} tag density level an sTag has to still wait for an average of $2500/2 = 1250$ transmission cycles, each transmission cycle is as short as possible under DCF. This results in an increase of sTag lifetime to about 300d which is only a third less than the pTag typical lifetime, and acceptable.

A detailed comparison between a pTag and an sTag is shown in Figure 5.11. It demonstrates that an assisted sTag has superior lifetime for all but the highest localization update intervals.

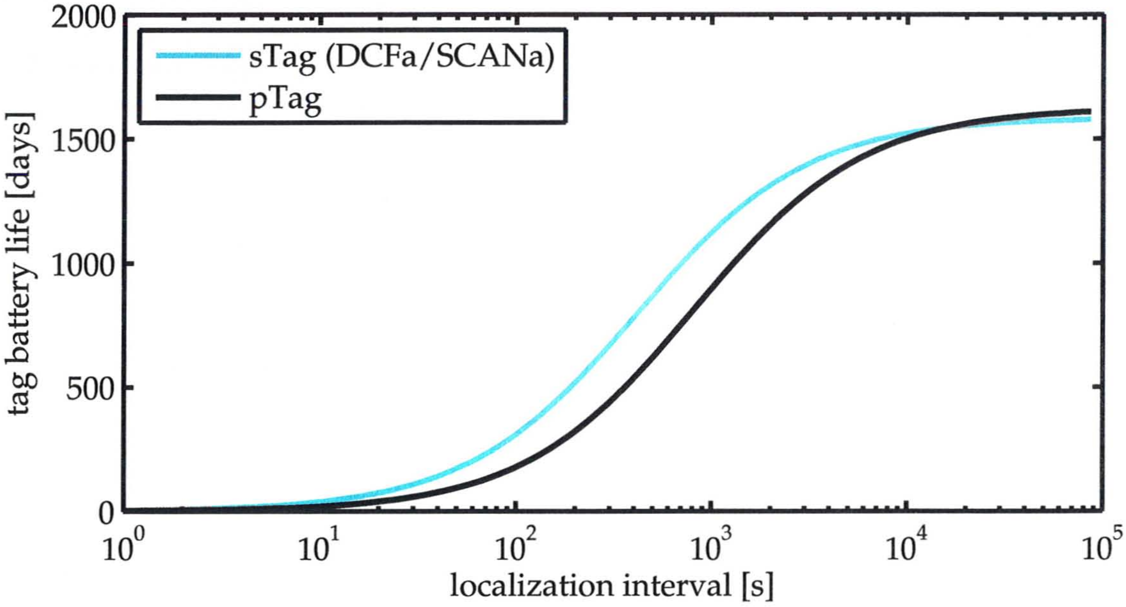


Figure 5.11: Power consumption comparison of typical-plain and assisted-smart tags

5.2.3 RSSI Upload Assistance Details

5.2.3.1 *CW* Optimization Analysis

Computation of the optimal *CW* according to Section [→4.4.3.2] is the cornerstone of the DCFa algorithm that allows sTags to upload RSSI data more efficiently. Figure 5.12 shows several instances of the optimization's objective function for various tag counts (in legend). As expected, the functions are roughly parabolic in shape. The side left of the vertex with steeper slope corresponds to a decrease of *CW* below optimum and more time wasted on collisions. The parabola's right side demonstrates a lower sensitivity to the *CW* increasing beyond an optimal level, especially for higher tag counts. For a low tag count the extra added TSs are wasted with high probability contributing to a sharply increasing DCF cycle duration.

Also shown in Figure 5.12 is a horizontal line representing ideal performance: a selected tag begins to transmit in the first slot while all the other tags select another slot. This is unachievable in DCF except for trivial situation where there is only a single tag in the system: the calculated curve for 1 sTag intersects the horizontal line, demonstrating agreement.

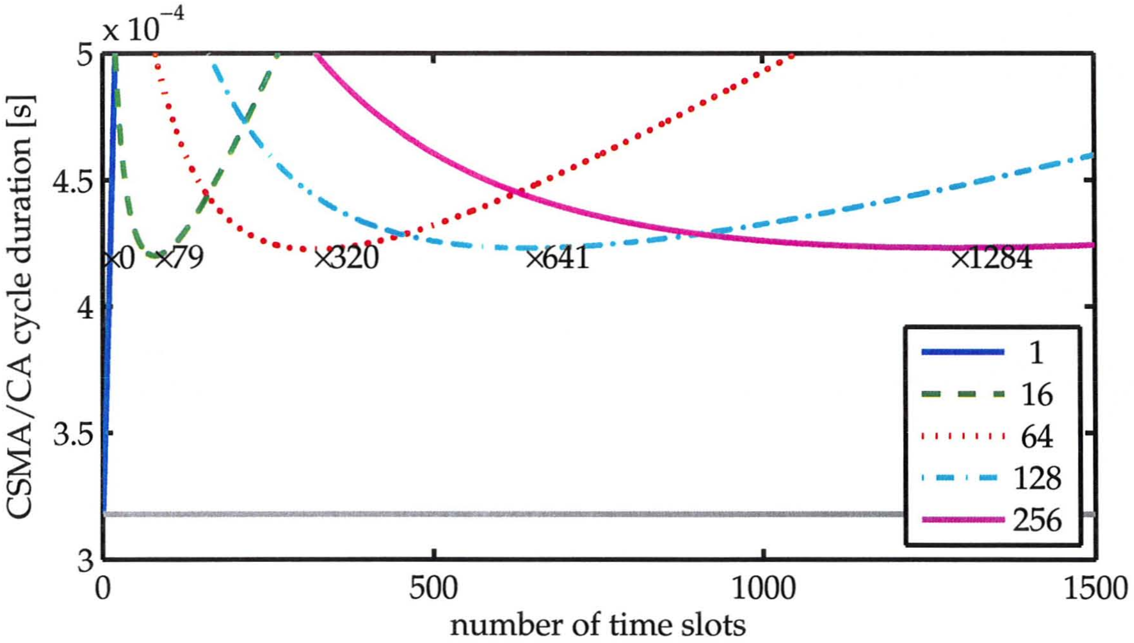


Figure 5.12: DCF optimization curves with minima for several tag counts

For $n > 1$, the parabola minima themselves lie on an approximately straight line with a small slope meaning that virtually constant RSSI upload efficiencies (4.2×10^{-4}s per RSSI packet) can be achieved even for a large number of sTags with the proposed optimization algorithm.

5.2.3.2 DCF Simulation Results

The DCF simulation was executed multiple times for various values of n using the three algorithms discussed previously: DCFs (standard IEEE 802.11 DCF), DCFe (CW_{max} extended beyond IEEE 802.11 PHY standards), and DCFa where CW computation is *assisted* through the $dcf\_opt[]$ function, as illustrated in Figure 5.12. The performance results are shown in Figure 5.13. DCFs exhibits reasonable performance up to about 1000 sTags. For $n > 1000$, an exponential increase is the duration is due to frequent collections. The simplest fix is implemented in DCFe where the CW_{max} parameter is increased to accommodate n_{max} tags. Indeed, the performance of this algorithm is much better in the 1500–2500 sTag range where linearity is maintained. However, for smaller tag counts in the 100–500 region there is a decrease in performance compared to baseline; in this range an sTag spends too much time on wasted TSs. Finally, the DCFa algorithm shows excellent performance remaining close to the ideal performance throughout the complete 0–2500 sTag density range.

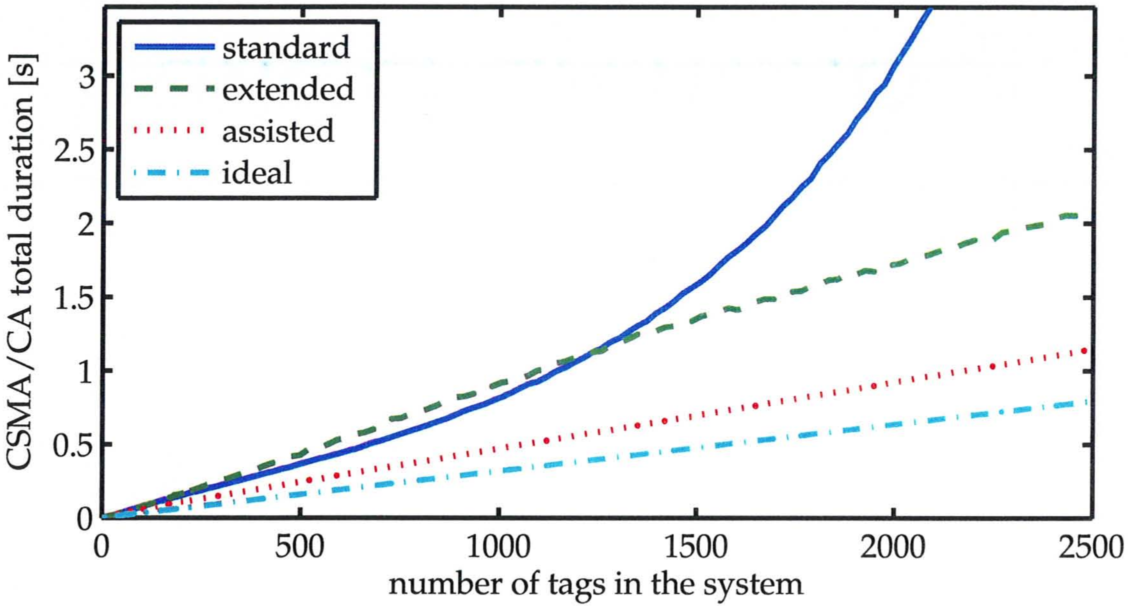


Figure 5.13: Performance comparison of DCF algorithms

5.2.3.3 Limiting Tag Count per Localization Interval

Energy consumption for high sTag densities can be reduced [→4.4.3.3] further by introducing a limit (*interval_max*) on the number of sTags that can assigned to a single LI. Figure Figure 5.14 considers a case where $n = n_{max}$, but with the tags subdivided into 1–25 LIs. As the number of LIs increases, SPLAN power usage increases linearly from about 245mW to 300mW. The increase is due to additional overheads of waking up, going to sleep, and sending a localization beacon. Since DFCa is used, the time per single RSSI upload transaction is approximately independent of number of tags per LI, so SPLAN does not experience any additional PS on this account.

On the other hand, an sTag does experience PS as a result of fewer tags present in an LI, because each individual sTag finds less sTags waiting ahead of it in the upload queue. Figure 5.14 shows that the most gains are experienced with the creation of about 10 intervals; beyond that the sTag's lifetime reaches a plateau. Therefore, the setting $split\_interval\_count = 10$ is a good compromise: it increases the sTag lifetime about two-fold, while increasing SPLAN power consumption by only about $1 - \frac{0.265}{0.250} \approx 6\%$. The corresponding $interval\_max = 250$ is calculated as:

$$interval\_max = \frac{n_{max}}{split\_interval\_count}$$

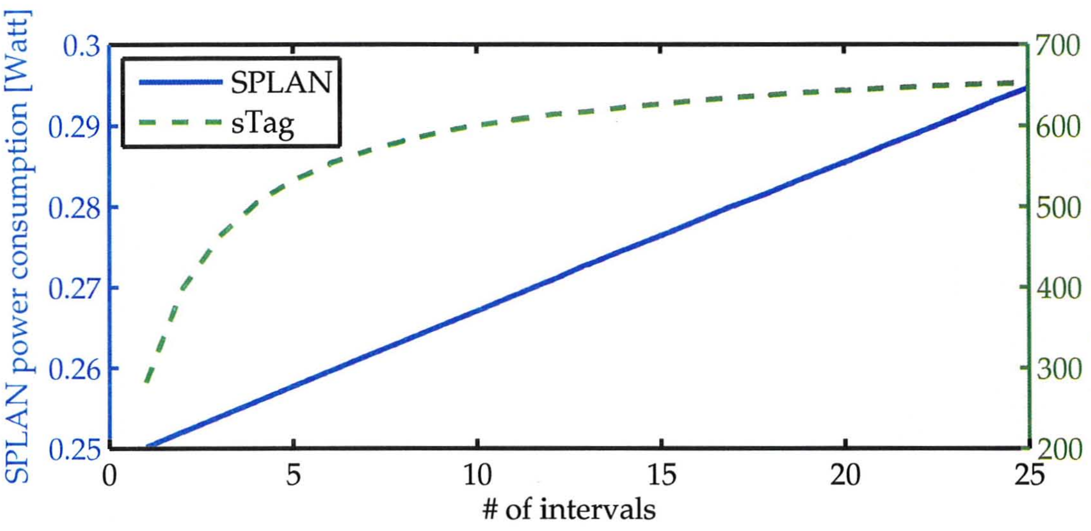


Figure 5.14: SPLAN and sTag energy consumption trade off in split LIs

The overall system performance of the DCFal method (DCF with *assistance* and LI *limiting*) is shown in Figure 5.15. An sTag utilizing DCFal experiences only minimal lifetime degradation over the entire range of supported sTag counts, up to n_{max}. DCFal allows the tag's lifetime to become nearly independent of the tag density, just as in the plain tag system.

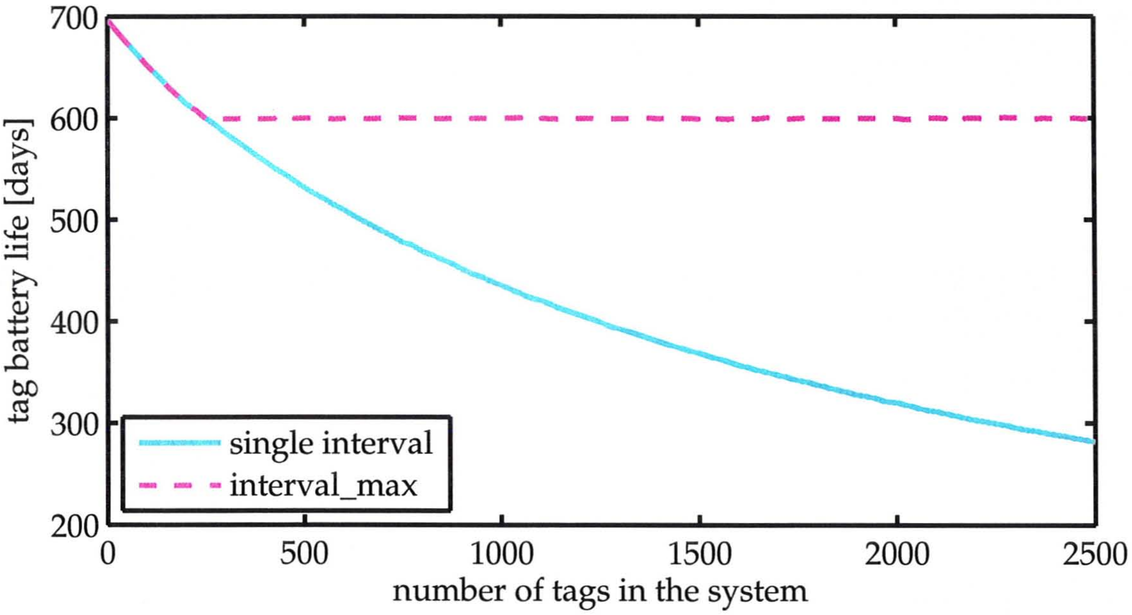


Figure 5.15: sTag energy usage with LI splitting enabled

5.2.4 RSSI Collection Assistance Details

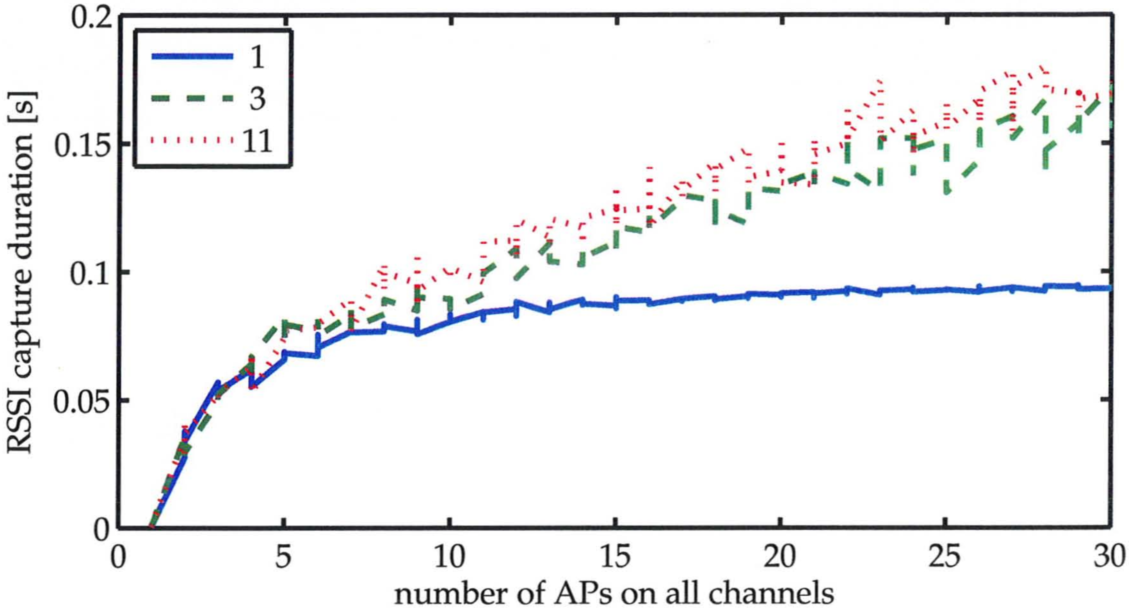


Figure 5.16: Performance of assisted collection of beacon RSSI

Figure 5.16 shows the simulation results of assisted RSSI collection duration for different number of WM channels in use. For a single channel and a single AP, the lone beacon can be captured almost instantly (intersection with the x-axis). Increasing the number of APs increases the capture duration to about 80ms at 10 APs. As the number of APs increases further, the entire *beacon_period* timeline is filled, giving utilization of about 95ms at 30 AP, close to 100ms that would be necessary with a conventional scan.

The strength of assistance is revealed when 3 channels are considered: the capture performance remains similar up to about 10 APs due to low probability of temporal beacon overlap on separate channels. As more APs are added, the overlaps become more frequent and the capture time increases to about 140ms at 20 APs. This result is still reasonable and compares favourably with the unassisted scan time of 3channels $\times$ (100ms/channel) $=$ 300ms. Also, 20 APs is a high AP density considering RSSI localization accuracy peak of $\approx$10 APs, hence the 1–10 AP range is the most important. The most gains are realized with 11 channels (maximum supported in the 2.4 GHz band) where the capture time is not much longer compared to that of 3 channels.

In summary, assisted RSSI collection saves a significant amount of time and hence reduced sTag's energy usage. Additionally, the assumption made previously that assisted RSSI scanning can be usually completed in two *beacon_period* (about 200ms) is justified.

Chapter 6

Summary and Conclusions

This thesis considers client and infrastructure power saving in IEEE 802.11 localization networks. Due to inaccurate models of RF signal propagation in *environment simulation* algorithms or inaccurate RSSI vector maps in *pattern matching* algorithms, localization accuracy can fall below required levels. To increase accuracy in existing systems or to inexpensively provide localization coverage in new systems, the use of a SPLAN is proposed. This device features a subset of IEEE 802.11 functionality adapted specifically for use in localization; it is powered by renewable energy allowing for freedom and optimality of placement.

To provide input data for SPLAN resource allocation (sizing of its photovoltaic system components), a detailed hardware power consumption model was developed that considers both the SBC and the IEEE 802.11 NIC components of the SPLAN. State switching transition latencies were also considered to allow for a useful trade-off comparison between light and deep sleep states. It was found that the deep sleep mode was the most optimal for pTag densities in the low range of 0–7 pTags/SPLAN while power saving could still be realized in the range of 7–769 pTags/SPLAN in the light sleep mode.

An additional challenge found in the plain tag system was the intrinsic tag mobility that causes loss of synchronization between pTags and the SPLAN, reducing the possible frequency of localization assistance. The pSyncF synchronization algorithm requiring slight pTag and AP firmware changes was proposed and evaluated together with the completely legacy-compatible pSyncL algorithm. It was found that a reasonable level of accuracy improving assistance could be maintained with both algorithms with a proper choice of the chirp interval parameter.

These results demonstrate a successful design of SPLAN power saving for seamless integration with legacy pTag systems. For new systems, a *de novo* design of a smart tag system was developed; it described sTag/SPLAN behaviours and the related communication algorithms including their data packet structures. The smart system enabled collection of RSSI data from third-party APs, increasing localization accuracy at no extra cost. However, it presented a challenge of maintaining or exceeding the PS levels realized for the pTag system for both the SPLAN and the sTag, despite the necessary increases in the communication algorithm complexity.

Significant power saving was realized by the introduction of the SPLAN-assisted SCANa algorithm for AP beacon collection which consumes a large part of sTag's active time. It was found that with assistance, beacons of 30 APs located on 11 different channels could be collected in under 200ms; this usually takes 1.1s via a standard IEEE 802.11 passive or active scan.

Another power saving measure introduced was an assisted DCF algorithm (DCFa) for RSSI data uploading. Through the use of convex optimization and advisory data provided by the SPLAN's localization beacon packet, an sTag was able to dynamically adjust its DCF *CW* variable. It realized an optimal media access time of 4.2×10^{-4}s, nearly independent of offered sTag densities and close to ideal performance.

Through a simple QoS protocol and a connection admission scheme, the SPLAN was allowed to control the number of sTags assigned to each LI. At a slight cost of 6% power usage increase on the SPLAN, DCF contention queues were decreased resulting in even further sTag power saving. The sTag battery lifetime with all assistance algorithms applied was found to be $\sim$650d, nearly independent of sTag densities. This compares favourably with an unassisted sTag lifetime minimum of $\sim$100d and a typical pTag lifetime of $\sim$450d.

The corresponding goal of SPLAN power saving in the smart tag system was likewise realized. Power usage of $<$ 265mW for all allowed tag densities ($n_{max} = 2500$) was demonstrated as equivalent to supporting only $\sim$50 pTags in the plain tag system. A SPLAN without power saving was found to consume $\sim$1600mW, so the corresponding reductions in the necessary photovoltaic system size are significant.

Chapter 7

Proposed Future Work

This chapter outlines several research directions that complement the work presented in this thesis, to address some of the issues encountered or to further optimize the system's functions. Suggestions are provided for system designs, control algorithms, and performance evaluation experiments.

7.1 SPLAN Environment Monitoring

In the proposed scheme, the SPLAN of the smart tag system periodically scans for beacons from neighbouring APs; beacon **timing** information is disseminated to sTags to provide scanning assistance [→3.2.5]. The **RSSI measurements** of these beacons could also be recorded and automatically analyzed to detect relocated APs or changes in environmental conditions such as the relative humidity levels, the movement of people and the opening/closing of doors. Compensation for these effects through comparison with a baseline can improve localization accuracy, as demonstrated in Reference [82] where radio-frequency identification tags and environment sensors were used to provide environmentally-adapted radio maps that increased IEEE 802.11 localization accuracy by an average of 2.6m.

7.2 Motion Sensor Support

A motion sensor (a multi-axis accelerometer chip) is a standard feature of IEEE 802.11 tags, plain and smart. It remains active during the tag's sleep cycle, so that the tag can be woken up by motion detection. The tag activated this way uses an alternate, smaller i_l (i.e. i_{lm}) for its chirp, resulting in the loss of synchronization with SPLANs. This problem is particularly troublesome for pTags which experience frequent movement/stationary transitions.

Two solutions to the problem are presently proposed that preserve compatibility with existing systems but permit more effective SPLAN assistance. The first solution is the addition of a separate motion timer to tag firmware; a modified moving sTag continues to send i_l spaced chirps which contain the phase and i_{lm} of the additional motion chirp. If the tag's motion continues, the SPLAN adds a LI to capture the additional motion chirps. If the SPLAN misses the motion window, it simply continues to receive the original i_l spaced chirps. The second, even simpler solution without firmware changes is the restriction of configured motion chirp interval values to i_l/m_d, where $m_d \in \mathbb{Z}$. A SPLAN receiving non-contiguous chirps adds an i_l/m_d LI in response.

7.3 Explicitly Scheduled RSSI Upload

In the proposed RSSI upload scheme an assisted modification of DCF was used [→3.2.6]. Instead, upload windows could be explicitly scheduled for each individual sTag by the SPLAN as part of the sTag registration process. This approach was not evaluated due to its significant complexity and increased signalling overhead, including the propagation of scheduling data across the entire SPLAN network to support tag mobility. As an alternative to global propagation, a hybrid system could be used where sTags are assigned explicit TX windows only on the SPLANs that they are registered with; any new tags that come into the SPLAN's range would be allowed to TX in an additional standard DCF window until they are registered.

7.4 Network-wide Synchronization Data Dissemination

Due to the inherent tag mobility, a pTag occasionally leaves the service area of its SPLAN and enters the service area of another. That pTag requires a new synchronization process, losing opportunities for localization assistance. To avoid new synchronization and provide better support for pTag mobility, SPLANs could share among each other the tag data including the tag's MAC address, chirp phase, chirp interval, and chirp interval's standard deviation. Based on these data, LIs global to the SPLAN network could be scheduled.

7.5 Listening Interval Suppression

Sometimes a plain tag may enter an area with an inherently low localization error. If the pTag is expected to remain in that area, some SPLAN LIs could be suppressed as the associated measurements would not provide a localization accuracy enhancement, while a suppressed LI would reduce SPLAN's energy usage. Alternatively, a SPLAN energy control mechanism could be proposed that would take into account both the expected localization accuracy and SPLAN's current battery level to decide which LIs are actually needed.

To support LI suppression, a method for predicting tag mobility and a method for quick estimation of localization error for a given location based on a selected subset of available APs and SPLANs would have to be developed. Perhaps a heuristic such as "an activated SPLAN must be within a certain distance of the estimated position of the tag and not be co-linear within a certain degree of an AP" would be appropriate.

7.6 Enhanced Localization QoS Options

In the proposed design for the smart tag system, the QoS specification consists of only i_l—the required exact interval between RSSI scans. Neither longer updates (degraded real-time response of the system), nor shorter updates (increased tag battery drain) are deemed acceptable to the sTag. However, there is little value to configuring tags with an exact i_l; a configuration of an i_l range, i.e. $[i_{lL}, i_{lH}]$ may be sufficient. The SPLAN would benefit from such a specification because it could more effectively perform LI combining. To do that, it would check whether the $[i_{lL}, i_{lH}]$ range contains any i_l integral multiples that belong to previously registered sTags. If so, it would assign the new tag to an existing LI.

Another QoS possibility is discarding the use of the i_l specification completely and instead using the more fundamental *required accuracy* parameter. Note that the actual role of the i_l parameter is to indirectly control localization accuracy in response to tag mobility. By focusing on the *required accuracy* specification instead, a SPLAN can dynamically alter tags' i_l values depending on the observed tag mobility, conserving LIs and power. Moreover, a group of SPLANs could use the *required accuracy* parameter as one of the parameters for global power optimization to decide which SPLANs are necessary to provide *required accuracy* for a given sTag.

7.7 sTag RSSI Collection Analysis

Several simplifying assumptions were made to model the sTag RSSI beacon collection procedure [→3.2.5]. Other factors that may need to be considered when constructing a scanning schedule are the hardware-dependent channel switching duration and the sleep cycle entry/exit durations. Also, AP management beacon timing precision influences the duration of the guard buffer used to ensure a successful reception of the beacons.

Some heuristics for construction of the scanning schedule can be proposed for experimental evaluation. In particular, a "next on the time line" heuristic is easy to implement; if the next upcoming beacon is far in the future (as determined by consideration of transition latencies) then the tag goes to sleep. Otherwise, it tunes to the correct channel and waits for beacon reception. This algorithm is expected to function near optimality if the beacon density is low and the tuning parameters (in particular the sleep threshold) are correctly chosen. However, it is not expected to deal well with overlapping beacons. For the *control* experiment, a standard sequential scan where the tag spends 100+ms on each channel could be used, while a *perfectly optimal* reference would require iterating through all the possible beacon acquisition combinations.

Model trace data for these experiments may need to be collected from a live system. The data could include the typical distribution of beacons across radio channels, the declared beacon interval lengths and the actual beacon interval length variability due to CCA.

Appendix A

A.1 Localization Applications

Representative applications of localization that are suitable for the localization system proposed in this thesis are presented below. Most of the provided references discuss IEEE 802.11 localization systems specifically designed for these applications.

A.1.1 Agriculture[2]

Grazing farm animals equipped with localization tags can be monitored automatically reducing human involvement and hence cost. The obtained data are used to optimize pasture utilization and animal nutrition.

A.1.2 Robotics[3]

A mobile robot operating in a non-constrained environment requires localization for autonomous navigation. This is a difficult problem in the area of robotics, usually solved by analysis of data from the robot's various on-board sensors such as vision, infrared, ultrasonic, and contact sensors. Since robots are often equipped with IEEE 802.11 transceivers for obtaining instructions and providing feedback, the dual use of IEEE 802.11 for localization reduces the robot's cost.

A.1.3 Health Care[4, 5]

Expensive medical equipment is often over-provisioned to account for misplaced items. Localization systems enable that equipment to be found easily, reducing the need for spares. Patient and staff scheduling for operations can be optimized if their locations are available in real-time.

A.1.4 Network Security[6, 7]

A poorly secured wireless network can be used by intruder to access a company's secret data or its Internet bandwidth. With localization, the originating node's location can be used as an additional security check when granting access to a wireless network, e.g. access could be granted only inside a building and not on its adjoining parking lot.

A.1.5 Physical Security[8]

Tagged visitors can be monitored for compliance with access to designated areas. Equipment such as laptops and PDAs with a build-in IEEE 802.11 interface can be monitored to ensure that it is not stolen from the premises.

A.1.6 Emergency Response[8]

Training scenarios in a controlled environment can be evaluated and reviewed by tracking personnel movement during exercises. Actual emergency rescue operations could benefit from knowledge of the exact location of all personnel to coordinate rescue efforts from a central location. This long term goal could be realized when localization technologies such as the one proposed in this thesis become standardized and pervasive.

A.1.7 Public Transit[9]

Printed bus schedules become inaccurate due to unexpected street congestion increases. Metro-scale real-time localization systems can present bus locations and accurate departure times, leading to greater satisfaction and convenience in the use of public transit.

Some unifying characteristics of these examples are that the localization area is controlled by a *single entity* and is relatively *limited in size.* Networks owned by multiple entities are unsuitable for IEEE 802.11 localization because there are currently no provisions for sharing access, such that localization tags owned by one organization can work on a foreign network. Networks that are geographically expansive are better served by localization systems based on radio technologies such WiMAX designed for multi-kilometre range, compared to the typical $\sim$100m range of IEEE 802.11 networks.

A.2 Localization Tag Details

A.2.1 AeroScout Tag

Table A.1: AeroScout pTag OSI L2 frame header format

item	value description
ToDS	true
FromDS	true
TX address	MAC address of the tag
RX address	generic CCX multicast address (01:14:7E:00:00:00)
destination	not used (zero)
source	not used (zero)
fragment number	incremented by 1 for each packet within a chirp
sequence number	incremented by 1 for each new chirp

Table A.2: AeroScout pTag OSI L2 chirp data

size	label
5 byte	CCX header
4 byte	tag product type ID
9 byte	battery telemetry
variable	asset telemetry such as temperature, pressure, quantity (of the item being monitored), motion (speed), humidity, fuel volume

Table A.3: AeroScout pTag configuration options and [defaults]

option	description
channel selection	bitmap of channels used to send chirps [1,6,11]
motion/static TX interval	configurable interval (s) between successive activity cycles, with separate values for stationary and motion use
repetition count	number of packets to send per chirp [3]
repetition interval	interval (ms) between packets of each chirp [512ms]
TX power	transmission power for chirps [18dBmW]
data rate	data rate for chirps (1, 2 Mbit/s supported) [1Mbit/s]
data frame format	[CCX-compatible] or proprietary
CCA energy threshold	maximum permissible signal value for declaring a clear channel

A.2.2 PanGo Tag — Plain Mode

Configuration options of a PanGo tag shown in Table A.4 are similar to those of an AeroScout pTag.

Table A.4: PanGo pTag configuration options

option	description
channels selection	bitmap of channels used to send chirps
motion detection	motion sensor disable/enable
repetition count	number of packets to send per chirp
motion/static TX interval	1min–12h

The PanGo chirp frame shown in Table A.5 is also similar to the AeroScout frame. The destination and source fields encode battery status information, while an AeroScout data frame contains it inside its data portion. The PanGo encoding method reduces the frame size slightly at the expense of the distortion of the IEEE 802.11 standard (the MAC address field's meaning is changed) and lower flexibility (the battery's state is represented by two bits instead of nine bytes).

Table A.5: PanGo tag plain mode frame format

item	value description
ToDS	true
FromDS	true
transmitter address	MAC address of the tag
receiver address	01:14:7E:00:00:00
destination	control and miscellaneous data
source	more additional data

A.2.3 PanGo Tag — Smart Mode

Plain tags traditionally support IEEE 802.11b data rates of 1 and 2Mbit/s; the resulting long transmission range allows multiple APs to receive these chirps. Such OSI L2 multicast packets do not initiate the IEEE 802.11g protection mode, so they do not waste network capacity beyond the time used for their own transmission. On the other hand, a smart tag does associate with AP infrastructure and causes initiation of protection mode if it does not advertise the IEEE 802.11g capability to the AP. Since the protection mode can significantly degrade the overall network capacity for IEEE 802.11g devices, it is important for the smart tags to fully support the IEEE 802.11g modulation. The PanGo tag operating in smart mode supports *TX rate* selection of 1, 2, 5.5, 6, 9, 11, 12, 18 and 24Mbit/s, so it can inter-operate correctly with IEEE 802.11g networks.

A.2.4 Tag4M Tag

Tags for Measurement company produces IEEE 802.11b multi-purpose sensor tags[74] that include temperature and motion sensors. Their user-programmable firmware with configuration options shown in Table A.6 allows them to be used in a variety of applications including environmental monitoring and localization. Tag4M supports high and low power modes; in the high power mode its 32-bit CPU is driven by a 44.0 MHz clock, while in the low power mode a 32.768 kHz clock is used for timing and operation of sensor circuits. Both timer expiry and detection of motion can trigger a transition from a low power to a high power mode.

Table A.6: Tag4M configuration options

option	description and [defaults]
src IP address	tag's IP address
dst IP address	UDP data packet destination IP address
dst port number	UDP data packet destination port
channel map	list of channels used for association and RSSI scanning
SSID	SSID of the AP to associate with
AP channel	channel number of AP to associate with
sleep time	time between activity cycles [5000ms]
RX time	time to wait for a command packet after TX of packet [80ms]
motion monitor	enable/disable

After the Tag4M tag is powered on, it associates with an AP selected by a specified SSID/ channel combination and obtains an IP address via DHCP if it has no statically configured IP address. Following this initial start-up, the next time Tag4M wakes up from the sleep state it performs sensor measurements immediately and sends the measurement-containing packet to the AP it is associated with. It then waits for a command packet and if none is forthcoming, it enters the sleep mode. The command packet can be one of two types: configuration update request or AP RSSI scan initiation request. To perform the RSSI scan (useful for localization) the tag sends a *Probe Request* packet on each configured channel and listens for *Probe Responses*. The returned data packet contains a repeating 40-byte data structure that characterizes each AP found: 4 byte RSSI measurement, 4 byte channel number and 32 byte SSID string.

The Tag4M data packet begins with magic number *0x1* and includes a varying number of 32-bit integer pairs. The first integer is the item ID and the second is the item's value. The available items are listed in Table A.7.

Table A.7: Tag4M packet data

id	description
0x0	IP address of the tag
0x1	sleep duration
0x2	RX duration
0x3	cumulative time of low-power mode
0x4	cumulative time of high-power mode
0x5	battery voltage (current)
0x6	battery voltage (minimum)
0x10	temperature (current)
0x11	temperature (minimum)
0x12	temperature (maximum)
0x43	channel bitmap (for AP scanning)
0x8	configured motion sensing state

Hypothetical Use of Tag4M in Localization

The Tag4M company does not manufacture their tags specifically for localization and does not provide a turnkey companion localization system. It is expected that the end-user integrates the tag into their own localization system.

Based on the capabilities of the tag, a hypothetical Tag4M-based localization system can operate in either plain or smart mode. For plain mode operation, the tag can be configured for either UDP packet broadcast or unicast. However, even the broadcast configuration will not allow a standard AP to receive the packet as part of its normal operation, due to the lack of association. Therefore, infrastructure APs must be adapted to specifically look for a localization packet with the specific Tag4M parameters: a combination of the registered tag's MAC address, a UDP port number, and a unicast destination IP address. Following packet capture, an AP measures the packet's RSSI and sends the information to a localization server, as in the usual pTag system.

The smart mode operation would have to be different from a typical Ekahau/PanGo system in that the Tag4M is not programmed to measure RSSI by default; this functionality needs to be explicitly requested every time at the expense of additional complexity and network traffic.

This discussion demonstrates that even a highly configurable Tag4M tag faces significant problems for adaptation in a localization system. Therefore, a *de novo* joint sTag/SPLAN design that was proposed is superior.

A.3 Additional IEEE 802.11 NIC Power Consumption Data

Power Consumption of IEEE 802.11 network cards is discussed in Reference [83] where Atheros—a manufacturer of wireless chipsets—defines five generic power states of a wireless card (Table A.8).

Table A.8: IEEE 802.11 NIC power states—Atheros generic

state	description	power (mW)
Off	device is completely powered off	0
Sleep	majority of the circuitry is turned off	40
Listen	radio is listening for traffic but is not passing data to host	800
Receive	NIC is detecting, demodulating and passing packets to host	900
Transmit	NIC is modulating and sending packets	2000

Although specific power consumption values were given, the model number of a card with such specifications was not mentioned. To confirm these values, a datasheet[84] for a specific card (Wistron NeWeb CM9) card utilizing an Atheros AR5004 chipset was consulted. Its power consumption is shown in Table A.9 with the mW value calculated based on the 3.3V "Operation Voltage" specification.

Table A.9: IEEE 802.11 NIC power states—Atheros AR5004

state description	power consumption	
	mA (specified)	mW (calculated)
FTP Tx	410	1350
FTP Rx	310	1020
Standby mode	270	890
PS mode	50	160
RF Kill	40	130

These specific values are approximately consistent with the generic ones. The TX power is lower at 1350mW versus 2000mW and sleep power is higher at 130mW vs 40mW. The generic *Receive* state does not exactly correspond to the *FTP Rx* state (which includes an occasional transmissions of OSI L2 and TCP ACK packets), but the quoted power consumption is similar. Overall, the generic values corresponding to the ideal power states appear to be reasonable.

Power consumption values used in this thesis were instead obtained from *Cisco Systems PCM 350* (Table 4.5) because they were presented together with the needed latency data. They are likewise similar to values shown in this section.

A.4 Sleep Timer Accuracy and Precision Considerations

All the reviewed plain tag systems [→2.2.3.1] schedule their chirp transmission using a configurable timer. A poor accuracy timer requires the SPLAN to start listening far in advance of the anticipated TX time to account for clock variations. This extra listening window extends the usage of high-power mode duration and reduces the PS benefits.

Table A.10: Tag hardware configurable timer resolution

manufacturer	chirp interval
AeroScout	1 ms
PanGo	1 s
Tag4M	1 ms

Configurable timer accuracy obtained from pTag hardware documentation is shown in Table A.10 which suggests that tags use a high resolution timer during their sleep mode. In fact, Tag4M sTag specifications include a "32.768 kHz clock used for timing of sensor circuity in low power mode." The typical worst case accuracy of a 32.768 kHz tuning-fork crystal oscillator[85] is $\pm 20ppm$ at +25°C and up to $\pm 170ppm$ throughout the $-40°C$ to $+85°C$ industrial temperature range. For a typical 5min chirp rate, the worst case accuracy translates into a considerable 51ms error. Still, these values are misleading because they are based on differences from the true time, while the calibrated difference between two crystal timer sources experiencing the same environmental conditions (temperature) is more relevant.

It is actually the precision of the time measurements rather than the absolute accuracy which is relevant. Thus, a better approximation of timing precision which can be reasonably obtained from a standard quartz timer source is the accuracy of a temperature-compensated crystal unit such as the Maxim DS3231S[76] which specifies a ±3.5ppm accuracy over the industrial temperature range of -40°C to +85°C. The resulting error of ≈1ms per 5min of operation is expected to provide good performance.

Figure A.1 demonstrates that a clock accuracy decrease from 3.5ppm to 150ppm causes slight increase in SPLAN power consumption from about 0.335W to 0.385Wfor $i_l = 5\text{min}$. Longer i_l values require significantly longer guard time at low clock precision, but this effect is offset by the overall reduced power consumption due to longer sleep periods. Hence the slope of the $i_l = 5\text{h}$ line appears to be identical, still representing only a slight increase in power consumption. Therefore, even for unmodified tags with worst practical clock precision, the proposed PS scheme is feasible.

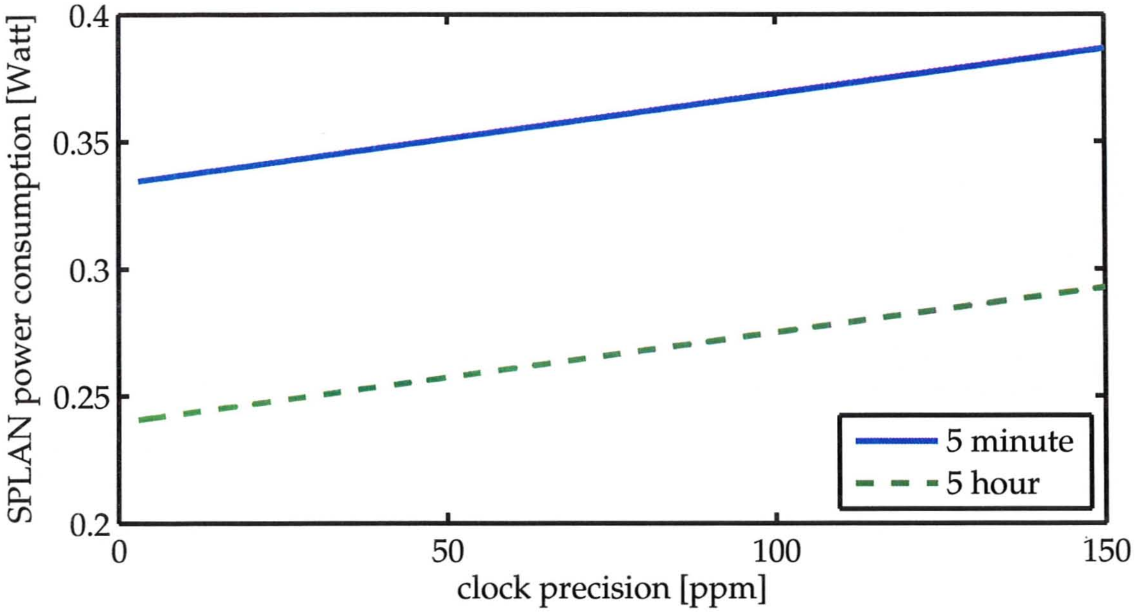


Figure A.1: Effect of clock accuracy on power consumption

A.5 AP/SPLAN Communications in pSyncF

To provide assistance to the SPLAN, APs maintain a tag transmission state table: the tag's MAC address, the timestamp of the first tag chirp, the chirp's phase, and the chirp's interval.

A SPLAN establishes a fixed schedule that allows it to communicate with its assisting AP. The communication window begins with the SPLAN sending out a broadcast beacon that contains data pairs of an AP MAC address and a corresponding last update timestamp. Each AP in the vicinity looks for its own MAC address and selects from its internal tag data table those tags whose TX timestamp is newer than that provided by the SPLAN. The AP includes the following data in its response packet to the SPLAN: the *tag identifier*—the 48-bit wide standard IEEE 802 MAC address, *timestamp*—a 64-bit wide unsigned value representing time in μs since the Unix epoch (1970-01-01T00:00:00Z), and *interval*—a 32-bit wide unsigned integer representing the length of tag chirp interval in ms.

This mechanism enables differential updates; a SPLAN is notified only about unsynchronized tags. Only a newly deployed SPLAN would receive packets containing the entire tag registration database. The mechanism also dispenses with an ACK from the SPLAN to the AP to confirm tag information reception, as any failed transmissions will automatically be repeated in the next scheduled communication cycle based on the old timestamp.

Bibliography

[1] V. Vinge, "Synthetic serendipity," *IEEE Spectrum*, vol. 41, no. 7, pp. 35–44, Jul. 2004. [Online]. Available: http://www.spectrum.ieee.org/computing/networks/synthetic-serendipity

[2] P. Sikka, P. Corke, P. Valencia, C. Crossman, D. Swain, and G. Bishop-Hurley, "Wireless adhoc sensor and actuator networks on the farm," in *Fifth International Conference on Information Processing in Sensor Networks*. New York, NY, USA: ACM, Apr. 2006, pp. 492–499.

[3] M. Ocaña, L. M. Bergasa, M. Sotelo, J. Nuevo, and R. Flores, "Indoor robot localization system using WiFi signal measure and minimizing calibration effort," in *IEEE International Symposium on Industrial Electronics*, vol. 4, Jun. 2005, pp. 1545–1550. [Online]. Available: http://www.depeca.uah.es/personal/mocana/Publicaciones/isie2005_wifi.pdf

[4] V. Stantchev, T. Schulz, T. D. Hoang, and I. Ratchinski, "Optimizing clinical processes with position-sensing," *IEEE IT Professional*, vol. 10, no. 2, pp. 31–37, Mar./Apr. 2008.

[5] J.-H. Youn, H. Ali, H. Sharif, J. Deogun, J. Uher, and S. H. Hinrichs, "WLAN-based real-time asset tracking system in healthcare environments," in *3rd IEEE International Conference on Wireless and Mobile Computing, Networking and Communications*, Oct. 2007, p. 71.

[6] P. K. Sagiraju, P. Gali, D. Akopian, and G. V. S. Raju, "Enhancing security in wireless networks using positioning techniques," in *IEEE International Conference on System of Systems Engineering*, Apr. 2007, pp. 1–6.

[7] R. A. Malaney, "Securing internal Wi-Fi networks with position verification," in *IEEE Global Telecommunications Conference*, vol. 3, Nov./Dec. 2005, pp. 1665–1669.

[8] K. W. Kolodziej and J. Hjelm, *Local Positioning Systems: LBS Applications and Services*. Boca Raton, FL, USA: Taylor & Francis Group, 2006, ch. 1. [Online]. Available: http://books.google.ca/books?isbn=0849333490

[9] J. Zheng, A. Winstanley, L. Yan, and A. S. Fotheringham, "Economical LBS for public transport: Real-time monitoring and dynamic scheduling service," in *International Conference on Grid and Pervasive Computing*. Los Alamitos, CA, USA: IEEE Computer Society, 2008, pp. 184–188.

[10] I. Ramachandran and S. Roy, "Clear channel assessment in energy-constrained wideband wireless networks," *IEEE Wireless Communications Magazine*, vol. 14, no. 3, pp. 70–78, Jun. 2007.

[11] R. Rzeczkowski, "Localisation of mobile nodes in IEEE 802.11 networks through signal strength based QCQP & SDP convex-optimisation trilateration algorithms." ECE710 project, McMaster University, Jan. 2007.

[12] J. Hightower and G. Borriello, "A survey and taxonomy of location systems for ubiquitous computing," University of Washington, Seattle, WA, Tech. Rep. UW-CSE 01-08-03, Aug. 2001. [Online]. Available: http://seattle.intel-research.net/people/jhightower/pubs/hightower2001survey/hightower2001survey.pdf

[13] R. Bajaj, S. L. Ranaweera, and D. P. Agrawal, "GPS: Location-tracking technology," *Computer*, vol. 35, no. 4, pp. 92–94, 2002.

[14] M. G. Kuhn, "An asymmetric security mechanism for navigation signals," in *Information Hiding Workshop*. Springer, 2004, pp. 239–252.

[15] G. Borriello, M. Chalmers, A. LaMarca, and P. Nixon, "Delivering real-world ubiquitous location systems," *Communications of the ACM*, vol. 48, no. 3, pp. 36–41, Mar. 2005. [Online]. Available: http://www.seattle.intel-research.net/pubs/gp.pdf

[16] "u-blox launches ultra-low power GPS technology platform u-blox 6," u-blox, Sep. 2009. [Online]. Available: http://www.u-blox.com/images/stories/PressReleases_adhoc/2009/01092009_u-blox_6_en.pdf

[17] R. Want, A. Hopper, V. Falcão, and J. Gibbons, "The active badge location system," *ACM Transactions on Information Systems*, vol. 10, no. 1, pp. 91–102, 1992. [Online]. Available: http://web.media.mit.edu/~dmerrill/badge/Want92_ActiveBadge.pdf

[18] R. Want, B. Schilit, N. Adams, R. Gold, K. Petersen, J. Ellis, D. Goldberg, and M. Weiser, "The PARCTAB ubiquitous computing experiment," Xerox Palo Alto Research Center, Tech. Rep. CSL-95-1, Mar. 1995. [Online]. Available: http://sandbox.xerox.com/parctab/csl9501/paper.html

[19] A. Juels, "RFID security and privacy: A research survey," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 2, pp. 381–394, Feb. 2006.

[20] L. M. Ni, Y. Liu, Y. C. Lau, and A. P. Patil, "LANDMARC: Indoor location sensing using active RFID," 2003, pp. 407–415. [Online]. Available: http://www.cs.ust.hk/~liu/Landmarc.pdf

[21] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-less low-cost outdoor localization for very small devices," *IEEE Personal Communications Magazine*, vol. 7, no. 5, pp. 28–34, Oct. 2000.

[22] P. Bahl and V. N. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," in *INFOCOM 2000: 19th Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 2, Mar. 2000, pp. 775–784. [Online]. Available: http://research.microsoft.com/en-us/groups/sn-res/infocom2000.pdf

[23] S. Pandey and P. Agrawal, "A survey on localization techniques for wireless networks," *Journal of the Chinese Institute of Engineers*, vol. 29, no. 7, pp. 1125–1148, Nov. 2006. [Online]. Available: http://www.crt.ntust.edu.tw/jcie/pdf/29-7-PDF/1125-1148.PDF

[24] J. O. Roa, A. R. Jiménez, F. Seco, J. C. Prieto, and J. Ealo, "Optimal placement of sensors for trilateration: Regular lattices vs meta-heuristic solutions," in *EUROCAST*, ser. Lecture Notes in Computer Science, R. Moreno-Díaz, F. Pichler, and A. Quesada-Arencibia, Eds., vol. 4739. Springer, 2007, pp. 780–787.

[25] A. LaMarca, Y. Chawathe, S. Consolvo, J. Hightower, I. Smith, J. Scott, T. Sohn, J. Howard, J. Hughes, F. Potter, J. Tabert, P. Powledge, G. Borriello, and B. Schilit, "Place lab: Device positioning using radio beacons in the wild," in *3rd International Conference on Pervasive Computing*, May 2005. [Online]. Available: http://www.placelab.org/publications/pubs/pervasive-placelab-2005-final.pdf

[26] Y.-C. Cheng, Y. Chawathe, A. LaMarca, and J. Krumm, "Accuracy characterization for metropolitan-scale Wi-Fi localization," in *3rd International Conference on Mobile Systems, Applications, and Services*, Jun. 2005. [Online]. Available: http://www.placelab.org/publications/pubs/IRS-TR-05-003.pdf

[27] T. Jiang, H. J. Wang, and Y.-C. Hu, "Preserving location privacy in wireless LANs," in *MobiSys 2007: 5th International Conference on Mobile Systems, Applications, and Services*. New York, NY, USA: ACM, 2007, pp. 246–257.

[28] A. Neri, A. D. Nepi, and A. M. Vegni, "DOA and TOA based localization services protocol in IEEE 802.11 networks," in *10th International Symposium on Wireless Personal Multimedia Communications*, Jaipur, India, Dec. 2007.

[29] Institute of Electrical and Electronics Engineers, "IEEE std 802.11TM-2007: Part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications," New York, NY, USA, Tech. Rep., Jun. 2007. [Online]. Available: http://standards.ieee.org/getieee802/802.11.html

[30] J. C. Engstrom, C. Gray, and S. Nelakuditi, "Clear channel assessment in wireless sensor networks," in *46th ACM Southeast Conference*, Mar. 2008. [Online]. Available: http://reu.cse.sc.edu/2007papers/EngstromPaper.pdf

[31] Y. Chetoui and N. Bouabdallah, "Adjustment mechanism for the IEEE 802.11 contention window: An efficient bandwidth sharing scheme," *Computer Communications*, vol. 30, no. 13, pp. 2686–2695, 2007. [Online]. Available: http://www.irisa.fr/armor/lesmembres/Nizar/papers/comcom07.pdf

[32] M. Heusse, F. Rousseau, R. Guillier, and A. Duda, "Idle sense: An optimal access method for high throughput and fairness in rate diverse wireless LANs," *Computer Communication Review*, vol. 35, no. 4, pp. 121–132, 2005. [Online]. Available: http://www.sigcomm.org/sigcomm2005/paper-HeuRou.pdf

[33] A. L. Toledo, T. Vercauteren, and X. Wang, "Adaptive optimization of IEEE 802.11 DCF based on Bayesian estimation of the number of competing terminals," *IEEE Transactions on Mobile Computing*, vol. 5, no. 9, pp. 1283–1296, Sep. 2006. [Online]. Available: http://www.inria.fr/sophia/asclepios/Publications/Tom.Vercauteren/AdaptiveOptimIEEE80211DCF-IEEETMC06-Toledo.pdf

[34] Cisco Systems, "Voice over wireless LAN 4.1 design guide," San Jose, CA, USA, Tech. Rep. OL-14684-01, Mar. 2009. [Online]. Available: http://www.cisco.com/en/US/docs/solutions/Enterprise/Mobility/vowlan/41dg/vowlandg.pdf

[35] W. Li, Q.-A. Zeng, and D. P. Agrawal, "A reliable active scanning scheme for the IEEE 802.11 MAC layer handoff," in *RAWCON: IEEE Radio and Wireless Conference*, Aug. 2003, pp. 71–74.

[36] H. Velayos and G. Karlsson, "Techniques to reduce the IEEE 802.11b handoff time," in *IEEE International Conference on Communications*, vol. 7, Jun. 2004, pp. 3844–3848. [Online]. Available: http://winternet.sics.se/workshops/sncnw2003/proceedings/30T-TechniquestoreduceIEEE80211bhandofftime.pdf

[37] T. T. M. Hamdan, H. I. Sigiuk, and Y. M. Omar, "Reduction of handoff search phase time in IEEE 802.11 WLAN to fulfill real time services requirements," in *International Conference on Telecommunications*, May 2009, pp. 346–351.

[38] Institute of Electrical and Electronics Engineers, "IEEE std 802.11k$^{\text{TM}}$-2008: Amendment 1: Radio resource measurement of wireless LANs," New York, NY, USA, Tech. Rep., Jun. 2008. [Online]. Available: http://standards.ieee.org/getieee802/802.11.html

[39] S. Hermann, M. Emmelmann, O. Belaifa, and A. Wolisz, "Investigation of IEEE 802.11k-based access point coverage area and neighbor discovery," in *32nd IEEE Conference on Local Computer Networks*, Oct. 2007, pp. 949–954.

[40] C.-H. Yu, M. Pan, and S.-D. Wang, "Adaptive neighbor caching for fast BSS transition using IEEE 802.11k neighbor report," in *International Symposium on Parallel and Distributed Processing with Applications*, Dec. 2008, pp. 353–360.

[41] T. D. Todd, A. A. Sayegh, M. N. Smadi, and D. Zhao, "The need for access point power saving in solar powered WLAN mesh networks," *IEEE Network*, vol. 22, no. 3, pp. 4–10, May/Jun. 2008.

[42] A. Farbod and T. D. Todd, "Resource allocation and outage control for solar-powered WLAN mesh networks," *IEEE Transactions on Mobile Computing*, vol. 6, no. 8, pp. 960–970, Aug. 2007.

[43] F. Zhang, T. D. Todd, D. Zhao, and V. Kezys, "Power saving access points for IEEE 802.11 wireless network infrastructure," in *IEEE Wireless Communications & Networking Conference*, vol. 1, Mar. 2004, pp. 195–200.

[44] Y. Li, T. D. Todd, and D. Zhao, "Access point power saving in solar/battery powered IEEE 802.11 ESS mesh networks," in *The 2nd International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks*, Aug. 2005, p. 49.

[45] A. M. Kholaif, T. D. Todd, P. Koutsakis, and M. N. Smadi, "QoS-enabled power saving access points for IEEE 802.11e networks," in *IEEE Wireless Communications & Networking Conference*, Apr. 2008, pp. 2331–2336.

[46] C.-H. Lim, Y. Wan, B.-P. Ng, and C.-M. S. See, "A real-time indoor WiFi localization system utilizing smart antennas," *IEEE Transactions on Consumer Electronics*, vol. 53, no. 2, pp. 618–622, May 2007.

[47] M. Wallbaum and S. Diepolder, "Benchmarking wireless LAN location systems," in 2^{nd} *IEEE International Workshop on Mobile Commerce and Services*, Jul. 2005, pp. 42–51.

[48] D. C. Hogg, "Fun with the Friis free-space transmission formula," *IEEE Antennas and Propagation Magazine*, vol. 35, no. 4, pp. 33–35, Aug. 1993.

[49] J. B. Andersen, T. S. Rappaport, and S. Yoshida, "Propagation measurements and models for wireless communications channels," *IEEE Communications Magazine*, vol. 33, no. 1, pp. 42–49, 1995. [Online]. Available: http://wsl.stanford.edu/~ee359/measmod.pdf

[50] D. B. Faria, "Modeling signal attenuation in IEEE 802.11 wireless LANs - vol. 1," Stanford University, Tech. Rep. TR-KP06-0118, Jul. 2005. [Online]. Available: http://www-cs-students.stanford.edu/~dbfaria/files/faria-TR-KP06-0118.pdf

[51] N. Papadakis, A. Economou, J. Fotinopoulou, and P. Constantinou, "Radio propagation measurements and modeling of indoor channels at 1800 MHz," *Wireless Personal Communications*, vol. 9, no. 2, pp. 95–111, 1999.

[52] H. MacLeod, C. Loadman, and Z. Chen, "Experimental studies of the 2.4-GHz ISM wireless indoor channel," in 3^{rd} *Annual Conference on Communication Networks and Services Research*, May 2005, pp. 63–68.

[53] M. Ciurana, F. Barceló, and S. Cugno, "Multipath profile discrimination in TOA-based WLAN ranging with link layer frames," in 1^{st} *International Workshop on Wireless Network Testbeds, Experimental Evaluation & Characterization*. New York, NY, USA: ACM, 2006, pp. 73–79.

[54] A. Günther and C. Hoene, "Measuring round trip times to determine the distance between WLAN nodes," in *IFIP Networking Conference*. Springer Berlin/Heidelberg, May 2005, pp. 768–779. [Online]. Available: http://www.tkn.tu-berlin.de/publications/papers/hoene_paper2.pdf

[55] M. Ciurana, F. Barcelo-Arroyo, and F. Izquierdo, "A ranging method with IEEE 802.11 data frames for indoor localization," in *IEEE Wireless Communications & Networking Conference*, Mar. 2007, pp. 2092–2096.

[56] R. Yamasaki, A. Ogino, T. Tamaki, T. Uta, N. Matsuzawa, and T. Kato, "TDOA location system for IEEE 802.11b WLAN," in *IEEE Wireless Communications & Networking Conference*, vol. 4, 2005, pp. 2338–2343.

[57] A. Rice and R. Harle, "Evaluating lateration-based positioning algorithms for fine-grained tracking," in *DIALM-POMC Joint Workshop on Foundations of Mobile Computing*. New York, NY, USA: ACM, Sep. 2005, pp. 54–61.

[58] A. Nasipuri and K. Li, "A directionality based location discovery scheme for wireless sensor networks," in 1^{st} *ACM International Workshop on Wireless Sensor Networks and Applications*. New York, NY, USA: ACM, 2002, pp. 105–111. [Online]. Available: http://www.ece.uncc.edu/~anasipur/pubs/p030-nasipuri.pdf

[59] A. LaMarca, J. Hightower, I. Smith, and S. Consolvo, "Self-mapping in 802.11 location systems," in 7^{th} *International Conference on Ubiquitous Computing*, ser. Lecture

Notes in Computer Science. Springer-Verlag, Sep. 2005, pp. 87–104. [Online]. Available: http://seattle.intel-research.net/people/jhightower/pubs/lamarca2005selfmapping/lamarca2005selfmapping.pdf

[60] K. A. Remley, H. R. Anderson, and A. Weisshaar, "Improving the accuracy of ray-tracing techniques for indoor propagation modeling," *IEEE Transactions on Vehicular Technology*, vol. 49, no. 6, pp. 2350–2358, Nov. 2000.

[61] A. Tayebi, J. Gómez, F. Saez de Adana, and O. Gutierrez, "The application of ray-tracing to mobile localization using the direction of arrival and received signal strength in multipath indoor environments," *Progress In Electromagnetics Research*, vol. 91, pp. 1–15, 2009. [Online]. Available: http://ceta.mit.edu/PIER/pier91/01.09020301.pdf

[62] H. Liu, H. Darabi, P. Banerjee, and J. Liu, "Survey of wireless indoor positioning techniques and systems," *IEEE Transactions on Systems, Man, and Cybernetics—Part C: Applications and Reviews*, vol. 37, no. 6, pp. 1067–1080, Nov. 2007.

[63] Cisco Systems, "Wi-Fi location-based services 4.1 design guide," San Jose, CA, USA, Tech. Rep. OL-11612-01, May 2008. [Online]. Available: http://www.cisco.com/en/US/docs/solutions/Enterprise/Mobility/lbswifig_external.pdf

[64] AeroScout, "Aeroscout tags network utilization," Tech. Rep., 2006.

[65] Cisco Systems, "Design considerations for Cisco PanGo asset tracking," San Jose, CA, USA, Tech. Rep. OL-13268-01, 2007. [Online]. Available: http://www.cisco.com/en/US/docs/solutions/Enterprise/Mobility/pango/PanGoEx.pdf

[66] Ekahau, "Ekahau positioning engine 4.1 user guide," 2007. [Online]. Available: http://www.ubitel.ru/files/epe4.1userguide.pdf

[67] Skyhook XPS overview. Skyhook Wireless. [Online]. Available: http://www.skyhookwireless.com/howitworks/

[68] F. Alizadeh-Shabdiz, "Systems and methods of gathering WLAN packet samples to improve position estimates of WLAN positioning device," U.S. Patent Application Publication US 2008/0 008 118 A1, Jan. 10, 2008. [Online]. Available: http://www.freepatentsonline.com/y2008/0008118.html

[69] M. Gast, *802.11 Wireless Networks: The Definitive Guide*, 2nd ed. O'Reilly Media, Inc., Apr. 2005. [Online]. Available: http://oreilly.com/catalog/9780596100520/

[70] J. Postel, "Internet Protocol," RFC 791 (Standard), Internet Engineering Task Force, Sep. 1981, updated by RFC 1349. [Online]. Available: http://www.ietf.org/rfc/rfc791.txt

[71] ——, "User Datagram Protocol," RFC 768 (Standard), Internet Engineering Task Force, Aug. 1980. [Online]. Available: http://www.ietf.org/rfc/rfc768.txt

[72] R. Droms, "Dynamic Host Configuration Protocol," RFC 2131 (Draft Standard), Internet Engineering Task Force, Mar. 1997, updated by RFCs 3396, 4361, 5494. [Online]. Available: http://www.ietf.org/rfc/rfc2131.txt

[73] S. Folea and M. Ghercioiu, "Ultra-low power Wi-Fi tag for wireless sensing," in *IEEE International Conference on Automation, Quality and Testing, Robotics*, vol. 3, May 2008, pp. 247–252.

[74] Tag4M, "Tag4M firmware API." [Online]. Available: http://docs.google.com/Doc?id=dgjmjsx9_28fq593h

[75] "RapidFlash™ lithium photocell batteries," Panasonic Industrial Company, 1999. [Online]. Available: http://www.panasonic.com/industrial/battery/oem/flash/images/specs.pdf

[76] "DS3231: Extremely accurate I^2C-integrated RTC/TCXO/crystal," Maxim Integrated Products, Oct. 2008. [Online]. Available: http://datasheets.maxim-ic.com/en/ds/DS3231.pdf

[77] Y. Agarwal, C. Schurgers, and R. Gupta, "Dynamic power management using on demand paging for networked embedded systems," in *Asia and South Pacific Design Automation Conference*. New York, NY, USA: ACM, 2005, pp. 755–759. [Online]. Available: http://mesl.ucsd.edu/yuvraj/research/documents/On_Demand_Paging-ASPDAC05-Yuvraj-Agarwal.pdf

[78] C. E. Jones, K. M. Sivalingam, P. Agrawal, and J.-C. Chen, "A survey of energy efficient network protocols for wireless networks," *Wireless Networks*, vol. 7, pp. 343–358, 2001.

[79] A. L. de A. P. Zuquim, L. F. M. Vieira, M. A. Vieira, A. B. Vieira, H. S. Carvalho, J. A. Nacif, C. N. C. Jr., D. C. da Silva Jr., A. O. Fernandes, and A. A. F. Loureiro, "Efficient power management in real-time embedded systems," in *9th IEEE International Conference on Emerging Technologies and Factory Automation*, 2003, pp. 496–505.

[80] H.-J. Kim, Y.-S. Han, E.-J. Kang, Y.-J. Won, and J.-I. Park, "Compression-incorporated hibernation for low-power mobile embedded system," in *KSEA: US-Korea Conference*, no. ICTS-41, Aug. 2005. [Online]. Available: http://www.cc.gatech.edu/grads/h/hkim362/paper/2005hibernation.pdf

[81] H. L. Vu and T. Sakurai, "Collision probability in saturated IEEE 802.11 networks," in *Australian Telecommunication Networks and Application Conference*, Melbourne, Australia, Dec. 2006, pp. 21–25. [Online]. Available: http://caia.swin.edu.au/pubs/ATNAC06/Vum.pdf

[82] Y.-C. Chen, J.-R. Chiang, H.-H. Chu, P. Huang, and A. W. Tsui, "Sensor-assisted Wi-Fi indoor location system for adapting to environmental dynamics," in *8th ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems*. New York, NY, USA: ACM, 2005, pp. 118–125.

[83] "Power consumption and energy efficiency comparisons of WLAN products," Atheros Communications, 2003. [Online]. Available: http://www.atheros.com/pt/whitepapers/atheros_power_whitepaper.pdf

[84] "CM9: WLAN 802.11a/b/g mini-PCI Module," Wistron NeWeb Corporation.

[85] "Timekeeping accuracy, automatic and affordable," Maxim Integrated Products, Tech. Rep. AN3566, Aug. 2005. [Online]. Available: http://www.maxim-ic.com/an3566