Delft University of Technology
Master's Thesis in Computer Science

Wireless Sensor Platform for Sporting Applications

RAMA SUBRAMANIAN SANKARANARAYANAN

embedded software

TU Delft Delft University of Technology

Wireless Sensor Platform for Sporting Applications

Master's Thesis in Computer Science

Embedded Software Section
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology
Mekelweg 4, 2628 CD Delft, The Netherlands

RAMA SUBRAMANIAN SANKARANARAYANAN
R.Sankaranarayanan@student.tudelft.nl

1st August 2011

Author
RAMA SUBRAMANIAN SANKARANARAYANAN (R.Sankaranarayanan@student.tudelft.nl)
Title
Wireless Sensor Platform for Sporting Applications
MSc presentation
12th August 2011

Graduation Committee

prof. dr. ir. Koen G. Langendoen(chair)	Delft University of Technology
dr. ir. Stefan Dulman	Delft University of Technology
Mr. Bastiaan de Groot, MSc	Philips Research
dr. ir. Hans-Gerhard Gross	Delft University of Technology

Abstract

With Wireless Sensors widely used in various domains like home automation, industrial monitoring there is a market urge to deploy the wireless sensors in sporting applications. By deploying wireless sensors in sports, various dimensions of use-case scenarios become obvious which include monitoring sports players to help assess their fitness levels during training sessions and during play, enhance game strategy and provide TV broadcasters with lucrative statistics for the audience. As a first step to realize these use-cases, a platform to create such applications is needed to rapidly prototype devices as a proof of concept. However, to monitor professional sports players and to help make scientific analysis, a deluge of information is needed with less error margin. In this thesis, a wireless sensor platform is designed and developed, customized for creating prototypes of nodes for the sports players. Multiple gateways can be used along the boundary of the play-field to cover the entire play-field and with the mobile sensor nodes making hand-off between the gateways based on their proximity. A time-sharing mechanism is used by the nodes to gain access to the channel and is centralized at the gateway. The gateway provides authentication to which sensor node can transmit data in a round-robin manner. Experimental results show that the packet losses are around 1% with varying cases explaining that only one node communicate with the gateway at any point of time. One of the major drawbacks of such time-slotted protocols is the latency due to the failed nodes and in this protocol a mechanism is devised to mitigate this latency. The net data-rate is also enhanced by transmitting multiple packets in a time-compacted slot without linearly increasing the slot-width.

Preface

With poised between hardware and software domains in Embedded Systems in the first year in TU Delft, I decided to incline towards the software for my Master Thesis in the second year. I appropriately had an opportunity to do my Master Thesis in Philips Research which not only provided ample scope for research but also provided me Industrial hands-on experience.

I heartily express my gratitude to my thesis supervisor Prof.Stefan Dulman, who helped me sail to the shores safely during the perturbations in the early stages of my thesis. His valuable comments and expert guidance had always helped me in finding a way out in various problems I encountered. I also thank him for being more than willing to meet me whenever I requested for. This thesis would not have been possible without the constant support of my industry supervisor Mr.Bastiaan de Groot, who had been a great source of encouragement throughout this project. I wish to thank him for also being a mentor providing professional guidance which I hope to take deep through my career. I also express my word of thanks to Venkat Iyer for taking time out of his schedule to provide valuable inputs and advice through his rich experience in this field.

Thank You is definitely an understatement to make, for the unconditional support my Mom and Dad had provided me all throughout these 23 years. Nothing could match the ever-growing affection and the continuous pampering I receive from them. I would also like to thank my sister who had been greatly understanding and shouldered responsibilities during my absence in the family, which are disproportionate for her young age. I wish to thank all my friends in Netherlands who made my stay here pleasant and made me feel at home, though in a foreign land. A special mention goes to Srinivasan for putting up all along with me in the past one year and making my stay at home enjoyable.

Rama Subramanian S

Delft, The Netherlands
1st August 2011

Contents

Chapter 1

Introduction

With the advent of the Nano-age, electronic components have become appreciably smaller in size and cheaper in cost. This has allowed the consumer electronics industry to employ additional electronic components to increase the add-on value of the products. For example, a PDA or a smart phone has no intrinsic requirement to have a gyroscope built within it. But nowadays, almost every smart phone comes with a gyroscope, which facilitates the user to align the pictures depending on the orientation of the phone and also helps redefining the gaming applications. This could be only possible with the smaller size of the gyroscopes and its lower-cost, considering the additional features it provide. With such advancement in the electronics field, we have numerous sensors to detect and measure any possible quantity. There are sensors available in the market to quantify proximity, temperature, pressure, motion, position, light and so on. These sensors provide a huge amount of data, which needs to be processed accordingly to provide the user with relevant applications. In most cases, more than a single sensor is needed to either monitor a particular parameter of various subjects or to extract information through data-fusion of various sensor data. However, these sensors would be placed far apart without any physical connections between them. Wireless Sensor Networking(WSN) plays the role of collecting the information on the distributed sensors onto a single system to help analyze and process the information for better understanding.

Nowadays, people have become more cautious towards thier fitness and are thus thronging the Gym and other fitness centers. They are in need of some real measurement of their physical activity they carry out to analyze their fitness levels. Mobile Apps have been developed as shown in Fig 1.1 to target these audience and have been a great success. These Apps analyze the intensity of the physical activity carried out by the individual which in-turn helps the person to adapt his training schedules better. This sheer interest on fitness levels has spread to their respective favourite sports teams

Figure 1.1: Sports Monitoring Apps in Smart Phones

also. People have also been curious on the fitness levels of their favourite stars and this has taken the way the sports is discussed to a different level. Moving from the personal sports care to professional sports analysis requires a higher degree of accuracy in the obtained data. To compare athletes who are supposedly of the same fitness levels, it requires higher precision to help make a better scientific decision. By combining the advancement in the sensors with the wireless technology, a deluge of information can be provided.

Though WSN had been in the foray of research for more than two decades,it has been put into action by the industry only in the recent years, mainly in the field of home automation, industrial monitoring, and health sector. There is a market urge to employ the WSNs in the sports sector to facilitate the coaching process and also to enhance the viewer experience. In the field of sports applications, there is tremendous scope for the deployment of WSNs, in this field. The motion sensors like accelerometer provide a clear picture of the movement of the sports players in the field. In addition, there are lots of physiological sensors readily available in the market, which could analyze the intensity of the physical activity of the players. But these sensor values from each of the players, needs to be transferred to a central location, to either process them or present it in an understandable manner.

To facilitate this, the sensor module is attached with a RF module to enable wireless transmission of the sensor data. A protocol has to be devised to enable smooth communication among these modules.

1.1 Requirements and Purpose of the Platform

In this project, a wireless sensor platform is required for rapid-prototyping targeting sports applications. The platform should be able to provide the necessary software for building a wireless network for the sports applications, which includes necessary software for both the sensor nodes and the gateways. With this platform the following devices need to be developed:

1. *Sensor Nodes*: Primarily considered as the transmitting nodes, the sensor nodes are those which are placed on each of the player's body. These nodes will be physically connected with one or more sensors. These nodes have to read the sensor data and transmit it to a gateway. The sensor nodes are battery powered.

2. *Gateway(s)*: The Sensor nodes transmit to the gateways, which then log the data into a PC through serial connection. The gateway can also have control over the entire network. Usually, the gateway should be mains powered. There can be one or more gateway in the network.

3. *Central Server*: If more than one gateway is present in the network, the gateways need to be connected to a single PC i.e., Central Server to log the data from all the sensor nodes in the network and process it. If there is only one gateway, the PC to which the single gateway is connected will act as the central server. The central server should also be mains powered.

The sensor nodes and the gateways can be realized from the platform SDK. The user will be able to set-up the wireless communication between the sensor nodes and the gateways, out-of-the-box by using the platform. Since the SDK is developed as a platform, the generic characteristic of the software should be emphasized and hence, there should be as little as hard-coding as possible. For example, the number of sensor nodes and number of gateways are not fixed and it can vary from one sports application to another. The platform SDK should support all such possibilities.

Though there are various sports to which this platform can be used, Soccer is used as a bench-mark application for this project. The dimensions of the playfield to be covered by the wireless communication is 110m* 70m. There will be a total of 22 concurrent players/sensor nodes sending out sensor data.

The wireless platform should also be able to support adding/removing of sensor nodes/players, even when the game is in progress. To bench-mark for the amount of data to be transmitted, it is considered that each of the sensor nodes is attached with a 3-axis accelerometer, streaming data at 200Hz. In terms of bytes/per second, it is calculated as 3(axis)*200(Hz)*22(nodes) = 13200 bytes/second=13.2 KBs/sec.

1.2 Problem Statement

The objective of this thesis is to develop a wireless sensor platform which could be used to develop rapid-prototyping for sporting applications. The platform should be able to support most of the sensors which are readily available on the market. Respective interfaces should be available on the software to accommodate these sensors and provide necessary functions to read and process the sensor data from these sensors. One of the main applications of the Platform is to hook-up an Accelerometer with each of the Sensor Node. These accelerometers will keep streaming data continuously; hence the collision of the radio messages on the gateway side needs to be minimized.

The main aim is to collect data, from the sensors attached to the players body, through a wireless network and logs them onto a central server. The data collected should be able to provide the details of the activities performed by the players and should be logged and displayed in the central server in real-time.The main applications of this platform is to use the sensor data to analyze the players fitness levels, record players activity, to facilitate the coaching process and improve strategy planning during the game. The same infrastructure can also be used to show vital and interesting statistics to the audience via TV broadcasters.

These sensor data should be available in the Central Server in real-time. For Example, most approaches of Wireless Sensor Networks in the Sports Applications log the sensor data on-board and then transfer it to a PC for analysis of the Player's activity. However, in this application, the sensor data should be sent in real-time to a Central Server, to enable the Coach or the viewer to analyze the Player's activity in real-time.

Considering the dimensions of the playfield, it is not feasible to cover the entire playfield in a single hop by the low-power radio modules. However, it should be ensured that all the players are able to send their sensor data to the central server, wherever they are placed in the playfield. Summarizing these problems, the following list provides a brief description of the problems:

- To encompass the entire playfield with a low-power wearable sensor nodes
- To aggregate data from all the sensor nodes in a central server in real-time
- To support high data-rate streaming sensors at each node/player

1.3 Thesis Contribution

As an ultimate goal of this thesis, a wireless sensor platform mainly targetted at the sports applications is developed. A customized radio protocol is developed to best suit this application taking into account the possible advantages and the dis-advantages, the sporting scenario is to provide. The main contributions of this thesis are:

- Designing a network topology with a chosen hardware specifically for the sports applications
- Design and implementation of a time-sharing protocol customized to target the sports application for the designed topology.
- Adapting the time-sharing scheduling mechanism to mitigate the latency caused due to failed nodes
- Enhancement of the net data-rate of the network with time compaction for multiple packets in a single slot

1.4 Thesis Organization

This report is further organized as follows: In Chapter 2, a brief description of the related works and necessary background information are provided. Chapter 3 details various choices made including the hardware and software choice made for the platform. Chapter 4 gives the description of the protocols considered for this project and also includes the detailed description of the protocol created for this project. Chapter 5 details the implementation of the selected protocol and the problems faced while implementing it and also provides the relevant adaptations made to counter the problems faced. Chapter 6 provides the software validation tests and also the performance test results. Chapter 7 provides the Conclusion and Future Work that can be carried out.

Chapter 2

Background

Firstly, this chapter gives a brief on the overview of the wireless sensor networks in Section 2.1 and Section 2.2 provides a short description on the Sports Applications to which it is to be employed. Later, Section 2.3 surveys various works in the research arena related to this project and also briefs on the work carried out in the Industry sector in line with these Sports Applications. This chapter concludes with the discussion on the above mentioned details in Section 2.4 and how this project can be taken forward with these informations.

2.1 Introduction to Wireless Sensor Networks

In this section, a brief introduction to the Wireless Sensor Networking is given, explaining the terminologies used in this domain. A wireless sensor network is a collection of nodes organized into a cooperative network [1]. Considering these nodes as a black-box, each node will have one or more sensors which provide the input to the system and a RF module which acts as the output of the system. The sensor nodes can have a microcontroller to provide some processing capability to the system. Sensors are chosen depending on the application and what the user wants to measure. As mentioned in the Chapter 1, with the advancement in the electronics, the size of the sensors have gone down greatly, allowing sensors to be worn by the user. This has resulted in the evolution of *wearable sensors*. The RF modules used widely with the Sensors are of low power, which increases the battery-life time of the system.

In the wireless sensor networks, a group of sensor nodes transmit the data to a particular destination called a *central server*, where the data from all the sensor nodes are aggregated in real-time. The data can be transmitted either in single hop or through a series of hops to reach the final destination. There are different topologies specified for defining the way the data reaches

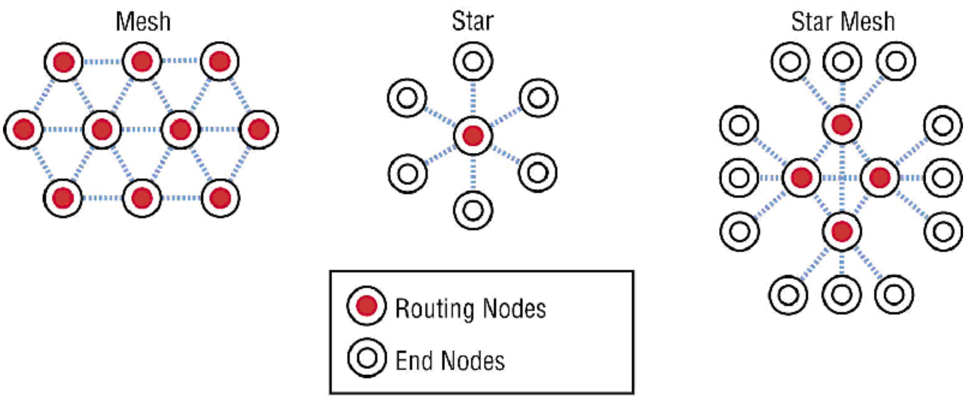

Figure 2.1: Various Topologies of WSNs

the Central Server. Some of the common topologies are shown in 2.1. Since there is a high chance that two or more nodes try to access the same channel to transmit to the same receiver, a protocol should be used to avoid collision. This protocol, usually called the *Medium Access Control (MAC)*, determines which of the nodes should transmit at a particular instant, thereby avoiding or reducing the chance of collision [1]. There are various ways with which the channel can be handed over to a particular node during periods of contention. Some protocols sense the channel to find whether there are any transmissions currently in the particular channel and if so they back-off for a particular period and then try to access the channel.In some cases, each of the node will have a particular time-slot when to send. It is taken care by the protocol that no other nodes in the contention neighborhood transmit at that instant, to avoid any possible collision. Owing to the distributed nature of the Wireless Sensor Networks, the design of the protocol is an important factor in the performance of any Wireless Sensor Network.

2.2 Overview of Sports Applications

With a wide range of use-cases for the wireless sensor networks, Sports Application is also a viable use-case scenario and promises to be a commercially profitable product too. The application will require each of the Players to be tagged with one or more sensors. The data collected from these players need to be aggregated in a single place, preferably on real-time. The major targets of the entire system are the coaches and the audience. With various physiological data available to the coaches, they can effectively estimate the fitness levels of the players either during the game or during training. This will help plan the strategies of the game better. The coaches can also customize their training patterns for individual players depending on their fitness levels. On the other hand, TV broadcasters can help themselves to keep the audience glued to the lucrative statistics, by broadcasting the information collected through the wireless system.

Some of the important factors that are to be taken under consideration while deploying the Wireless Sensor Networks in the Sports field are the dimensions of the playfield and number of players taking part in the game. The table 2.1 below gives a listing of some of the common sports and afore mentioned factors.

Name of the Sport	Dimension of the Playfield	Active Players
Volleyball	18m*9m	12
Basketball	28m*15m	12
Handball	40m*20m	14
Hockey	91m*55m	22
Soccer	110m*70m	22

Table 2.1: Different Sports Requirements

The wireless network system to be developed should be capable of comprising these requirements. The data from all these active players, located anywhere in the play-field should be collected in a single place. The Substitution of the players is a common aspect in any team-sports and the wireless system should thus be capable of adding and removing nodes even when the game is in progress. The coach or the audience should be able to view the new player in the network while the removed player should not constitute a part in any form in the network. As can be noticed from the table 2.1, some of the sports are outdoor applications while others are indoor. So, the wireless network system should be capable of performing in both indoor and outdoor environments effectively.

2.3 Related Work

Though wireless sensor networks had been in the research arena for more than two decades, it had been mainly deployed only in the field of Home Automation, Industrial Monitoring, and Healthcare sector. However, there are many other fields which can tap the potential of Wireless Sensor networks to improve its own efficiency. One such field is the Sports Applications. In the Sports Applications, not only the fitness levels of the Players are of concern but also interesting statistics recorded during the play are also important like the Height of the Jump, acceleration of the latest sprint and any other interesting statistics. This section details the previous works in the Sports Applications employing wireless sensor networks to monitor the Players during the game.

In the project MarathonNET [2], a customized wireless sensor network is developed for monitoring the runners, in a Marathon race. This project is one of the very few projects that have deployed WSN in a Sporting Application and has also demonstrated successfully that WSN could be a great tool in Sports Applications. The authors developed a system to acquire the data from each of the runners and collect it in a central server. The data from each of the runner contains 7 bytes (every minute), with a PlayerID, Heart Rate Monitor value and the information on the location of the Player. This data are then uploaded to the Internet to enable audience to track and monitor their favorite players during the game.

As a part of this Project [2] , Pfisterer et.al conducted a connectivity analysis, in which they manually collected information on the distribution of the runners in the track during the various stages of the game. From their study, it was inferred that at any point of the time, a high degree of connectivity is available between the runners. They based the design of their network topology on these results including the various points where to place the base stations along the track, the distance between the base stations among others. With the assurance provided by the connectivity analysis, they employed a multi-hop algorithm to route the data to one of the base stations. The idea was to pass the data to one of the neighbors until it finds a base station. The data then collected are passed onto a central server from which it can be either analyzed or uploaded to the Internet for audience viewing.

One of the major drawbacks of this project is its inherent assumption of the distribution of the Players along the track which enables guaranteed multi-hop paths. This assumption does not hold good for all sports. In most Sports, there is a possibility that most of the Players are concentrated in a particular spot, with one or few players isolated completely. MarathonNet also employs very low data-rate. However, in most of the field Sports Applications, sensors like Accelerometer, Gyroscopes prove to be a reasonable use-case. And, these sensors push out huge amount of data in the order of hundreds of bytes every second.

As a first step in monitoring players in field sports, Alan et.al. developed a system in which the various sensor values of the players are logged on-board in the memory during the game [3]. These sensor data are then retrieved post-event to analyze the player's performance. One obvious limitation of this system is its in-capability to provide real-time analysis of the Player's performance. Another limitation is the constraint on the on-board memory space. In [4], Wilhelm et.al., developed a Sports Performance Analyzer(SPA), which is intended to be deployed for indoor team sports applications, with an area coverage of 40m*20m. High resolution video system is used to track the movement of the players, while their physiological data are collected through a wireless network system.

These above studies give a picture of the various technological needs for the deployment of the WSNs in the Sports Applications. However, as mentioned in the section 2.1, the design of the protocol defines the performance of the wireless system. There are numerous studies based on the design and implementation of such protocols, while some of the related protocol studies are discussed here. In these studies, the protocols define how and/or when the nodes can transmit message by accessing the common channel and also defines how the message will be routed to the destination.

In [5],Degesys et.al., created a collision-free TDMA protocol, based on the DESYNC algorithm. Some of the main advantages of this protocol are that it does not require a global clock for synchronization and it also automatically adjusts with the changes in the active nodes in the network. Each of the node senses its location to fire based on the shot it hears just before it fires and also just after it had fired. Then, each of the node moves accordingly to its spot for its next shot. After a certain rounds of firing by all the nodes, the network completely gets de-synchronized and each node can fire without any potential collision from other nodes in the network. This allows for perfect inter-leaving of the messages from each node in the network and also adapts the schedule to adding and removing of nodes and thereby utilizing the bandwidth to a possible maximum extent, which many TDMA algorithms overlook[6].

Numerous protocols are developed based on the Low Energy Adaptive Clustering Hierarchy (LEACH), put forward by Chandrakasan and Hari Balakrishnan [7]. LEACH employs a self-organizing, adaptive clustering mechanism, which assigns different nodes in the cluster as the local base station or cluster head. This distributes the energy burden of being the base station to different nodes and thereby extending the total life-time of the system. Many protocols customized the LEACH concept for their application and suggested various ways in determining the cluster-head [8][9][10]. In [6], G.P.Halkes et.al, proposed a MAC protocol, CrankShaft, targeting dense wireless sensor networks. It uses distributed slotting mechanism for the nodes in the network to schedule the wake-up time for each nodes based on an offset from the start of the frame.

In most of the Sports Applications discussed above, it requires the deployment of the Wireless sensor networks in the outdoor space. Outdoor deployment of the Wireless sensor networks provides some interesting scenarios. With little obstacles between any transmitter and the Receiver, line of sight path is obtained in most cases, excluding the possibility of players obstructing the path. However, the advantage of multi-path is lost in the outdoor as compared with the indoor deployment. In [11] M.Arif et.al, studied the Outdoor propagation analysis of the 802.15.4 and have inferred that the position of the nodes also play a critical role in the packet reception ratio. This should also be taken into account while deploying the wearable solutions for the Sporting Applications.

As a reflection of the various researches carried out, different ranges of products are lined up for monitoring the players. In Philips Research,Eindhoven a prototype of the wearable physical activity monitoring is developed internally. The device is capable of measuring a couple of physiological parameters and logs them to be analyzed with the help of customized software post-event.Few products available in the market like Apple's iPod [12] provide the fitness levels of the wearer, in collaboration with Nike.It motivates the wearer to increase their physical activity and also analyze the same post-run. Clearly, there has been this market urge to take this monitoring of Sports Applications to next level, with real-time monitoring for all players in the playfield at a centralized place.

2.4 Conclusion

The surveys above provide information on various aspects needed by the project. The MarathonNet project is a great motive to employ WSNs for Sports Applications but due to the inherent properties of the Marathon Sport, the project is being customized only for this application.[3] and [4] helps understand the possible network topology that can be deployed in the playfield for various team-sports. The protocols discussed also suggest the design approach that can be taken forward for this application. The rest of the report will detail the approaches and choices made during the design and Implementation of the project and will further detail the characteristics of the entire system.

Chapter 3

Platform Choices

As a first step in the project, different choices relating to the hardware to be used, software tools to be used have to be decided before developing the SDK for the platform. There are numerous factors, relating to the development of this platform, which has to be made clear. This will reduce quite a lot of development time in the project. These decisions are made based on extensive studies performed on the various choices available and also against the requirements to be meted out. This chapter briefs the choices made in these aspects.

3.1 Hardware Choice

The initial step in the platform building is choosing the Platform Hardware. The common architecture of a wireless sensor node is shown in Fig. 3.1.

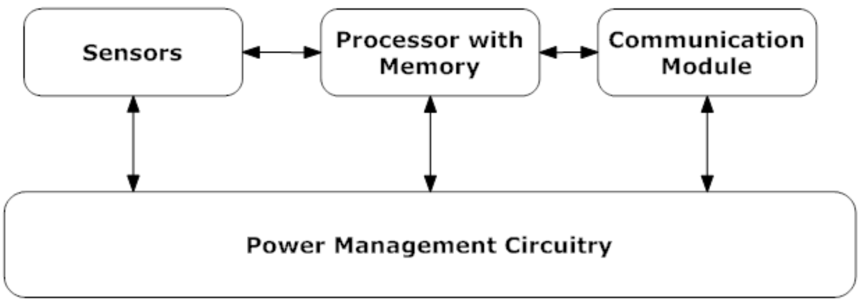

Figure 3.1: Wireless Sensor Node

As seen in Fig. 3.1, the main components of any Wireless Sensor node are

1. **A Processor with some Memory**
   Usually a microcontroller, to control the operations of other modules of the Wireless sensor node. It also provides some computation

capability to the node and to perform tasks and process the sensor data.

2. **A Communication Module**

   This module is responsible for the wireless transfer of the sensor data and can however be capable of acting as a transceiver. Here, we are limiting the operation of the communication module to the ISM band and will be operating as RF Transceivers.

3. **Sensors**

   Application-specific sensors can be used. These serve as the input to the system. The processor can either just transmit the raw data or perform some processing with it before transmitting it. Some examples for the Sporting Applications are Accelerometer, Gyroscopes, Heart-rate monitor and Jump Sensor among others.

4. **Power Management Circuitry**

   This module is used to supply stable voltage to the other modules of the wireless sensor node. Usually, the development kits come with the on-board power management circuitry.

Having briefed on the various componenets of the sensor node, in this section we explain the possible architectures of the two main modules of the node viz. Microcontroller and RF Transceiver. Fig 3.2 shows the various possible architectures for a wireless sensor node.

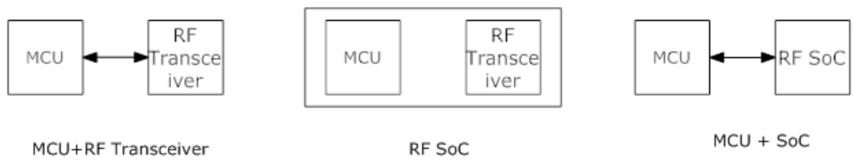

Figure 3.2: Architecture Hardware

1. **Independent Microcontroller and RF Transceiver**

   In this architecture of the Sensor Node, the RF Transceiver will exclusively take control of the communication. The MCU can be used to perform other tasks such as data collection from the sensors and provide the computatuional capability for the platform and also contols other peripherals.

2. **Single Chip solution**

   In this kind of architecture, the Microcontroller and the RF Transceiver will be embedded on a single chip, which redcues the form-factor to a great extent. These type of solutions are commonly called as RF SoCs.

3. **MCU + RF SoC**

   This architecture provides more flexibility and increases the computation power of the node but naturally the form-factor also increases. This is a good candidate for applications which need high local processing capabilities.

Based on the above architecture discussion, it was decided to use the Architecture 2: RF SoC for the following reasons:

1. Smaller form-factor. Saves board-space.
2. Lesser development time and effort since MCU and RF Transceiver reside in the same chip. No need to set-up explicit communication interfaces between MCU and RF Transceiver.
3. This application does not need basically involve local-processing. Only raw-data transmission is the requirement. However, it can be extended with an external MCU, when needed.
4. Cost-Effective.

The next step in this phase is to select the specific hardware, viz., RF SoC to be used for this platform. With the help of [13] which provides an evaluation of various radio technologies, it was decided to use the standard 802.15.4 modules for the RF. A handful of hardware from various vendors were studied and the suitable SoC is chosen after comapring against the requirements for this project. The short-listed candidates for the selection are Texas Instruments CC2530, Atmel's ATmega128RFA1 and FreeScale's MC1321X. These SoCs are short-listed from a numerous other SoCs from various vendors, based on the previous experiences within Philips Research and their own basic capabilities.

From the table 3.1, a close choice between ATmega128RFA1 and TI CC2530 has to be made. After further analysis it was decided to use Texas Instruments CC2530 for the Platform for the following reasons:

1. Availability of full-fledged development Kits, which reduces the development time significantly.
2. Satisfies all the requirements expected from the hardware for the Platform.
3. Better compatibility with MSP430, the Digital Systems and Technolgies, Philips Research standard platform application processor.

	TI CC2530	**ATmega 128RFA1**	**MC1321X**
Data-Rate	250Kbps	250Kbps	250Kbps
Memory	256K Flash 8K RAM	128K Flash 16K SRAM 4K EEPROM	60K Flash 4K RAM
Communication Interfaces	2 programmable USART modules	2 programmable USART modules	2 SCI modules
ADC	12bit-8 channel ADC	10bit-8 channel ADC	10bit-8 channel ADC
Accesible GPIOs	21 GPIOs	38 GPIOs	32 GPIOs
Tx Current	29mA	18.6mA	35mA
Rx Current	24mA	18.6 mA	42mA
Timers	2 * 8-bit Timer 1 * 16-bit Timer 1 MAC Timer	4 * 16 bit Timers 2 * 8 bit Timers	2 *16-bit Timers
Development Boards	Available in various customizations by TI, well documented with good support.	Not available directly by Atmel Available only from 3-rd party.	Not available directly by Atmel Available only from 3-rd party

Table 3.1: Hardware Comparison

3.2 Software Choice

With the choice of the Hardware being made, the vendors of the hardware provides various choices of software compatible to be used with the specific hardware. There are three different software available from Texas Instruments for the 802.15.4 compliant radio from the same vendor. The main advantage of using the software available from the vendor is that though they are proprieatry properties,there is no need to pay any royalty when used with the Hardware from the same vendor.

The three Software stacks available are:

1. **SimpliciTI**
   As the name suggests, a simple radio protocol compatible with CC2530. SimpliciTI also needs very less code space of less than 8K Flash. It is capable of supporting peer-to-peer communication and simple star network with one co-ordinator.[14]

2. **Z-Stack**

   Z-Stack is TI's ZigBee compliant protocol stack for a growing portfolio of IEEE 802.15.4 products and platforms. Z-Stack is compliant with the ZigBee 2007 (ZigBee and ZigBee PRO) specification, supporting both ZigBee and ZigBee PRO feature sets on the CC2530 SoC.[15]

All these software are extensively documented and provide necessary details for the developers. It was decided to use the SimpliciTI software for the following reasons:

1. Simple C-based API available, which reduces the start-up time when developing Application with the Platform.

2. Less weight and very primitive protocol which allows for customization correspondingly for our application.

3. Fully-available source code for the developers and access to any modifications in it.

4. Extensively documented to provide support for development of products based on it.

The choice is also made in line with some of the approaches handled for this project as discussed in Section 3.3. The SimpliciTI stack can be compiled using the IAR Embedded Work Bench with the 8051 Tool chain.

3.3 Network Topology

This section details the choices we made on the network and communication set-up in terms of the topology. The requirement in an abstracted level is to obtain the data of various players spread across the playfield in a single location. The area of the deployment of the wireless sensor network is studied and provided in the Table 2.1. The playfields are usually rectangular in shape with a smooth landscape with very minimal obstacles. This allows for any topology to be deployed as shown in 2.1. As a first step in this phase, we decide on the topology of the network to be deployed in the Playfield.

The platform should be capable of transmitting raw-data on a continuous manner from all the participating players. As stated in Section 1.1, the receiver should be capable of aggregating data at a rate of 13.2KBps which relates to a 3-axis accelerometer attached with each of the player's sensor node, streaming data at 200Hz each. The Sports application considered here is the Soccer with 22 active players at any instant of time. This requirement on the high data-rate influences the number of hops needed to reach the destination. In a multi-hop networking, the available bandwidth gets halved after the first hop, with each hop thereafter. In such a networking, the node passes its data to one of its neighboring node which then appends the received data with its own data and passes it to one of its neighbor. This process is repeated until the destination is reached. At each hop, the available bandwidth for that node decreases proportionally to the number of hops that has already been traversed.

As seen in the [2], a high degree of connectivity is needed for application to reliably transfer data through neighbors. But in Sports Applications like Soccer, it is highly possible that either a group of players can be clustered tightly with a few other players isolated completely from the cluster. For example, the Goal Keeper may at many times during the play be isolated from the rest of the Players and hence will be forced with low connectivity. After analyzing these reasons, it was decided to use as much as Star Topology as possible.

With the low-power 802.15.4 RF modules aided only with the PCB Antenna, as would be in the case of the wearable solutions, it is not possible to cover the entire play-field in a single hop. For example, as mentioned in the section ref RelatedWork , the prototype of the wearable solution developed internally within Philips research, was tested in outdoors to measure its range and was found to be 60 meters. With this range, it is possible to cover only a few of the Sports Applications mentioned in 2.1 on single hop. As a result to cover larger playfields, such as Soccer, Hockey Playfield, we chose to deploy multiple receivers along the sides of the Playfield. These receivers will be stationary and be mains powered.
The next important decision to be made is the manner in which the Stationary receivers are to be connected with each and finally to the Central Server.As seen in Figure 3.3, the Receivers are connected through an Ethernet network for the following reasons:

1. Higher bandwidth can be provided through the Ethernet network as compared with the Wireless Networking, for the same cost.
2. Lesser latency in the Wired-Network.
3. Better reliability and Minimal Interference.
4. Easier set-up and feasible to deploy wired-network along the sides of the Play-field.

Taking all these factors into account, the Network Topology to be used in the Platform design can be shown as 3.3 for a Soccer Playfield with 4 Receivers along the sides of the playfield, which could provide coverage

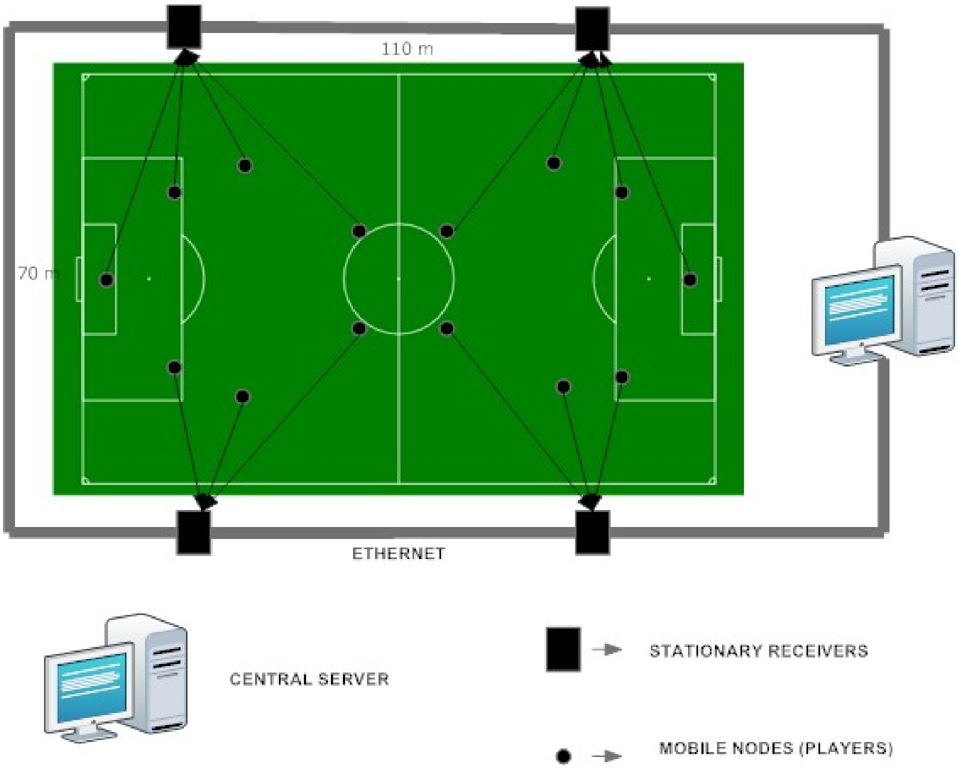

Figure 3.3: Proposed Topology for Soccer

to the entire play-field. The number of receivers depends on each of the Sports Applications and hence the Platform should provide flexibility with the number of receivers.

3.4 Conclusion

This chapter details the various decisions made with regard to the Hardware and Software to be used for this Platform in the Section 3.1 and Section 3.2 respectively.The Platform will be developed with Texas Instruments CC2530, with one of the proprietary software provided by Texas Instruments, SimpliciTI. Further, in Section 3.3,the network topology to be used for the Sports Applications is also explained and an example scenario is also given. In the next Chapter 4,the protocol in which the data will be collected from the mobile nodes onto the Gateways will be explained.

Chapter 4

Protocol Design

In this chapter, we will discuss the design of the protocol by moving through the various steps underwent while designing the protocol. The protocol has to be designed keeping in mind the requirements specified for the Platform and also for the topology mentioned in the Chapter 3.

4.1 Channel Accessing Mechanism

This section will detail the manner in which the sensor node will access the wireless channel to transmit its data to the receiver. As mentioned in 2.1, there are two mechanisms by which the channel can be usually accessed by the sensor node, which are Channel Sense multiple access and Time-shared mechanisms.

4.1.1 Carrier Sense Mechanism

In the Channel sense mechanism, the radio module will snoop the channel to sense if there is any transmission in the channel currently. Most of the WLANs and WPANs employ this mechanism and the nodes in these networks sense the channel for any on-going transmissions before transmitting. They start their own transmissions only when it infers that there is no other transmission or back-off for a specific time if they find any transmissions and retry later. This kind of channel sensing is usually referred to as Clear Channel Assessment(CCA), a PHY layer activity and is an important component of the CSMA/CA MAC protocols[16].

4.1.2 Time-Shared Mechanism

In the Time-Shared mechanisms, usually a single frequency channel is shared by multiple transmitters and each of the transmitters gets a particular time-slot to transmit their data. Only the particular node should ideally attempt to access the channel, which minimizes the chance of interference. When time-sharing is used in a wireless sensor system, the sensor data can be batched and sent whenever the node gets its slot and control over the channel.

Having briefed on the two mechanisms for channel access, Table 4.1, shows the comparison of the two mechanisms [17] and also helps choose the appropriate one for the platform.

	Carrier-Sense	**Time-Sharing**
Power Consumption	High	Low
Bandwidth Utilization	Low	Maximum
Preferred Traffic	Low	High
Network Change Response	Good	Poor
Effect of Packet Failure	Low	Latency
Synchronization	NA	Crucial

Table 4.1: Comparison of Channel-Access Mechanisms

With this comparison, it is decided to use the Time-Shared mechanism for this application, since it has a high requirement on the data-rate and hence channel utilization is a major factor. It should also be noted that with a wearable solution, power consumption should also be kept as low as possible. It should also be noted that there are a few disadvantages with the ideal TDMA approach like the need for synchronization of the sensor nodes and packet or node failure causing latency in the network. However, these issues are to be mitigated with the further design approach of the system.

In the Time Shared channel access mechanisms, at any instant of time, there can be only one node transmitting and no other nodes in the vicinity can transmit that would potentially cause any collision. A scheduling algorithm should be implemented and the nodes should transmit based on these scheduling algorithm. The scheduling algorithm can be made to either reside internally in the sensor nodes or reside only in the Gateway with full control to the gateway. The former is called as a Distributed Algorithm, while the latter can be termed as a Centralized algorithm. As mentioned in the 4.1, one of the few dis-advantages of the TDMA kind-of protocol is that they fare poorly with dynamic network change. In other words, the scheduling algorithm needs to be updated very frequently if the network is prone to frequent changes. In our Sports Applications, with multiple receivers and mobile player nodes in the network, it is highly possible to be faced with frequent network changes.

4.2 Proposed Time-Sharing Centralized Protocol

To solve the above mentioned issues, we developed a simple Time-Sharing mechanism to be employed in this project. As a first step, we decided to have the scheduling algorithm reside in the Gateway. This gives control to the entire network and also will provide necessary provisions to handle the changes in the network. In our protocol, the gateway will have the full control as to decide which of the nodes in the network can talk at any point of time. This type of centralizing the slot scheduling mechanism works good for networks of these sizes and hence we decided to use this technique in the protocol. Since the gateway is in full control with the network with all the information on any new nodes arriving into the network, it made sense to use the centralization of the scheduling algorithm for this platform.

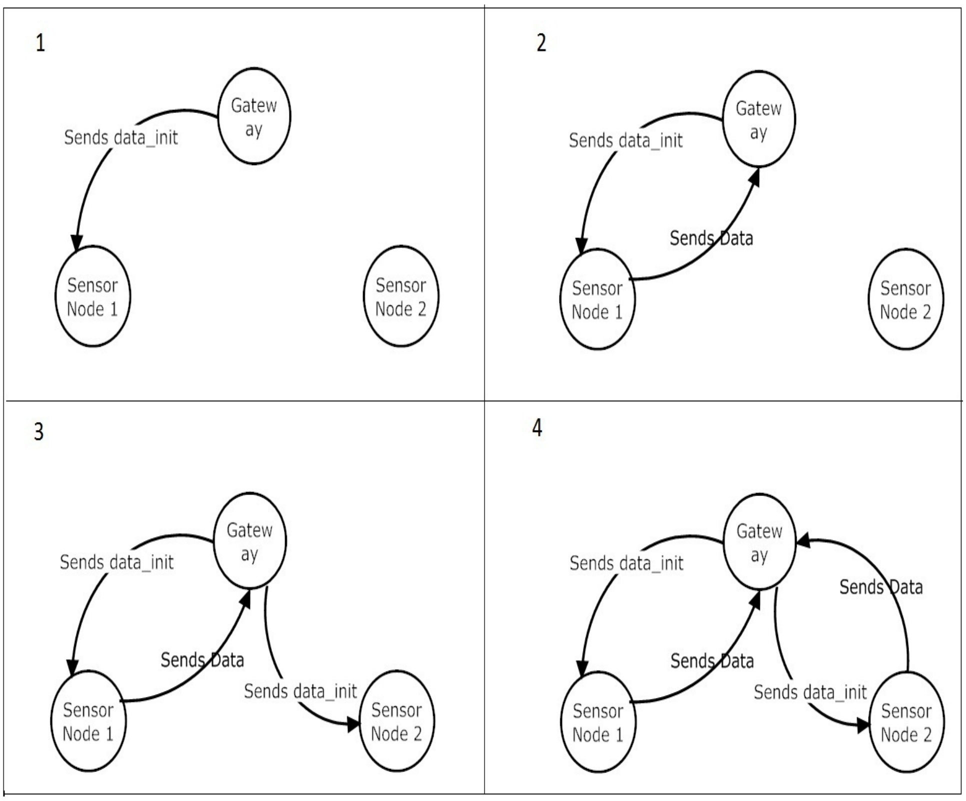

Figure 4.1: Overview of Time-Sharing Mechanism

Hence, the design is to make the Gateway send commands to each of the nodes in the network to inform the nodes on who to transmit at a particular time. For this purpose, we used a data_init command from the gateway that acts as the data initiation for the sensor node to which the command is sent.

Each of the node waits for its turn to receive the data_init command before attempting to transmit. This forbids the need for sensing the channel before transmitting and thus saves energy. The nodes can batch-up their sensor data while waiting for the data_init command from the Gateway and send the whole data in larger chunks when it has access to the medium. This can be explained as shown in the 4.1.

In the Figure 4.1, we have considered two sensor nodes in the network for illustration purpose. It is assumed that the two networks have been linked with the access point and is part of the network. The order of linking determines the order in which the data_init command will be sent and it will follow a round-robin fashion to encompass all the linked nodes. As seen in 1, the Gateway first sends the data_init command to the Sensor Node1 and only then in 2, the Sensor node sends its data to the Gateway. The same steps are repeated for the Sensor Node2 as shown in 3 and 4 in the 4.1 and then returns back to Sensor Node1. Thus at any point in time, there can be only one node accessing the channel and hence there is very minimal possibility of interference from the neighboring nodes. This time-sharing mechanism is shown in 4.2below.

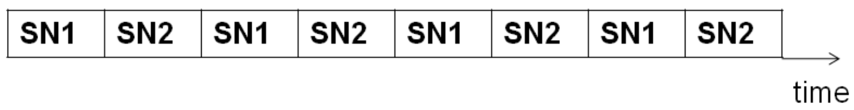

Figure 4.2: Time-Distribution of Sensor Node Transmission

Further, the design should take into account the changes in the network and should be able to adapt its schedule accordingly. When a new node enters the network, the schedule should be adapted to allocate a slot for the new node. These slot adaptation mechanism is done before every cycle of retrieving data from the nodes starts. In other words, a slot is reserved for various adaptations to be performed in the scheduling mechanisms depending on the network changes. This slot will be the first to be carried out in each of the round-robin cycle. Once the network changes have been incurred into the scheduling mechanism, then the cycle of sending data_init to each node and obtaining data from them will take place in the round-robin fashion.

It can be seen that the slot length is not fixed and hence each of the node can send varying amount of data to the gateway. This maximizes the bandwidth utilization because it assigns dynamic slot length to each of the sensor node. The sensor node will have the access to the channel depending on the amount of data it needs to send within the slot. This information is sent to the gateway in the first message from the Sensor Node to the Gateway. The Gateway waits and receives the messages as specified by the sensor node in its first packet. This allows for dynamic allocation of slots on-demand.

4.3 With Multiple Gateways

In the Section 4.2,we designed the protocol constituting one gateway and in this section, the manner in which data is collected in multiple gateways is discussed. Depending on the playfield size, the number of gateways increases. However, the sensor node should know to which gateway it should send the data. Hence, this algorithm decides to which gateway the data should be sent will reside on the Sensor-node. Further, as the data are sent only as a Uni-cast message, the process of duplication removal at the server side is not needed. This helps in minimizing the time for application development with this platform. The data collected by each of the gateway can be collected in the central server through an ethernet network. The advantages of using the wired network for this part of the transportation are listed in Section 3.3.

In the previous sections, it is only considered that the nodes are stationary and the network topology does not change. But whenever the player keeps moving away from the Gateway to which the sensor node sends data currently, the node should be handed over to the other gateway which is in the proximity to the current position of the Gateway. A criterion should be developed to make the sensor node know to which gateway it should send the data. Based on this criterion, the sensor data will make a hand-off to the next gateway which would satisfy the criterion. RSSI values provide a good estimate of the proximity of the sensor nodes and the gateways in the outdoor applications[18]. The data_init message which is sent to each of the sensor node can be used as good signal to retrieve the RSSI value from this packet. Since this packet is received by the sensor node in a continuous basis, the RSSI value and in turn the movement of the player gets updated frequently. This helps in tracking the movement and the proximity of the players/sensor nodes to a particular gateway. The sensor node maintains a set of RSSI values from each of the data_init packet it received. The sensor node then computes if it falls below a particular threshold value, it then makes the hand-off to the next gateway. This decision is not made with just one RSSI value falling below the threshold. Only when a series of RSSI values falls below threshold continuously, only then the hand-off is initiated. This will counter the dips in RSSI values caused by another player blocking the line-of-sight between the sensor node and the Gateway. Considering a player just crosses the line-of-sight, the RSSI value dips but however when the player has crossed it reflects the real proximity of the player with respect to the gateway.

4.4 Conclusion

This chapter details the protocol in which the data is collected from the various sensor nodes to onto the gateway(s). In this protocol, a time-sharing mechanism is used to access the common channel. The gateway sends a command to each of the sensor node and this command gives the sensor node the authentication to access the channel. Only one sensor node will talk with a particular gateway at any instant of time. To deal with multiple gateways, a hand-off mechanism based on RSSI value is developed. The next Chapter gives deeper picture of this protocol and also describes the implementation of the same.

Chapter 5

Platform Implementation

This Chapter gives details of the Implementation of the platform and also mentions some of the problems faced and their counter-measures taken. To start with, this Chapter will brief the development Environment in the Section 5.1. Section 5.2, will detail the steps taken for the Implementation of the platform in a stepwise manner.

5.1 Development Setup

This section will detail the development environment used in this project. As mentioned in Section 3.2, SimpliciTI comes with the full-source code and can be compiled with an IAR Embedded Workbench with 8051 toolchain for the target hardware CC2530. The SimpliciTI runs without any need for any under-lying operating system. Texas Instruments provide the SmartRF05EB, which is the evaluation and development platform for various Texas Instruments RF Evaluation Modules. For this project we use the CC2530 Evaluation Module for the development purpose, which could be mounted atop the SmartRF05EB, which provides the debugging interface for the module.

5.2 Wireless Protocol Implementation

In this section, the implementation of the Design discussed in Chapter 4, is described. In Section 5.2.1,the implementation of the communication between one transmitter and one receiver is explained. Further with that, Section 5.2.2 and Section 5.2.3, describes the implementation of the manner in which N transmitters can talk with One Gateway and N transmitters selectively transmit to 'm' Gateways respectively. Section 5.2.4 describes the manner in which dynamic slotting is executed. In Section 5.2.5 briefs the advantages in sending too multiple packets within one slot and the way it is implemented in this protocol.

5.2.1 One-to-One Communication

To understand the implementation of the communication set-up, first an overview of the Layers of the SimpliciTI are shown in the Figure 5.1.

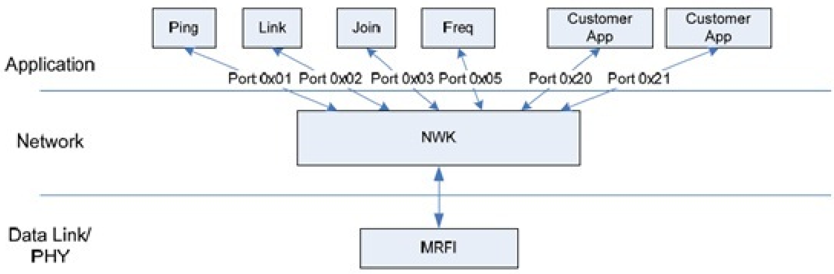

Figure 5.1: SimpliciTI Logical Layers

As can be seen from the Fig 5.1,there is no formal PHY/DataLink Layer and the MRFI layer encompasses the functionalities of both these layers. The NWK layer manages the Rx and Tx queues and dispatches frame to their destination. The destinations are identified by the Physical address appended with the Port number, which determines the functionality. The specific application will reside on the Port, which can read the frame destined to it for further processing. To further provide security against any rogue device, each of the devices in the network should be provided with a "Join" Token during build-time. This will be used as the initial security-level to gain access to a network controlled by the Gateway, which is a 4-byte identifier. As a side effect of the Initialization in the Sensor Node, it tries to find whether there are any Gateways in the proximity and joins the network if available. With a successful Join, the Gateway then provides the network infrastructure information like the Link Token, encryption key in the payload of its reply message. The Link Token is responsible for creating a peer-to-peer link between any two nodes, which is again of length 4-bytes. These links can naturally be set-up only in pairs with one of them *listening* for the link and the other *linking* to the listening node. However, the link token will not be used in every transaction between the peers. A local Link Handle will be created in both the peers, which will be used to identify the connection for later usage. Every time the Local-Link handle is accessed, it is matched with the physical address of the device set-up earlier during the linking process.

SimpliciTI does not provide any interface with the stack and hence before setting up the wireless set-up the interfaces for the sensors are implemented. The ADC interface is implemented with the stack to provide the possibility to attach analog sensors like Accelerometers. The ADC interface can read from any of the 8 channels available as the analog ports in the CC2530EM and also with a variable resolution of 8bits, 10bits and 12bits. The UART Interface is also implemented with the stack which can be used to program any of the 2 USART modules available in the CC2530. The UART is used with the baud-rate of 115200 and is used in the Gateways to log the data onto the PC.

As a first step in implementing the Wireless Network for the Platform, the basic communication module with one sensor node and one gateway is implemented. This will form the basis for the entire network set-up described in the further sections, with necessary modifications. With two device configurations, an Access Point and End-Device, it will be discussed in this section how a node when switched ON, be identified in the network and incorporated into the network. This takes place in two steps as following:

1. Joining the Network controlled by the Gateway.

2. Peer-to-Peer Linking with the Gateway.

The Gateway and the Sensor Node are specified with the Network-specific join token during the build-time. The Join Token specified in the Sensor Node should match the one specified in the Gateway during the build-time to get access into the network initially. This is used to avoid any rogue devices or unintended nodes to get into the network. During the Initialization of the stack in the Sensor Node, it sends out a broadcast message with the Join Token specified. When the gateway hears the Join Request message, it first matches the Join Token in the payload of the received packet with its own Join Token. If they both match, then the Gateway identifies that this node is the one to be the part of its own network and hence sends a "Join Reply" message to the node which sent the Join Request. The Join Reply message will contain the network infrastructure information like the Link Token of the Network which would facilitate the sensor node to create a peer-to-peer link with the Gateway. After receiving the *Join Reply*, the sensor node understands that its Join Request has been successful and hence sends the *Link Request* message with the Link Token that it received from the successfully joined Gateway. The gateway hears the message and then sends back a *Link Successful* message to the node. Only at this point, a peer-to-peer connection is set-up between the gateway and the sensor node. The Link will have two different handles, called the Link IDs on both the sides of it to enable access to the devices on the other sides. The Link IDs are matched with the physical address of the device appended with the port number mentioned during the linking process. The *Request* messages are broadcasted continuously after a specified amount of time, until they get a reply from the Gateway.

To send a message to the Gateway from the sensor node, the application can send the data specifying the LinkID of the gateway, while the NWK layer matches the mentioned LinkID with the corresponding physical device address and the port number. Similarly, on the other side, the gateway radio receives the packet and the NWK layer matches the physical address specified in the packet with its table of LinkIDs. A callback function is implemented on the listener side to read the message available in the LinkID specified and this call back function runs in an ISR thread. To efficiently use the ISR, only a flag is set in the callback function indicating the availability of message in the LinkID and the message is read out actually in the application.

5.2.2 N-to-One Communication

Having briefed on the way in which a single link is established, in this section, a Star network with 'N' sensor nodes and one receiver is implemented. In this section of the implementation, the design part discussed in Section4.2, will be implemented to access the shared channel in a time-slotted manner. Once the communication with One Access Point and One End device is set-up, the implementation of N-to-One communication, which is the Star Network, with converge-cast data is set up. There could be many ways in which this can be set up like completely random listening to End devices by the Access Point, which has an inherent dis-advantage of collisions or data-loss. However as discussed in 4, an arbitration model is setup for the Gateway to listen to one node at a time and importantly only that single node will be transmitting in the network at that time. This will reduce the possibility of data collision in the network. The following section describes the steps in which the communication is setup between the Gateway and the N Sensor nodes and the arbitration model used based on which the sensor nodes can transmit data without any possible collision from other nodes.

The communication set-up between the Access Point and each of the End-Device follows the similar procedure as explained in the section above. However, they are constrained in a manner not to interfere with the arbitration scheduling. When the Gateway is switched ON, it keeps looking for the End Devices, through a non-blocking call. When a new End Device joins, a Join flag is raised in the Gateway. If there are nodes in the network, the Gateway has nothing to process except to keep looking for the first node. Once, it finds the first node the peer-to-peer communication is set-up between the Gateway and the node as mentioned in the Section 5.2.1.
Once the first node is incorporated into the network, the Gateway sends a

Unicast message to the newly joined node with a pre-determined 3-byte payload, which is the data_init command. The sensor node receives the message and decodes it to understand the message as the data_init command from the gateway. Since it has received the data_init command from the gateway, it has the authentication to send the data to the gateway. It sends the data it has collected to the gateway and does not expect any acknowledgment from the gateway. Once it has received the data from the sensor node, the gateway processes the data and sends it to the PC through the serial line.

Before sending out the data_init command to the sensor node again, the gateway first checks if there are any flags raised, indicating a new sensor node waiting for connection. If yes, then the gateway first serves the new connection as described in section 5.2.1. After incorporating the nodes into the network, the gateway then starts sending the data_init command to each of the nodes in the network to get the data from these nodes. The data_init command is sent to each of the nodes in the order they entered the network and in the round-robin manner.

5.2.3 N-to-m Communication

To cover the Entire playfield, Multiple Gateways are to be employed along the sides of the playfield to collect the data from the End Devices and then send it across to a central server. The nodes will transmit only to one Gateway i.e., it will be a Uni-cast message from the nodes to the Gateway. However, when the node moves away from the comfortable range of the Gateway, the node will hand-off to a newer Gateway which has a better proximity than the previous Gateway. The Hand-Off is decided on the basis of the RSSI values of the data_init received from the Gateway. The following section explains the procedure how the hand-offs from one gateway to another is implemented.
As discussed in the Chapter 4, RSSI provides a measure of proximity well in the outdoors relatively. Hence, the RSSI value will be used as the criterion to decide whether to make the Hand-off or not. The sensor nodes are not broadcasting their data but will be just sending the message only to the specific gateway. This allows for skipping the duplicate removal on the central server side and also there is no need for any synchronization among the gateways deployed in the play-field. The sensor-nodes receive the data_init message continuously from the Gateway, whenever it is a part of the network. Hence, this can be used as the base to calculate the RSSI criterion.

However, deciding based on just a single sample of the RSSI was not reliable with the proximity of the players to the controlling gateway. When a player just crosses the line of sight between the gateway and the sensor node, there was a dip in the RSSI value measured. This resulted in erroneous decisions made with regard to the Hand-Offs. This made the sensor-node to look for another gateway, though it was well close to the current gateway. Hence, to avoid such conditions, a set of RSSI values are used to make the hand-off decision. Every time the data_int packet arrives at the sensor-node, the RSSI value is retrieved directly from the last received signal, after it becomes valid. By this method, it provides a much reliable RSSI value than the instantaneous one which is very much affected by the noise. This RSSI value is checked against the threshold value every time, the packet arrives. The threshold value is varied according to the play-field size and the number of gateways deployed. If the RSSI falls below the threshold value "continuously" for 20 times, then the hand-off is initiated. The main aim here was to minimize un-necessary hand-offs as much as possible. Hence, the decision is made on the basis of continuous monitoring of the RSSI values, rather a single or average value. Thus, the surge and the dips in the RSSI value caused erroneously or by a player blocking briefly does not affect the decision on the hand-off.

While changing the channel to make the hand-off, the sensor-node needs to know the *Join Token* of the new gateway or the network. By having different Join Tokens for each network, it is possible to have 2^{32} different gateways to provide the coverage over a vast area. This would make the platform possible to be deployed for any sports, including track events which need larger coverage compared with the field-sports. To change the Join Token of the Sensor node at run-time, IOCTL interface calls are implemented which change the Join Token specified in the NWK layer of the Sensor nodes. For this purpose, the sensor-nodes maintain a list of *Join Tokens* of the corresponding gateways available in the play-field.

5.2.4 ReUsing Inactive Slots

With such mobile nodes in the network making hand-offs to other networks, it was noticed that many slots are left vacant and un-used in due course of time. This in turn increased the latency or the response time of the other sensor nodes in the network. In most of the field-sports, substitutions of players take place during game. When a player is removed, the slot would be vacant and hence needs to be re-scheduled to reduce the latency. In both the cases, there would not be any explicit communication from the sensor node. Hence, the gateway has to sense the node to be dead if it does not receive any data from the sensor node after sending the data_init successively for 10 times. If the sensor-node does not have any sensor data to send, it will transmit at-least it's Node-ID. Hence, if the node does not send any data to the gateway for the specified runs, the Gateway unlinks the node and puts the slot-id in a queue maintained for un-used slots. The Gateway will not send the data_init command to these nodes in the next cycle and thus reduces the latency by un-necessarily waiting for data from a removed node. These slots will be re-used by the newly joining nodes or the newly handed-off nodes from other gateways. Figure 5.2 and Figure 5.3 respectively show the state chart of the protocol implemented in the Gateway and the Sensor Nodes. The text in colours show the interaction between the Gateway and the Sensor Nodes among each other.

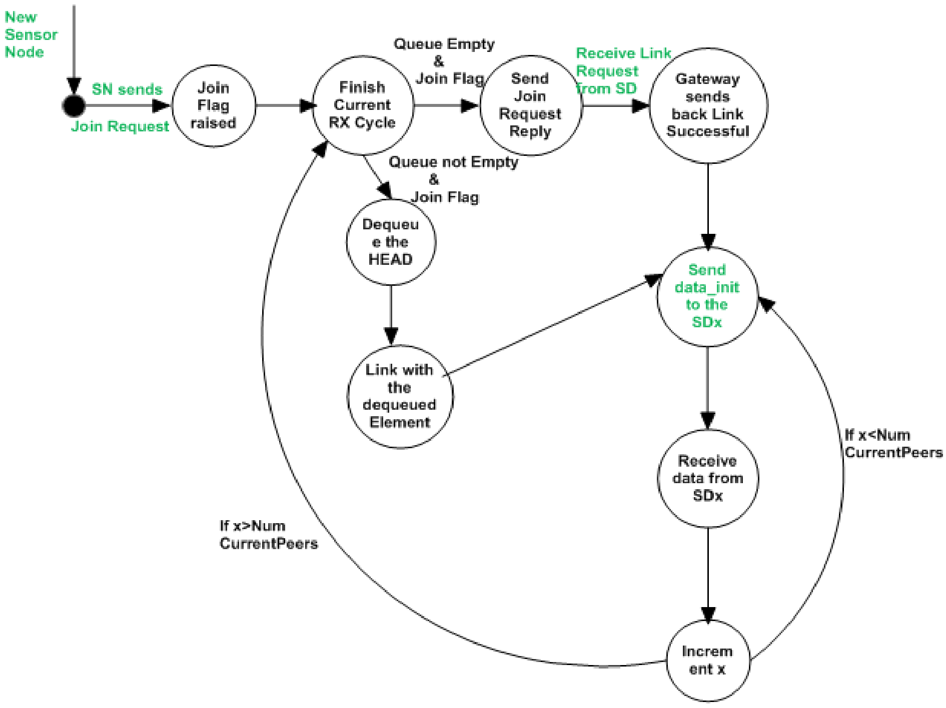

Figure 5.2: Implementation State Chart of the Gateway

5.2.5 Multiple Packets with Time Compaction

With increased nodes in the network, the time each of the sensor-nodes receives the data_init proportionally increases. As a result, the amount of sensor data collected before the sensor-node gets its slot also increases tremendously. To counter this, the sensor data are packed in packets of size 100 bytes each, limited by the TX/RX FIFO Queues of the radio. When the sensor-node gets its turn to send the data, it occupies the channel until it sends all the data in the form of packets each of 100 bytes. This has been found to increase the effective throughput and is discussed in Chapter 6.

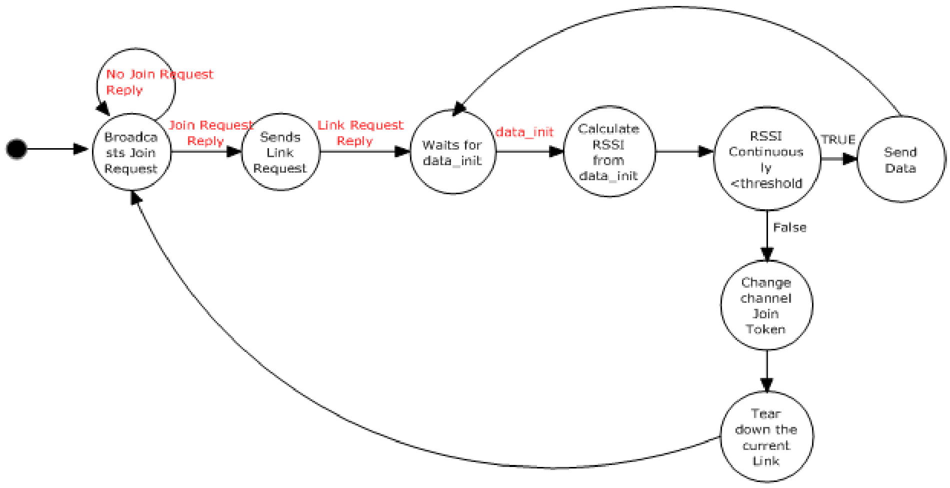

Figure 5.3: Implementation State Chart of the Sensor Nodes

5.3 Conclusion

In this Chapter, the implementation details of the protocol designed in Chapter 4 is discussed. Starting with the communication set-up between the gateway and each of the sensor nodes in the time-shared mechanism proposed, different adaptations are under-taken to solve the problems that are encoutered while implementing the protocol. Chapter 6 details the various experiments done in line with the implementations performed in this Chapter.

Chapter 6

Validation and Performance Test Results

With the software for the Platform being developed, the next step is to test the Software to check its robustness to various parameters. This chapter gives the procedure how extensive the Software is tested and also the various aspects covered in these tests. These tests are conducted to analyze the reliability and performance of the wireless communication system in a long run. The tests were conducted systematically with a pre-defined set of procedures and the results presented wherever possible.

6.1 Validation Tests

As a first step in the Testing, all the functionalities of the software are tested intentionally to analyze the reliability of the software. The functionalities of the software includes the adding of new nodes in the network when various other nodes are already part of the network and transmitting data, removing of nodes from the network, making hand-offs form one network to another and back. The main aim of this test is to verify the working of the various functionalities of the software. To analyze the performance of the wireless system, further experiments are done by varying different parameters and analyzing its results in the Section 6.2.

A test setup is first created for performing this validation test on the developed software. The Sensor nodes are programmed to generate sequence numbers and the size of the packet sent from the sensor node to the gateway is programmed to be a fixed 100 bytes for test purpose. The contents of the packet will be the replica of the same sequence number. In other words, each of the packets will have the sequence number repeated. The packet sent from the sensor node to the gateway will have the payload start with a Node-ID and the number of packets that will be sent in the slot. The remaining 98 bytes are the sequence number repeated in it. When the sensor node is switched ON, they join the network and start transmitting this data. The sequence number will be repeating from 1 to 255 and after 255 it will wrap to 1. The number 0 is reserved for a special case in the Gateway. The sensor nodes transmit the data for a random amount of time and after the random time, they either remove itself from the network and joins back later or will be handed-over to the other network.

There are two networks formed with two gateways; with each gateway having a set of sensor nodes which join and leave the network randomly. The gateway receives the packets from the sensor nodes and processes the message and logs it in a text file in the central server. The gateway checks the contents of the packet received and retrieves the information on which sensor node the packet is received from the first byte of the packet and this NodeID is sent to the PC. The second byte will tell the gateway for how many packets it should listen to and the gateway listens accordingly. The remaining bytes are the actual contents of the message and the gateway checks all the 98 bytes of the message and if they are of the same value it sends the value to the PC. If any of the bytes in the message does not match, then the packet is corrupted and the Gateway will send a '0' to the PC, which indicates that the packet has an error.

There are specific messages to announce to the Gateway of various changes in the network like a sensor node joining the network or being handed-over to it. These messages will be differentiated from the other messages by the first byte of the packet, since in the other messages the first byte will be the NodeID, which does not exceed the limit. Hence, the unique messages which are logged in the file details the activity, that had underwent during the test. The tests were conducted indoors and three of the nodes were kept in the RSSI overlapping region of the network. Hence, these nodes can make hand-offs from one network during the test and also will join back and forth between the networks. This test-setup is used to check the hand-off functionality of the sensor software for the mobile players.The Software passed these setups and the results are shown in Table 6.1

Packet Length	Num Nodes	Num Packets	Packet Loss%
100	13	1	0.0612
100	13	3	1.1481

Table 6.1: Validation Test Results

6.2 Performance Testing

In the previous set of experiments, the main aim of the tests was laid on the validation of the functionalities that the software provided. But in this section of the experiments, the performance of the wireless system is analyzed through a set of tests. There are various parameters that can be varied to analyze the performance of platform software and a trade-off on these factors should be made while deploying the platform in the playfield. The factor with which the performance tests are based is the packet error rate or the percentage of the packets lost or received with incorrect payloads. This value of the packet loss percentage provides a good estimate on the performance of the wireless communication between the sensor nodes and the gateways. Having logged the received packets in the PC, an application is run on the PC side to calculate the missed packets and also the packet errors. The packet losses are calculated with the help of the sequence number and the packet error is calculated by the value 'zero' which is sent to the PC from the gateway if it finds that the contents of the packets are not as expected and are corrupted. If a new node has joined the network it will send an *announcement* message to the gateway which is also logged into the PC. Hence, the application which calculates the packet losses can understand from these messages that a new node has joined the network and will start calculating the difference in the sequence number from there on for that particular node, mentioned by its NodeID. Further, the losses are accumulated for the different nodes in the network and the loss percentage is calculated from the total packets expected.

These tests are conducted with the Platform hardware with internally generated data as mentioned above, from the CC2530 modules. The gateway is connected to the PC through the UART and the PC logs the file for further analysis. The tests are automated as much as possible and the log file is processed after the test is completed. These tests are conducted in line with the requirement and the results are then analyzed based on the trade-offs that could be made in the system.

Considering the design of the platform, mentioned in Chapter 4, there are various parameters that can be varied according to the application. These parameters are varied manually in the test cases to infer the behavior of the system to changing these parameters. Some of the parameters that could be changed are the size of the packets sent, number of packets sent in each slot and number of nodes in the network at any instant of time. Increasing the size of the packet and/or increasing the number of packets sent in each slot increases the effective data-throughput. However, it should be noted how much extra time it requires to send the extra packets through the obtained slot. This will in-turn affect the amount of time the particular node occupies the channel. Hence if a single node sends two packets in the slot with a time occupying the channel equal to two nodes sending one packet each in two different slots, then there will be no increase in the effective net data-rate on the side of the central server. Hence, the amount of time needed to send multiple packets should not be increased linearly but at the same time the node should be provided with sufficient slot-length to allow for successful transmission of the multiple packets and successful reception at the gateway side. It should be noted that acknowledgement is not sent for the packets and hence sufficient time should be provided to the nodes to send the data and for the gateways to receive the data without any possible over-riding.

The number of nodes in the network is also varied in the experiments to infer how the switching between the nodes affects the system and also how the entire system scales in terms of number of nodes in the network. The number of nodes in the network is increased and the corresponding packet losses are also calculated for each of the run. The effective data-rate is calculated for the whole network on the gateway side. i.e., the effective data-rate is calculated as the amount of data received by the gateway with respect to time from all the sensor nodes in the network for this run. Though the data-rate increases with increase in the number of sensor nodes in the network, the latency or the response time of the sensor node increases. This is mainly because with increase in the sensor nodes, the gateway has to listen to each of the nodes in the network in a round-robin fashion before servicing again the other nodes. Hence, in the following sections each of these scenarios are tested with real-life experiments and analyzed with the results.

6.2.1 Effect of increasing number of nodes

In this part of the testing, the number of nodes in the network is increased in steps of two for each of the run of the experiment. The packet error/loss will be calculated for each of the run as mentioned above. The tests are conducted in indoors with a spacing of 5-8 meters between different sensor nodes and the Gateway. One gateway is employed in these tests to check the potential of a single network. The size of the data payload in the packet sent is fixed at 100 bytes. However, the number of packets which are sent in every slot is varied in steps of one until a maximum of 5 packets is reached. In each case, the packet error and packet losses are calculated and the results are analyzed with these results reflecting on the response of the performance of the system.

The test software used for this part of the testing will e run for a fixed set of slots for each of the sensor node. For the test purposes, the gateway will send how many packets the sensor nodes need to send in each slot and will get the packets with a fixed packet size of 10 bytes. After a fixed number of cycles of receiving data from the sensor nodes, the gateway increases the number of packets sent in each slot by a factor of one. This information is sent to the sensor nodes, along with the data_init only for the testing purposes. The sensor nodes then send data accordingly in the mentioned packets and the whole process takes place until the number of packets gets to five packets in each slot. The data are received by the gateway and processed to find any errors in the contents of the packet and then is sent accordingly to the PC and logged in a file. The packet error/loss percentage is calculated by running the respective application with the log file and finding the missing packets or the packets with value '0' which represents the error in the contents of the packet.

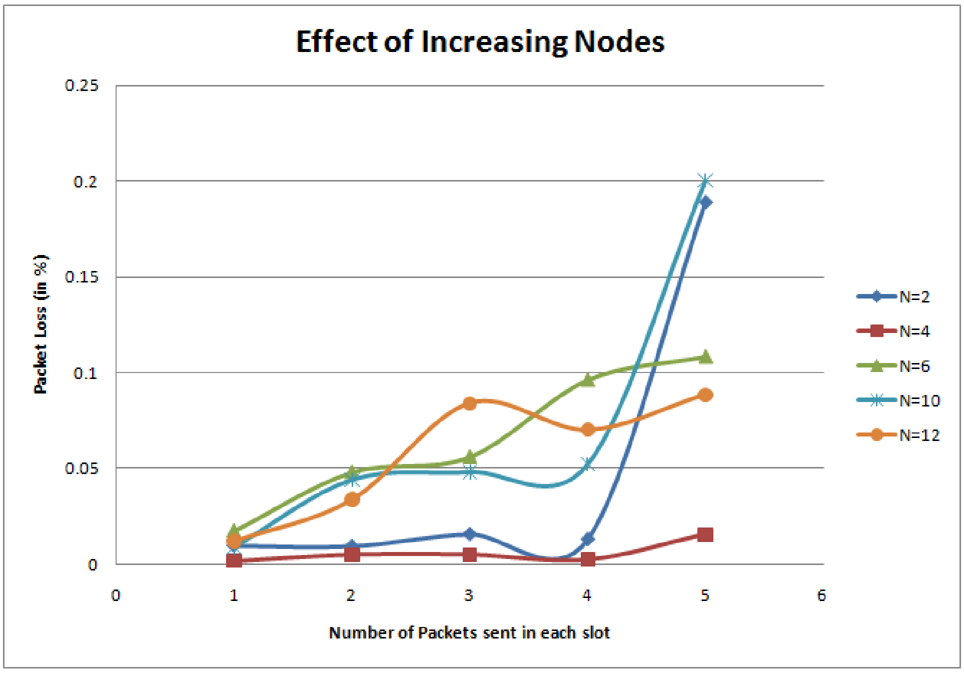

Figure 6.1: Effect of Increasing Nodes

The Fig. 6.1,shows the variation in the Packet loss percentage against the number of packets sent in every slot for various number of nodes in the network. In the X-axis there is the number of packets sent in every slot and the Y-axis has the packet loss percentage. For various number of nodes in the network, a plot is made in this graph. It can be seen from the graph that the packet loss percentage is less than 1% in average for all these cases. It clearly shows that there is no effect of increasing the number of nodes in the packet losses. The major reason for collision in the context of the wireless sensor networks can be attributed to the collisions caused with the neighboring nodes trying to access the channel to transmit to the gateway. But in the case of this protocol, only one node can transmit at any point of time. Only after the data_init command from the gateway is received by the sensor node, it will access the channel to transmit its data. Hence, at any point of time there cannot be ideally more than one sensor node transmitting data. This completely minimizes the chance of losing data packets due to collision with the packets from the near-by nodes.

Though every time only one peer-to-peer connection ideally exist, there can be chances of nodes trying to join the network and hence start transmitting the packets to the gateway and may cause collision. And also, increasing the number of packets in the slot may cause over-riding of packets in the RX FIFO of the gateway module. However to counter this, the time-slot is extended to provide for successful transmission of the multiple packets.

6.2.2 Effect of Outdoor Deployment

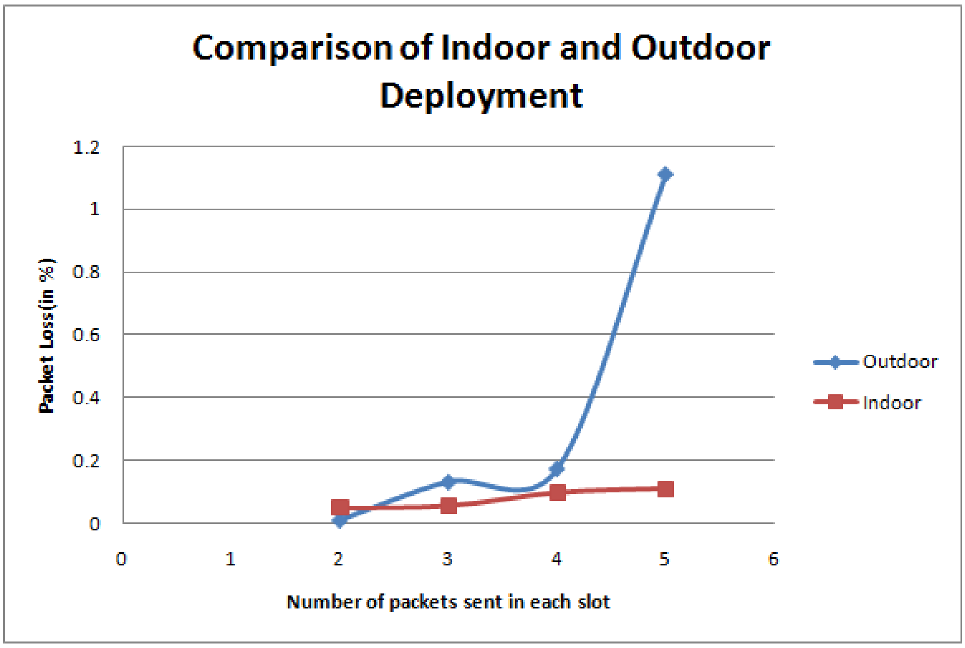

Figure 6.2: Effect of Outdoor Deployment

With the above results, it was seen that increasing the number of nodes has less impact on the packet loss percentage. The next step was to deploy the same in outdoors and see how it varies with the increasing number of packets sent in every slot. In this test scenario, six nodes were deployed first in complete outdoor space with almost no possibility of reflections. The ground was a grass surface and hence will also provide very less reflection.It was carried out during a period of day with large number of people moving between the nodes. It was to provide a close scenario of a playfield. All the sensor nodes are placed in random locations within a radius of 30 meters from the gateway. Similar procedure is followed to calculate the packet loss percentage and the following graph is plotted with the result and is compared against the result we obtained from the Indoor calculation.

From the Figure 6.2, we again see that the packet loss percentage is kept as low as 1% for these different cases. This clearly holds well with the theoretical expectation of such a design of the protocol, in which only a pair of nodes communicate with each other at any point of time.

6.2.3 Effect of failed Nodes in the Network

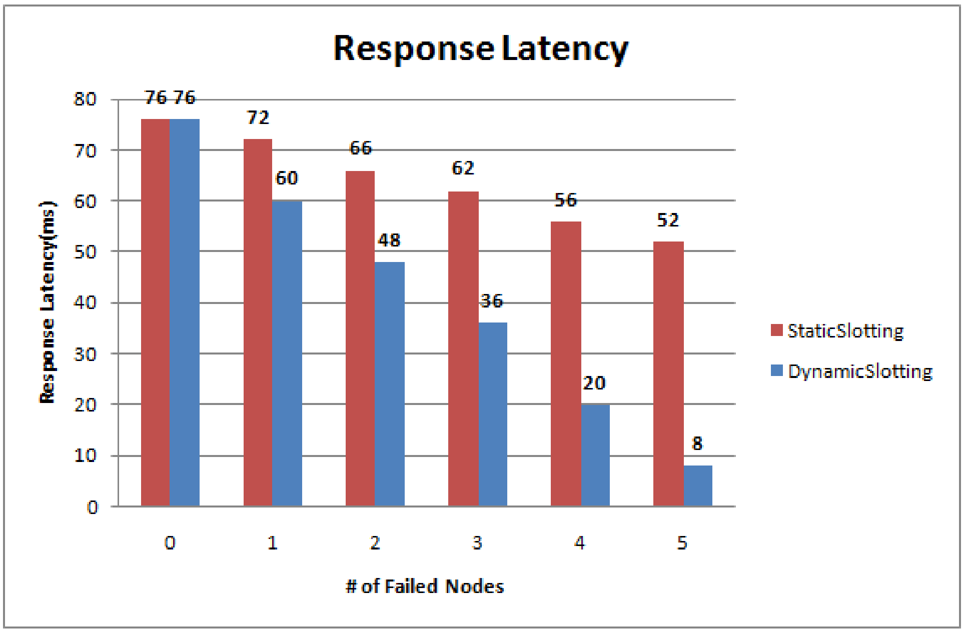

Figure 6.3: Effect of Response Latency with Failed Nodes

In the round-robin schedule of collecting data from the sensor-nodes by sending the data_init command, when one or more of the nodes fail in the network, the gateway has to wait and listen unnecessarily for that particular sensor node. However, when a node is dead or removed from the network, there will be no explicit transmission from the sensor node to the Gateway. However, as incorporated in the protocol the gateway infers that the sensor node is dead if it does not receive data from the sensor node continuously for a particular number of its slots. In the graph below Fig 6.3, the latency or the response time caused due to the failed nodes are analyzed. But in this protocol, the dynamic allocation of slots is employed which reduces the latency or the response time of the other sensor nodes significantly.

By reducing the latency due to the failed nodes, the channel utilization is also increased. Instead of waiting for the dead nodes, those slots are allocated to active nodes which use the channel efficiently, thereby increasing the channel utilization. This will also naturally have an effect on the throughput on the gateway side, because the channel is packed with data continuously without any idle periods due to the dead nodes.

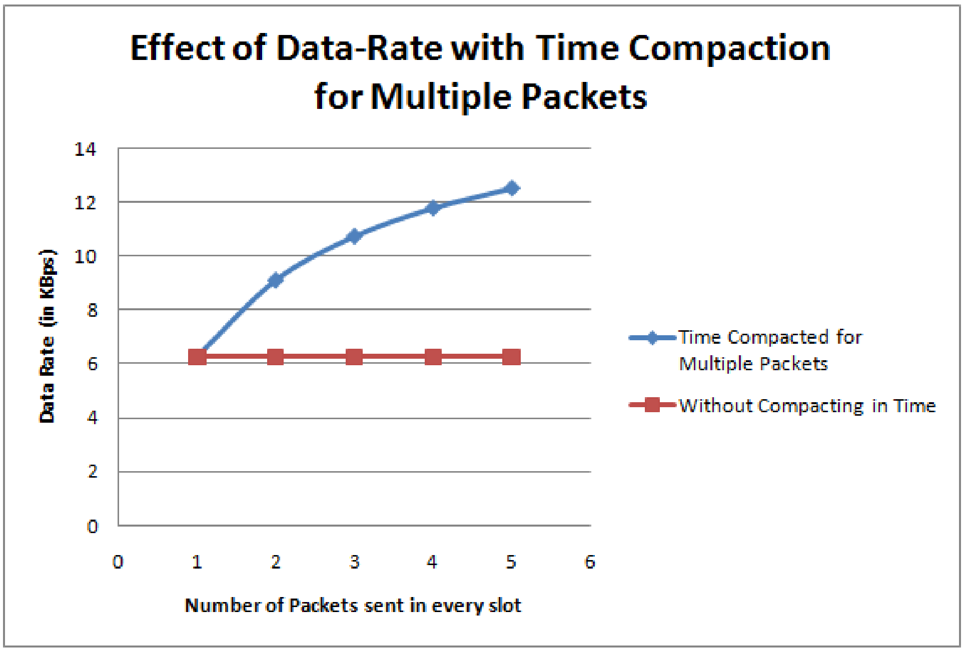

Figure 6.4: Data-Rate with Multiple Packets

6.2.4 Effect of Time Compaction for Multiple Packets

With this application being critical on data throughput, there are possibilities that there needs to be packets larger than 100 bytes to be sent in a particular slot. However, due to the limitation on the Tx/Rx FIFO of the radio module, the data are sent in multiple packets as discussed in 5.2.5. However, the slot-length for one packet is determined as 16ms and instead of extending the slot-length for two packets or sending two packets in two slots which would take up-to 32ms, the slot length is kept as 22ms. This slot-length of 22ms for two packets is compacted in a manner just to include the over-the-air time of the packet. Thus by using the time slots more efficiently, the net data-rate on the gateway is increased significantly as compared to the manner in which the Time-Compaction is not performed.

6.3 Conclusion

In this Chapter the wireless platform is analyzed through various experimental results. In the first section, the various validation tests are performed to verify the functionality of the software and they had passed the tests in the long run. In the second section, different parameters are varied in accordance to the requirements of the Application and is analyzed how the system reacts to those variations.

Chapter 7

Conclusions and Future Work

A Wireless Sensor Platform is developed for rapid-prototyping mainly targetted at sports applications. From a varied options, Texas Instruments CC2530 is chosen as the platform hardware for the development of the wearable sensors for the Players. A network topology is designed with gateway(s) along the sides of the playfield, by which it is scalable for play-fields of any size. A wireless protocol is designed and developed with a time-sharing mechanism to aggregate the sensor data in real-time in a central server and its perfomance is analyzed.

7.1 Conclusions

In line with the Section 1.2,the following have been achieved,

- A wireless sensor platform is developed to mainly target rapid-prototyping at sports applications, with a chosen hardware module CC2530 from Texas Instruments.

- The Platform can support atleast one sensor streaming raw data continuously at high data-rate like a 3-axis accelermeter streaming data at 200Hz.

- Any off-the-shelf sensors with Analog, SPI or UART interface can be hooked up with the platform to develop the sensor nodes for different sports.

- The sensor data from all the nodes can be aggregated and analyzed/displayed at a single location or a central server.

- A Network topology is proposed for these sporting applications which could encompass the entire play-field with multiple gateways connected through ethernet along the dimensions of the playfield.

- A wireless radio protocol is proposed and implemented which uses time-slotted mechanism to grant access to the channel. The developed protocol is a centralized algorithm with gateway having complete authority as to which sensor ndoe should transmit at a particular instant of time.

- With the help of various experiments, the wireless setup is tested with varying number of nodes and found to provide efficient delivery with just around 1% packet error.This strongly proves that whatever be the number of nodes in the network only one transmit at a time to the gateway which is the base of this time-sharing mechanism.

- With mobile nodes handing-off between multiple gateways, it was observed that slots are being wasted in a long run corresponding to a particular gateway which also had unwanted impact on the response time or latency of the system. To mitigate this, dynamic slotting was employed by which the gateway re-assigns the inactive slots or skips them.

- To attain high data-rates, multiple packets are sent within a time-compacted slot in which the slot-length is increased linearly with the packets sent and increased just enough to send the extra packet.

- The Platform is capable of supporting adding or removing of players during the game which is a characteristic of most field-sports in the form of Substitution and it takes place without hampering the functioning of other nodes.

7.2 Future Work

Though with this platform a range of applications can now be developed customized for different sports, there is still space for improvement in the platform inherently. Some of the future works on this platform are listed below:

- Currently, there are no acknowledgements sent from the gateway to the sensor nodes for the successful reception of the sensor data. However, to reduce the packet losses further closer to 0%, a block acknowledgement mechanism can be developed to further enable re-transmission of lost packets.

- With this centralized algorithm, the sensor nodes are in listening mode waiting for the command from the gateway to send the data. An algorithm can be developed to better estimate the time when the nodes may receive the command from the gateway. The nodes can wake up during that window of time where it can expect the command from the gateway.This will allow the nodes to enter the sleeping mode without missing any commands from the gateway.

Bibliography

[1] John A. Stankovic. Wireless sensor networks. Technical report, University of Virginia, 2006.

[2] D.Pfisterer, M.Lipphardt, C.Buschmann, H.Hellbrueck, S.Fischer, and J.H.Sauselin. Marathonnet: Adding value to large scale sport events - a connectivity analysis. In *Proceedings of the First International Conference on Integrated Internet Ad hoc and Sensor Networks*, 1-59593-427-8/06/05, Nice, France, May 2006.

[3] A.F. Smeaton, D.Diamond, P.Kelly, et al. Aggregating multiple body sensors for analysis in sports. In *International Workshop on Wearable, Micro and Nano Technologies for the Personalised Health*, Valencia, Spain, June 2008.

[4] P.Wilhelm, E.Monier, P.Thomas, and U.Rckert. SPA a system for analysis of indoor team sports using video tracking and wireless sensor network. In *Proceedings of 6th International Symposium on Image and Signal Processing and Analysis, 2009*, pages 237 – 241, Salzburg, Sept 2009.

[5] J.Degesys, I.Rose, A.Patel, and R.Nagpal. DESYNC: Self-Organizing desynchronization and TDMA on wireless sensor networks. In *6th International Symposium on Information Processing in Sensor Networks*, page 11, April 2007.

[6] G.P. Halkes and K.G. Langendoen. Crankshaft: An energy-efficient MAC-protocol for dense wireless sensor networks. In *European Conference on Wireless Sensor Networks*, page 228, 2007.

[7] W.R.Heinzelman, A.Chandrakasan, and H.Balakrishnan. Energy-efficient communication protocol forwireless microsensor networks. In *Proceedings of the 33rd Hawaii International Conference on System Sciences*, page 10, January 2000.

[8] M.Lehsaini, H.Guyennet, and M.Feham. CES: Cluster-based energy-efficient scheme for mobile wireless sensor networks. In *International Federation for Information Processing*, volume 264, page 13 to 24, 2008.

[9] Y.Liu, Y.Zhao, and J.Gao. A new clustering mechanism based on LEACH protocol. In *International Joint Conference on Artificial Intelligence*, 2009.

[10] S.H.Manjula, B.Reddy, Shaila K, L.Nalini, K.R.Venugopal, and L.M.Patnaik. Base-station controlled clustering scheme in wireless sensor networks. In *Wireless Days*, 2008.

[11] M.Arif, L.M.Kamarudin, A.Y.Md Shakaff, M.N.Ahmad, and R.B.Ahmad. Outdoor propagation analysis for lr-wpan ieee 802.15.4. In *Second International Conference on Network Applications, Protocols and Services*, 2010.

[12] Apple+nike sports. `http://www.apple.com/ipod/nike/`.

[13] Evaluation of various radio technologies. Philips Research,Eindhoven.

[14] Simpliciti. `http://focus.ti.com/lit/ml/swru130b/swru130b.pdf`.

[15] Z-stack. `http://focus.ti.com/docs/toolsw/folders/print/z-stack.html`.

[16] I. Ramachandran and S. Roy. Wlc46-2: On the impact of clear channel assessment on mac performance. In *Global Telecommunications Conference, 2006. GLOBECOM '06. IEEE*, pages 1 –5, 27 2006-dec. 1 2006.

[17] V. Cionca, T. Newe, and V. Dadarlat. TDMA Protocol Requirements for Wireless Sensor Networks. In *Sensor Technologies and Applications, 2008. SENSORCOMM '08. Second International Conference on*, pages 30 –35, aug. 2008.

[18] Kannan Srinivasan and Philip Levis. Rssi is under appreciated. In *In Proceedings of the Third Workshop on Embedded Networked Sensors (EmNets*, 2006.