Universidad Autónoma de Baja California

Facultad de Ingeniería

Maestría y Doctorado en Ciencias e Ingeniería

Puente WiFi-WSN

Tesis que para obtener el grado de

Maestro en Ingeniería

Presenta

Ignacio Eduardo Lerma González

Director de la Tesis: Dr. Juan Ivan Nieto Hipólito.

Ensenada Baja California

Agosto de 2010

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA

FACULTAD DE INGENIERÍA
UNIDAD ENSENADA

Puente WiFi-WSN

TESIS

Que para obtener el grado de maestría en ingeniería presenta:

Ignacio Eduardo Lerma González

Aprobada por:

Dr. Juan Ivan Nieto Hipólito
Director de tesis

Dra. Mabel Vázquez Briseño
Miembro del comité

Dr. Juan de Dios Sánchez López
Miembro del comité

M.C. Elitania Jiménez García
Miembro del comité

M.C. José Antonio Michel Macarty
Miembro del comité

Ensenada Baja California, México. Noviembre 2010

R e s u m e n

Sistemas para el cuidado de la salud han recibido en los últimos años un especial interés principalmente en países desarrollados como lo son E.E.U.U. y países europeos. El gobierno de E.E.U.U. gasta más de $2 trillones de dólares en el cuidado de la salud anualmente. Un punto clave que toma en cuenta para el manejo de estos costos es la computarización o modernización del sector médico. Durante la administración del presidente Obama se tiene planeado automatizar los expedientes de salud de Norteamérica en solo 5 años [1].

Una de las áreas que ha cobrado auge en el desarrollo de estos sistemas son las redes de sensores inalámbricas *(WSNs)*. Se buscan alternativas de bajo costo que permitan incrementar el uso de este tipo de redes aprovechando sus múltiples ventajas. Uno de los puntos clave es la unión de ésta tecnología con tecnologías más robustas como lo es *WiFi*, de esta manera se pueden diversificar aplicaciones y aprovechar las bondades en el enlace de estas tecnologías.

A continuación se presentan la motivación para el desarrollo de este proyecto, luego los estándares estudiados (IEEE 802.11 e IEEE 802.15.4), posteriormente un panorama completo del desarrollo de puentes y finalmente el desarrollo del presente proyecto: un dispositivo puente prototipo que comunique dos protocolos: el estándar IEEE 802.11 y el IEEE 802.15.4 *(WiFi y WSN)*, incluyendo en este último punto el hardware y software utilizados, una descripción detallada de su desarrollo y las pruebas aplicadas a éste que muestran su confiabilidad en el manejo de datos.

D e d i c a t o r i a

Dedico este trabajo de tesis principalmente a mis P a d r e s quienes con amor y cariño me inculcaron siempre el estudio y conclusión de las metas que me propongo a pesar de las adversidades, apoyándome siempre en cualquier proyecto o idea que yo quise emprender. "Nunca me hago a un lado, P a p á s".

Dedico también este trabajo a mi querida hermana y mis queridos sobrinos el "chango" (Axel) y la "pelusa" (Briana).

A g r a d e c i m i e n t o s

Agradezco primeramente a la Universidad Autónoma de Baja California y a CONACYT por el apoyo económico, instalaciones y recursos que permitieron realizar mis estudios de posgrado.

A mi Asesor y Director de Tesis Dr. Juan Ivan Nieto Hipólito por la dirección, seguimiento y apoyo en este trabajo, siempre de una manera amable, responsable y profesional.

A mis maestros que durante el estudio de la maestría me brindaron tiempo, experiencia, consejos y conocimientos para el desarrollo del presente trabajo y en general para mi desarrollo profesional.

A mis amigos Verónica Aguilar, Oscar Salazar, Victor Ramírez, Nancy Ortega, Claudia López, Esteban Rodríguez y Raúl Jiménez por su apoyo y amistad genuina e incondicional que siempre me han brindado.

Agradezco finalmente a mis compañeros de Maestría y Doctorado por sus consejos y aportaciones a mi trabajo.

Índice general

Índice de figuras

Capítulo 1

Introducción

1.1. Redes de Sensores Inalámbricos

Una Red de Sensores Inalámbrica *(WSN ó Wireless Sensor Network por sus siglas en inglés)* se conforma por un grupo de dispositivos autónomos generalmente pequeños, distribuidos sobre cierto espacio limitado en cobertura y diseñados para trabajar en conjunto y ser capaces de monitorizar condiciones físicas y ambientales como pueden ser temperatura, sonido, vibración, presión, entre otros [16].

Cada uno de estos dispositivos se compone de una unidad inalámbrica de comunicación, un microcontrolador, uno o más sensores y una fuente de energía, usualmente una batería. El objetivo de estos nodos llamados también motes es recolectar datos por medio de los sensores y transmitirlos a una central base en la cual se pueda disponer de dichos datos para realizar acciones en base a la información obtenida.

Una *WSN* se puede componer de una o más celdas. Una celda se compone de un grupo determinado de motes dependiendo del área a monitorizar y en cada una de dichas celdas se encuentra un nodo coordinador que es el encargado de recolectar los datos de todos los motes existentes en su celda y enviar toda esta información a un repositorio central que generalmente es una computadora. Esta central recibe la información proveniente de todos los nodos coordinadores que estén dentro de la red y una vez aquí dicha información puede ser manipulada de forma local y/o remota. La figura 1.1 muestra 2 tipos de redes según su topología (estrella y punto a punto).

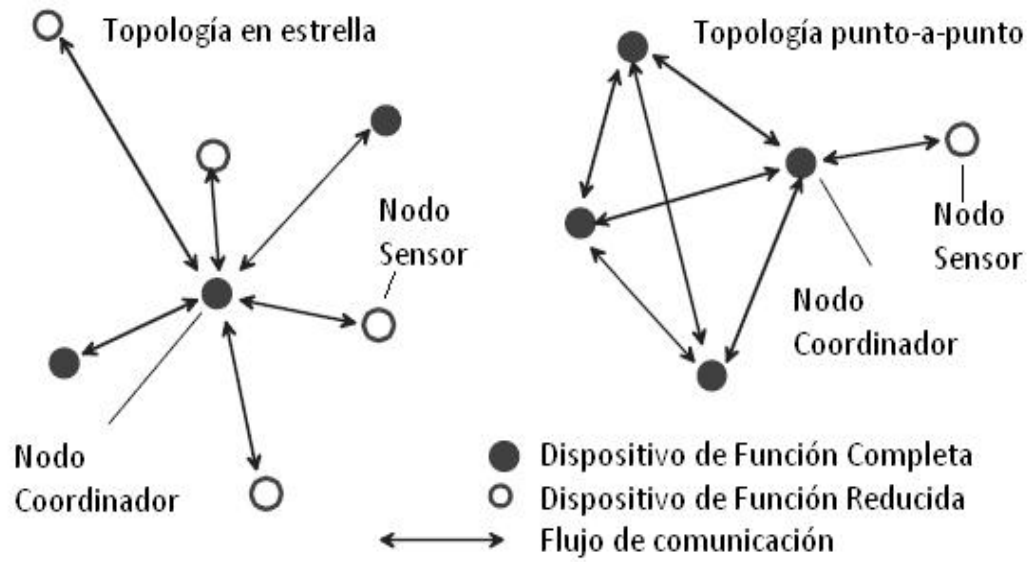

Figura 1.1: Red Inalámbrica de Sensores

Las *WSN´s* son una tecnología que ha evolucionado de forma significativa y aunque en sus inicios se enfocó -como muchos de los avances tecnológicos importantes- en el área militar, rápidamente se extendió a otras áreas.

Una de las áreas donde las *WSN´s* son aplicadas es en el área de la agricultura donde se pueden obtener beneficios como reducción en el consumo de agua y pesticidas, lo cual contribuye a la preservación del entorno. Por medio de sensores estratégicamente situados, se pueden monitorizar parámetros tales como el clima, la temperatura, humedad de las hojas, entre otros [2].

El cuidado de la salud de personas es una tarea que resulta muchas veces tediosa, delicada, exhaustiva y compleja para los encargados de prestar estos servicios a pacientes; mediante la implementación de una WSN con sensores situados de forma estratégica dentro del área de interacción de la persona a ser monitoreada, así como en objetos de uso cotidiano, dichas personas encargadas del cuidado de pacientes podrían monitorizar las actividades de pacientes en tiempo real y de esta manera tomar las medidas necesarias para mejorar su calidad de vida [13].

El uso de esta tecnología también se usa en la monitorización de signos vitales de pacientes como pueden ser: ritmo cardiaco, concentración de oxígeno en la sangre, datos *EKG* de electrocardiograma, entre otros. Un ejemplo de esto es el proyecto *CodeBlue* [15] (ver figura 1.2), desarrollado en la Universidad de Harvard. Toda la información es captada por los sensores y distribuida de forma inalámbrica a un *PDA por sus siglas en inglés Personal Digital Assistant (Asistente Digital Personal)* o una computadora, de esta manera se centraliza y procesa.

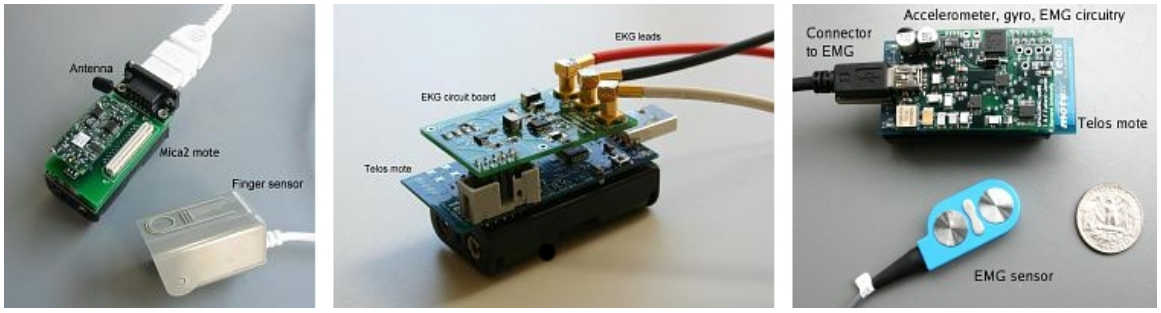

Figura 1.2: Proyecto *Code Blue*

Se podría implementar una red con un conjunto de cámaras miniatura interconectadas capaces de procesar, enviar y recibir audio y video diseñando así sistemas sofisticados de vigilancia que pueden resultar muy eficaces en contra del crimen, como podría ser la monitorización de áreas públicas (estacionamientos, parques, entre otros.) y privadas, fronteras, entre otros [5].

En los capítulos siguientes se presenta el proyecto de tesis: puente WiFi-WSN, el cual es un prototipo de un dispositivo puente que comunica los estándares IEEE 802.11 y el IEEE 802.15.4 *(WiFi-WSN)*.

1.2. Planteamiento del problema

Con el surgimiento de nuevas tecnologías de red, protocolos, dispositivos, y demás, se hace presente la investigación y desarrollo de técnicas de miniaturización y bajo consumo de

energía, alimentación remota e inalámbrica de nodos, diseño de protocolos y dispositivos que permitan el enlace y transporte de datos y todo esto con un costo accesible para el usuario final.

Actualmente existe un sinnúmero de tipos de redes cada una con aplicaciones propias. Sin embargo el lograr conectar 2 tipos de red con protocolos totalmente diferentes es un reto y es un área en constante crecimiento y múltiples aportaciones por las ventajas que ésta ofrece.

Se buscan alternativas de bajo costo que permitan incrementar el uso y desarrollo de *WSN's*, siendo una de éstas el desarrollo de puentes (*Bridges*) / puertas de enlace (*Gateways*) que permitan la interconexión de múltiples protocolos de forma práctica y así diversificar las aplicaciones.

Múltiples aplicaciones requieren de la coexistencia o comunicación de dos o más protocolos, es aquí donde entra el presente proyecto en el cual se diseñó un puente que permite enlazar dos estándares (IEEE 802.15.4 y IEEE 802.11) que puede servir como base para el desarrollo de diferentes aplicaciones de forma práctica, segura, económica y transparente al usuario, creando una conexión entre dos puntos para transportar datos y así aumentar la funcionalidad de la red. La figura 1.3 muestra de forma abstracta el puente desarrollado.

Figura 1.3: Puente WiFi/wsn

1.2.1. Justificación

Actualmente existen múltiples esquemas de red donde es posible que coexistan diferentes tecnologías y protocolos. Una *WMN* (*Wireless Mesh Network* - Red inalámbrica heterogénea) se compone de un conjunto de nodos de diferente tipo que establecen una red *ad hoc* [12] y mantienen la conectividad de la red. Es una red que puede contener otras redes de diferente tipo como pueden ser Wi-Fi (IEEE 802.11)[28], redes de celular, redes *WiMAX* (IEEE 802.16), *WSN´s* (IEEE 802.15.4)[25] o clientes alámbricos e inalámbricos [11].

Una *WMN* se compone de dos tipos de nodos: *routers mesh* y clientes *mesh*. Un *router mesh* contiene enrutamiento adicional en comparación con un *router* inalámbrico convencional. Un *router mesh* es usualmente equipado con múltiples interfaces inalámbricas de acceso por lo que nodos convencionales equipados con *NIC's* (*Network Interface Cards* - Tarjeta de interfaz de red) se pueden conectar directamente a una *WMN* a través de *routers mesh* inalámbricos. La figura 1.4 muestra el concepto de una *WMN*.

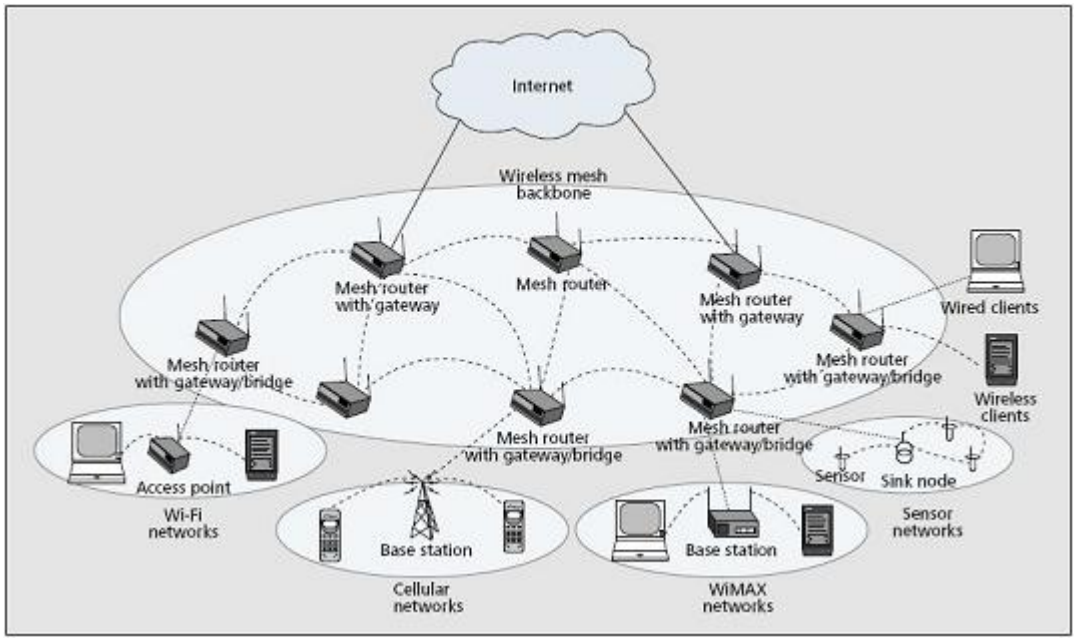

Figura 1.4: Red Inalámbrica Heterogénea - *WMN (Wireless Mesh Network)*

Las *WMNs* diversifican las capacidades de redes *ad-hoc*, esto provee muchas ventajas como el bajo costo y su fácil mantenimiento, robustez, cobertura de servicio segura, entre

otras. Esto provoca que las *WSN's* actualmente experimenten una amplia comercialización en muchos escenarios de aplicación.

Para hacer posible el desarrollo de *WMN's* se requiere de investigación en muchos ámbitos. Por ejemplo el *MAC* (*Medium Access Control*, es el mecanismo encargado del control de acceso de cada dispositivo al medio) [3] disponible y protocolos de enrutamiento aplicados a *WMN's* no tienen suficiente escalabilidad (permite incrementar el número de dispositivos presentes en determinada red. La escalabilidad es importante ya que existen situaciones donde se require que un gran número de dispositivos se encuentren intercomunicados). La escalabilidad de *WMN's* puede ser direccionada por la capa *MAC* de dos formas. La primera forma es aumentar los protocolos *MAC* existentes para incrementar el rendimiento cuando solo un canal está disponible en un nodo de red. La segunda forma es permitir la transmisión sobre múltiples canales en cada nodo de red [11].

Si hablamos de la interacción de múltiples redes (protocolos), coexistiendo entre sí, debemos hablar sobre la conectividad que permite esta mezcla; la forma en que es posible ir de un nodo a otro pasando sobre diferentes protocolos y dispositivos, y mediante esto lograr transmitir la información lo más fiel y rápido posible de un punto a otro. Esto nos lleva al estudio y desarrollo de dispositivos que permitan enlazar 2 o más protocolos y así establecer la comunicación entre estos tipos de redes.

La presente investigación se centra en el desarrollo de uno de estos dispositivos que permite enlazar dos protocolos y promueve así la heterogeneidad de redes con sus múltiples beneficios, entre ellos el costo. Este dispositivo prototipo forma parte de un proyecto desarrollado en conjunto por una serie de estudiantes y maestros de la Universidad Autónoma de Baja California - Facultad de Ingeniería - Unidad Ensenada, cuya meta final es diseñar una plataforma de *e-salud* generada en su totalidad con tecnología propia.

El objetivo del desarrollo del puente es la funcionalidad que al ser integrado, le dará a la plataforma *e-salud* del proyecto antes mencionado, sin embargo este dispositivo no se limita a ser utilizado únicamente en este proyecto, ni a proyectos relacionados exclusivamente al área de salud. Este puente permitirá integrar cualquier Red inalambrica de sensores [25] a

un sistema IEEE 802.11 [28] y de ésta manera aprovechar sus beneficios. La figura 1.5 representa el proyecto.

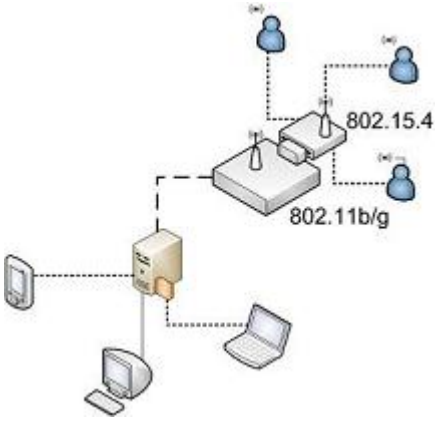

Figura 1.5: Proyecto plataforma e-salud

1.3. Antecedentes

En 1993, la IEEE y el *National Institute for Standards and Testing (NIST)* iniciaron una actividad que condujo al desarrollo de un estándar para sensores inteligentes, al cual asignaron el número de IEEE-1451 [6]. El objetivo fue desarrollar una familia de estándares para interconectar sensores inteligentes [32]. Este conjunto de estándares está encaminado a facilitar el desarrollo de dispositivos inteligentes que puedan interconectarse en redes, sistemas e instrumentos, los cuales puedan incorporarse a sensores y redes existentes y emergentes.

Básicamente dos clases de dispositivos son descritos por el estándar: El Procesador de las aplicaciones en la red, NCAP por sus siglas: *Network Capable Applications Processor (NCAP)*, y el módulo de los transductores. El *NCAP* forma el puente entre la red y el modulo de los transductores [33]. A continuación se describe la familia del IEEE-1451 que a la fecha se ha definido:

El IEEE-1451.1, aprobado en 1999, define un modelo de objeto del software común para el *NCAP*, que tiene distintas versiones dependiendo de la red y el tipo de módulo de

los transductores [33] [32].

El IEEE-1451.2 aprobado en 1997, en el cual se define la forma en que los transductores son incrustados o incluidos en el módulo interfaz del transductor inteligente, *STIM* por sus siglas: *Smart Transducer Interface Module*, conectado al *NCAP* por medio de una conexión punto a punto. Y también define el formato electrónico de las especificaciones del transductor *TEDS*, por sus siglas *Transducer Electronics Data Sheet (TEDS)* [33] [32].

El IEEE-1451.3 aprobado en el 2003, define la forma en que los transductores son incluídos en el módulo interfaz del transductor inteligente, *STIM* por sus siglas: *Smart Transducer Interface Module*, conectado al *NCAP* por medio de una conexión multipunto. Y también define el formato electrónico de las especificaciones del transductor *TEDS*, por sus siglas *Transducer Electronics Data Sheet (TEDS)* [33] [32].

El IEEE-1451.4 aprobado en el 2004, define la forma en que se agrega la función de autoidentificación a sensores y actuadores tradicionales del modo analógico. Define el concepto de un transductor del modo mixto que alimenta una interfaz analógica y digital. Y también define el formato electrónico de las especificaciones del transductor *TEDS* [33] [32].

El IEEE-1451.5 aprobado en el 2007, define o establece el estándar para una comunicación inalámbrica y el formato de la información para el transductor. Y también define el formato electrónico de las especificaciones del transductor *TEDS*, de acuerdo al formato IEEE-1451 [32] [34].

El IEEE-P1451.6 actualmente se encuentra en desarrollo, el propósito de este estándar es definir la implementación de los conceptos del *TEDS* de sobre una red con *CAN*. El propósito es desarrollar un puerto de entrada simple para una red de sensores en cascada combinando las especificaciones del IEEE 1451 y el *CAN* [32] [34].

El IEEE 1451.0 aprobado en el 2007, contiene un nuevo estándar que define un conjunto de funciones comunes, protocolos de comunicación y formatos *TEDS* para varias formas de comunicación. Con el propósito de lograr la interoperabilidad del estándar y simplificar su aplicación.

Este estándar utiliza la familia IEEE 802 como la base de protocolos de comunicación.

En el año 2001 una nueva iniciativa apuntó hacia la revisión de estándares con la meta de extender algunas partes de la familia 1451 para satisfacer la demanda de la nueva industria. A raíz de esto se adoptaron los siguientes protocolos de comunicación: *bluetooth* con IEEE 802.15.1, *ZigBee* con IEEE 802.15.4 y *WiFi* con IEEE 802.11. Éstos dos ultimos estándares son la base del desarrollo del presente proyecto.

En Febrero de 2003 el *MIT* (*Massachusetts Institute of Technology* - Instituto Tecnológico de Massachusetts) identificó las 10 tecnologías emergentes que cambiarán el mundo, y en primer lugar aparecen las *WSN's (Wireless sensor networks)*. Las pruebas comerciales para las *WSN* iniciaron en el año 2004 [10].

En el año 2006 se ratificó el estándar IEEE 802.15.4, propio de las *WSN´s*, y aunque éstas redes han demostrado un excelente desempeño dentro de su área de cobertura, su potencial de desarrollo y aplicación es grande y se hace indispensable su interacción con otros ambientes, es así como surge la necesidad de intercomunicarlo con otros sistemas de comunicación, como los sistemas alámbricos tradicionales y redes inalámbricas de cobertura más amplia; de esta manera se aprovecharán al máximo sus capacidades.

1.4. Objetivos

1.4.1. Objetivo General

Desarrollar un puente de comunicación que permita intercambiar datos entre una red de sensores inalámbrica y un servidor embebido IEEE 802.11 estableciendo así la comunicación entre los estándares IEEE 802.11 e IEEE 802.15.4.

1.4.2. Objetivos Específicos

Para cumplir con el objetivo general, se plantearon los siguientes objetivos específicos:

- Diseño de una interfaz que comunique un mote IEEE 802.15.4 y un nodo IEEE 802.11 utilizando el puerto serial de ambos.
- Diseño de una aplicación que permita la comunicación entre ambos nodos formando así un puente que servirá como nodo coordinador dentro de la *WSN*.
- Diseño de una red bajo el estándar IEEE 802.15.4 permitiendo la comunicación entre nodo coordinador y cada uno de los motes de su celda.
- Enlace de la red de sensores con el dispositivo IEEE 802.11 mediante el nodo coordinador.
- Diseño de un sistema de administración del puente.
- Pruebas del funcionamiento general y específico de todos los puntos de comunicación de la red heterogénea, siendo la confiabilidad el aspecto relevante.

1.5. Metodología

- Investigación en diversas fuentes de información.
- Elección y adquisición de 1 dispositivo modulo inalámbrico embebido de tipo serial a IEEE 802.11, así como un kit de desarrollo y motes que soporten el estándar IEEE 802.15.4.
- Elección, adquisición y aprendizaje de un lenguaje para la programación de nodos y motes.
- Diseño de un nodo coordinador que une dos nodos de diferente estándar: uno con el protocolo IEEE 802.15.4 y el otro con el protocolo IEEE 802.11 de forma alámbrica por el puerto serie.
- Diseño de una interfaz que comunica ambos nodos por medio del puerto serie.

- Diseño de una aplicación que establece la comunicación entre el nodo coordinador y cada uno de los motes de su celda.
- Diseño de un sistema de administración embebido de tipo web.
- Pruebas del funcionamiento de todos los puntos de comunicación de la WSN realizando pruebas para obtener estadísticas y eficiencia de la WSN.

1.6. Metas

- Obtención como producto final un dispositivo estable y seguro que permite la comunicación entre los dos estándares (802.11 y 802.15.4).
- Evaluación del puente desarrollado y la red.

1.7. Delimitación del problema e interrogantes del estudio

- Se utilizó un dispositivo para el desarrollo de aplicaciones 802.11 que cuenta con un puerto serie.
- Se utilizó un mote con estándar 802.15.4 que cuenta con un puerto serie y con un puerto de entrada y salida genérico.
- No se diseñó ningún tipo de hardware, salvo de interconexión de dispositivos.
- La tesis se enfocará al desarrollo de programación necesaria para la comunicación entre los estándares antes mencionados.

Capítulo 2

WiFi y WSN

2.1. Introducción

La aplicación de la tecnología inalámbrica se disparó de forma exponencial en los últimos años estando presente en la mayoría de las tecnologías de comunicación actuales, siendo la clave de motivación para su uso, la reducción en los gastos de instalación, facilidad, movilidad, entre otros.

Es una tecnología con una amplia gama de aplicaciones, por ejemplo conexión a Internet, redes de computadoras, redes de audio y video, automatización, seguridad, entre otros. Cada una de ellas tiene diferentes necesidades de ancho de banda, costos y procedimientos de instalación.

Por otro lado las aplicaciones como la automatización del hogar y aplicaciones de seguridad han relajado necesidades de ancho de banda principalmente. Estas aplicaciones no pueden manejar protocolos muy pesados ya que afectarían seriamente el consumo de energía y requerirían de mayor poder de procesamiento.

Por ejemplo un sensor de temperatura pequeño en una ventana solo necesita reportar sus datos pocas veces por hora, es discreto y tiene un precio muy bajo, por lo cual se manejaría muy bien con un enlace de comunicación inalámbrica de baja potencia. El uso de cables seria impráctico por el uso mismo de la ventana. Además, los costos de la instalación del cable excederían el costo del sensor. Se buscaría que el consumo de energía fuera poco, ya que el cambio constante de baterías resulta impráctico. La tecnología *WiFi* ó IEEE 802.11

resultaría sofocante ya que solo satisface los requerimientos de conexión.

Es aquí donde entra la necesidad de un nuevo estándar para redes inalámbricas de bajo poder y por consecuencia bajos costos en ambientes tanto industriales como caseros y por consiguiente se da la entrada al nuevo estándar de baja transmisión en redes inalámbricas para áreas personales, nace así el estándar IEEE 802.15.4, el cual encaja en el ejemplo.

Sin embargo vayamos mas allá, supongamos que se requiere recolectar la información de todas las ventanas de determinado edificio para su monitorización. Es un flujo mayor de datos donde aquí si sería útil el uso de la tecnología IEEE 802.11 para su agrupación y envío masivo por decirlo de esta manera del conjunto de datos obtenidos.

Este capítulo presenta los estándares IEEE 802.11 y IEEE 802.15.4

2.2. IEEE 802.11 (*Wi-Fi*)

Wi-Fi es el nombre dado por la alianza Wi-Fi al grupo de estándares del IEEE 802.11 [28]. IEEE 802.11 definió el estándar inicial para las redes de área local inalámbricas (*WLANs*), pero era considerado demasiado lento para algunas aplicaciones y fue reemplazado por las extensiones IEEE 802.11a y IEEE 802.11b, más adelante por IEEE 802.11g y el lanzamiento más reciente IEEE 802.11n. IEEE 802.11b describe la capa de acceso al medio y enlace para su implementación en la banda de 2.4 GHz, la cual puede comunicarse a una velocidad de transmisión de 11 Mbit/s. Otros estándares describen una puesta en práctica más rápida (54 mbit/s) en la banda de 2.4 Ghz (IEEE 802.11g). Wi-Fi (IEEE 802.11b/g) es la implementación más común y más rentable actualmente disponible. Ésta es la implementación que se utiliza con el módulo *RCM5600W MiniCore* utilizado en el proyecto de tesis que se presenta. Existe una amplia variedad de hardware *WiFi*; puntos de acceso inalámbricos (*WAP´s*), Dispositivos de acceso *WiFi* con *PCI, PCMCIA, CompactFlash, USB e interfaces SD/MMCA*, existen también dispositivos *WiFi* como cámaras basadas en Web y servidores de impresión [9].

2.2.1. Ventajas del *WiFi*

La adición de radio en comparación con el sistema tradicional alámbrico proporciona más opciones para la supervisión, el control y la difusión de la información. En la práctica, las posiciones remotas llegan a ser más accesibles y los costos más bajos.

La lista siguiente resume algunas de las ventajas de una red *WiFi*.

- Ethernet inalámbrico. *WiFi* es un reemplazo de Ethernet. *WiFi y Ethernet*, ambas IEEE 802 redes, comparten algunos elementos de base.
- Acceso extendido. La ausencia de cables amplía el acceso a los lugares en donde los cables no pueden ir o donde es demasiado costoso instalarlos.
- Reducción de costos. Según lo mencionado anteriormente, la ausencia de cable disminuye los costos. Esto se logra por una combinación de factores; costo relativamente bajo de routers inalámbricos, perforación y otros métodos que puedan ser necesarios para hacer conexiones físicas, entre otros.
- Movilidad. Una red cableada permanece sobre un mismo lugar físico. Los medios inalámbricos tienen libertad para cambiar su localización sin perder su conexión.
- Flexibilidad. El acceso extendido, las reducciones de costos, y la movilidad crean las oportunidades para nuevas aplicaciones y soluciones creativas.

2.2.2. Aplicaciones de sistema embebidos *WiFi*

El alcance de la comunicación inalámbrica en sistemas embebidos continúa creciendo. *Forrester* [31], una compañía que se centra en investigaciones acerca de cómo la tecnologia implica un cambio en la manera de ver los negocios, ha divulgado que en pocos años, hasta el 95 % de los dispositivos usados para tener acceso a internet no serán *PCs*, sino que serán sistemas embebidos. Existen muchas aplicaciones para los dispositivos embebidos con un interfaz *WiFi*:

- Aplicaciones para control y proceso industrial donde el uso de conexiones alámbricas sería demasiado costoso. Ej. maquinaria continuamente móvil.
- Aplicaciones de emergencia que requieren la disposición inmediata y transitoria, tal como campo de batalla o situaciones del desastre.
- Aplicaciones móviles, tales como seguimiento del activo.
- Cámaras de vigilancia, ya que los cables son difíciles de ocultar, por lo que resulta ideal que sea inalámbrico
- Los mercados verticales como médico, educación, y fabricación.
- Comunicación con otros dispositivos *WiFi*, como una computadora portátil o un *PDA*.

2.2.3. Arquitectura

La arquitectura describe la estructura y la organización de la red. Esto permite seleccionar el mejor modo de operación según la aplicación.

Componentes básicos

Todos los dispositivos inalámbricos que se unen a una red *WiFi*, sean móviles, portables o fijos, se llaman estaciones inalámbricas (abreviado en ingles *STAs*). Una estación inalámbrica puede ser una *PC*, una computadora portátil, un *PDA*, un teléfono, o un módulo del tipo *Rabbit*. Cuando dos o más *STAs* inalámbricas están conectadas, forman un sistema del servicio básico (abreviado en ingles *BSS*). Éste es el bloque básico de una red *WiFi*.

Un BSS es un sistema de *STAs* controlado por una sola función de coordinación (*CF*). El *CF* es una función lógica que determina cuando un *STA* transmite y cuando recibe.

El *BSS* demostrado en la figura 2.1 es un ejemplo de la Red *WiFi* más simple posible: dos estaciones inalámbricas. La forma oval alrededor de ellas representa el área de la cobertura.

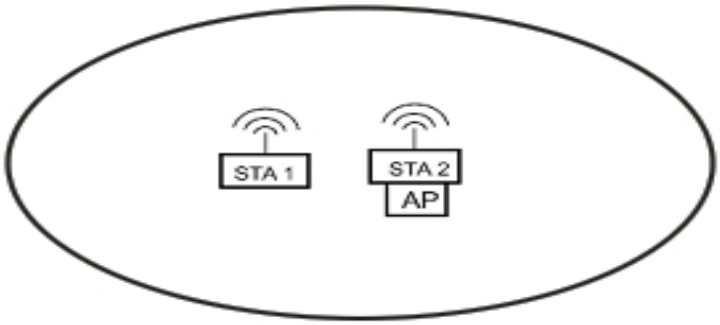

Figura 2.1: *BSS-a*

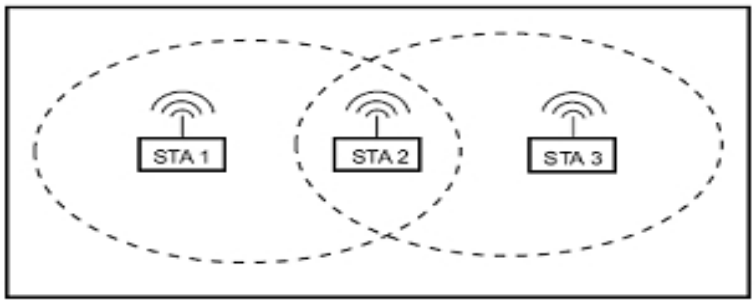

Figura 2.2: *BSS-b*

Mientras que un círculo puede representar el área de cobertura idealizada de un solo radio, no es muy exacto en situaciones del mundo real. Los factores ambientales causan variaciones dramáticas al área de cobertura. Por ejemplo, un *STA* con una antena omnidireccional colocado en la esquina de un edificio puede tener la mayor parte de su área de la cobertura fuera del edificio y en el estacionamiento adyacente.

No todos los *STAs* en un *BSS* pueden necesariamente comunicarse directamente. La figura 2.2 muestra que el *STA 1* y el *3* están mutuamente fuera de rango, por lo tanto se requiere el uso del *STA 2* para reenviar mensajes.

Modos de funcionamiento

El estándar de IEEE 802.11 especifica dos modos de funcionamiento: modo tipo infraestructura y modo tipo *ad hoc*. Cada uno hace uso de *BSS*, sin embargo ofrecen diferentes topologías de red. El modo de funcionamiento se selecciona durante la configuración de la estación inalámbrica. Todas las estaciones inalámbricas deben seleccionar un modo de funcionamiento antes de intentar crear o unirse a una red *WiFi*.

Modo *ad hoc*

La *BSS* de tipo independiente (*IBSS*) es el tipo más simple de una red IEEE 802.11. Las estaciones inalámbricas se comunican directamente la una con la otra usando el modo de funcionamiento *ad hoc*. Tal red sigue un modelo *peer-to-peer*. Un funcionamiento de *BSS* en modo *ad hoc* se aísla. No hay conexión a otras redes *WiFi* ni *LANs* (*Local Area Network* ó Redes de Área Local). Sin embargo, el modo *ad hoc* puede ser muy útil en muchas situaciones, por ejemplo en situaciones donde se requiere de una rápida instalación, donde no se cuenta con infraestructura como sitios de emergencia y zonas de combate.

Modo infraestructura

El modo de funcionamiento infraestructura requiere que los *BSS* contengan un punto de acceso inalámbrico (*AP*). Un *AP* es un *STA* con funcionalidad adicional. El papel principal de un AP es ampliar el acceso a las redes alámbricas para los clientes de una red inalámbrica. Todos los dispositivos inalámbricos que se intentan unir al *BSS* deben asociarse a un *AP*. Un *AP* proporciona el acceso a su *STAs* asociado, el cual es llamado sistema de distribución (*DS*). El *DS* es un componente que permite la comunicación entre los *APs*. Toda la comunicación inalámbrica a o desde un *STA* asociado pasa a través de un *AP* cuando la red se configura para utilizar modo infraestructura. La figura 2.3 muestra este concepto.

En la especificación IEEE 802.11, el AP ("hub") y *"host"* se llaman estaciones inalámbricas o *STAs*.

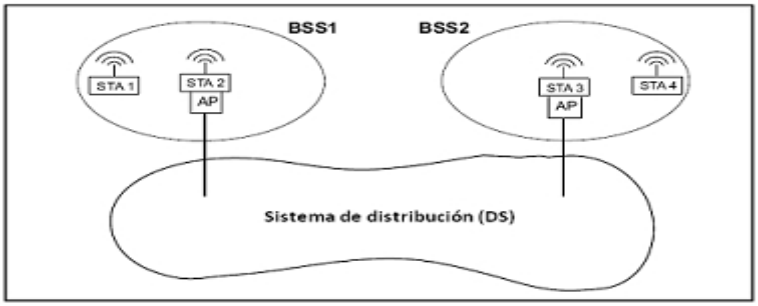

Figura 2.3: Comunicación de *APs* utilizando *DS*

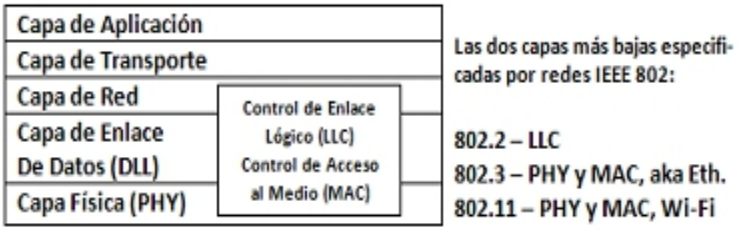

Figura 2.4: Modelo TCP/IP de 5 Capas

2.2.4. Modelo de internet (TCP/IP)

IEEE 802.11 y sus extensiones (a, b, g y n) definen dos capas en el modelo de TCP/IP de cinco-capas: la capa física y la capa de enlace de datos. Éstas son las mismas dos capas que son definidas por IEEE 802.3 (*Ethernet*). La capa de enlace de datos se compone realmente de dos capas: Control de acceso al medio - *Media Access Control (MAC)* y control de enlace lógico - *Logical Link Control (LLC)*. La especificación de IEEE 802.11 define la subcapa *MAC*. La figura 2.4 muestra el modelo TCP/IP de 5 Capas.

Capa física del IEEE 802.11 (*PHY*)

Hay varias capas físicas descritas en la especificación IEEE 802.11 y sus extensiones. La capa física es responsable de cosas tales como métodos de la modulación, esquemas de codificación y la transmisión real de las señales de radio a través de espacio. Implementaciones de capa física funcionan en bandas específicas. Una banda define las frecuencias asignadas para aplicaciones particulares. Muchos dispositivos *WiFi* se diseñan para el uso en la

banda industrial, científica y médica (*ISM*). La banda *ISM* es para dispositivos libres de licencia; los requisitos reguladores exigen que los dispositivos libres de licencia utilicen tecnología del espectro esparcido. El espectro esparcido de secuencia directa (*DSSS*) es el más ampliamente utilizado.

La capa física IEEE 802.11 tiene algunas características:

- No tiene protección de otras señales que puedan compartir el medio.
- Se comunica sobre un medio menos confiable que capas físicas alámbricas.
- Tiene topologías dinámicas.
- Carece de conectividad completa.
- Tiene propiedades de variación de tiempo y propagación asimétrica.
- Puede experimentar interferencia de otras redes IEEE 802.11 operando sobre las mismas áreas de cobertura.

Capa de Control de Acceso al Medio de IEEE 802.11 (*MAC*)

La capa *MAC* del IEEE 802.11 es técnicamente una subcapa de la capa de enlace de datos (*DLL*). Se monta sobre la capa física, controlando la transmisión de datos y provee interacción con una red alámbrica base, si existe una. La capa del *MAC* también proporciona los servicios relacionados con la radio y manejo de movilidad. Para mover los paquetes de datos a través de un canal compartido, la capa *MAC* utiliza *CSMA/CA* Protocolo de Acceso Múltiple con Detección de Portadora y Evasión de Colisiones *(Carrier Sense Multiple Access with Collision Avoidance)* para acceder al medio físico, que es muy similar a la estrategia que se utilizó en IEEE 802.3 capas del *MAC*: *CSMA/CD* (detección de colisión). Ambos son protocolos *peer-to-peer*, difieren solo en que *CSMA/CD* se ocupa de las transmisiones después de que ha ocurrido una colisión, y *CSMA/CA* actúa para prevenir colisiones antes de que sucedan.

2.2.5. Servicios especificados por IEEE 802.11

El estándar de IEEE 802.11 no define ninguna implementación específica. En lugar, se especifican nueve servicios que todas las implementaciones deben proporcionar, que son los siguientes:

Servicios de la estación (*SS*)

Todas las estaciones inalámbricas IEEE 802.11 (*STA´s*) deben ejecutar los cuatro servicios de la estación definidos en la especificación de IEEE.

Los servicios son:

- Autentificación - Una estación inalámbrica necesita ser identificada antes de que pueda tener acceso a los servicios en red. Este proceso se llama autentificación.
- Desautentificación - Éste servicio anula una autentificación existente.
- Privacidad - Una estación inalámbrica debe poder cifrar marcos para proteger el contenido del mensaje de modo que solamente el receptor previsto pueda leerlo.
- Servicio de entrega de la Unidad de Datos de Servicio (*MSDU*) del *MAC* - Un *MSDU* es un marco de datos que se debe transmitir al destino apropiado.

Servicios del sistema de distribución (*DSS*)

Una estación inalámbrica que funciona como un punto de acceso y debe ejecutar los cuatro servicios de la estación más los servicios del sistema de distribución enumerados aquí:

- Asociación - Éste servicio establece un *AP/STA* después que la autentificación ha tomado lugar entre las dos estaciones inalámbricas. Un *STA* puede asociarse solamente a un *AP* a la vez. Este servicio es iniciado siempre por la estación inalámbrica y cuando finaliza con éxito permite el acceso de la estación al *DSS*.
- Reasociación - Éste servicio mueve una asociación actual a partir de un *AP* a otro *AP*.
- Desasociación - Éste servicio anula una asociación actual.

- Distribución - Éste servicio maneja la entrega de *MSDU´s* dentro del sistema de distribución; es decir, intercambio de marcos de datos entre los *AP´s* en un sistema extendido del servicio (*ESS*).

- Integración - Éste servicio maneja la entrega de MSDUs entre el sistema de distribución y una estación. Ésta es básicamente la función de enlace entre redes inalámbricas y alambricas.

2.2.6. Modelo de referencia

Las 2 grandes partes de este estándar son la capa de enlace de datos y la capa física, las cuales corresponden a las capas mas bajas del modelo de interconexión de sistemas abiertos (*OSI*). Las capas y subcapas se muestran en la figura 2.5.

Los diferentes segmentos del modelo de referencia *ISO/IEC* se comunican por medio de puertos. Iniciando en la capa física, la subcapa *PMD (Physical Medium Dependent ó Dependiente del medio físico)* se comunica con la subcapa *PLCP* (*Physical Layer Convergence Procedure* ó Procedimiento de convergencia de la capa física) por medio del puerto *PMD-SAP* (*Physical Medium Dependent - Service Access Point* ó Dependiente del medio físico - Punto de acceso al servicio). Tanto la subcapa *PLCP* como la *PMD* están en comunicación directa con la entidad de administración de la subcapa física ya que se encuentran al mismo nivel. La subcapa *MAC* se comunica con la capa superior (RED) por medio del puerto *MAC-SAP* (*Medium Access Control - Service Access Point* ó Control de Acceso al Medio - Punto de acceso al servicio) y directamente con la entidad de administración de la subcapa *MAC*. Ésta última se comunica con la entidad de administración de la estación por medio del puerto *MLME-SAP* (*MAC Sublayer Management Entity - SAP* ó Entidad de administación de la subcapa *MAC - SAP*). La entidad de administración de la subcapa física se comunica con la entidad de administracion de la estación por medio del puerto *PLME-SAP* (*Physical Sublayer Management Entity - SAP* ó Entidad de administración de la subcapa física - *SAP*).

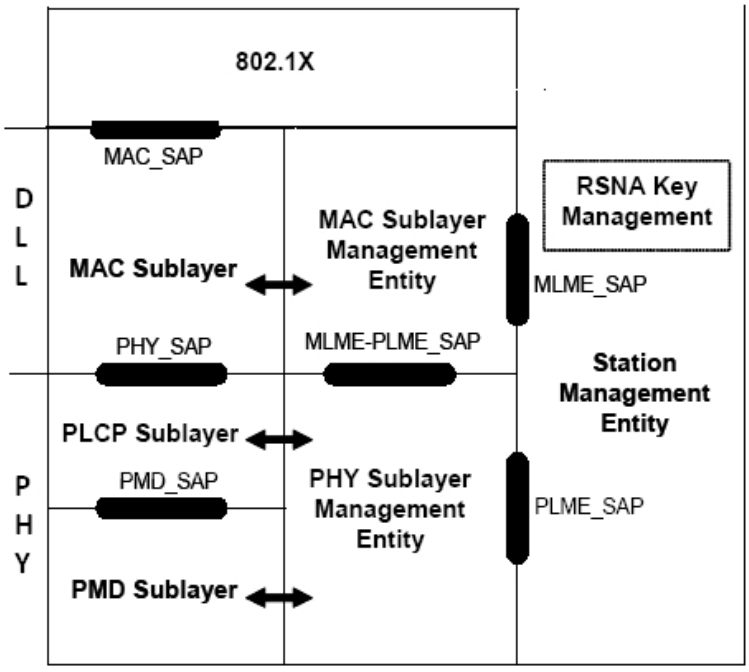

Figura 2.5: Modelo de referencia ISO/IEC

2.2.7. Marcos

Un *STA* debe ser capaz de construir un grupo de marcos especificados para transmisión y decodificación de marcos. El conjunto de marcos que un *STA* construye y decodifica es determinado por las funciones soportadas por ese *STA*. Todos los *STA´s* deben ser capaces de validar cada marco recibido usando sequencia de revisión de marcos *(FCS - Frame Check Sequence)* e interpretar ciertos campos de las cabeceras *MAC* de todos los marcos.

La figura 2.6 muestra el formato de marcos del estandar IEEE 802.11. El puente convierte marcos provenientes de la red de sensores (IEEE 802.15.4) a éste estándar (IEEE 802.11). Los nodos sensores envian paquetes al nodo coordinador de la red de sensores, éste los recibe, los desempaqueta y por medio del puerto serial transfiere únicamente la carga útil al dispositivo Rabbit el cual lo empaqueta de acuerdo al estándar IEEE 802.11.

Formatos de marcos *MAC*

Cada marco consiste de los siguientes componentes básicos:

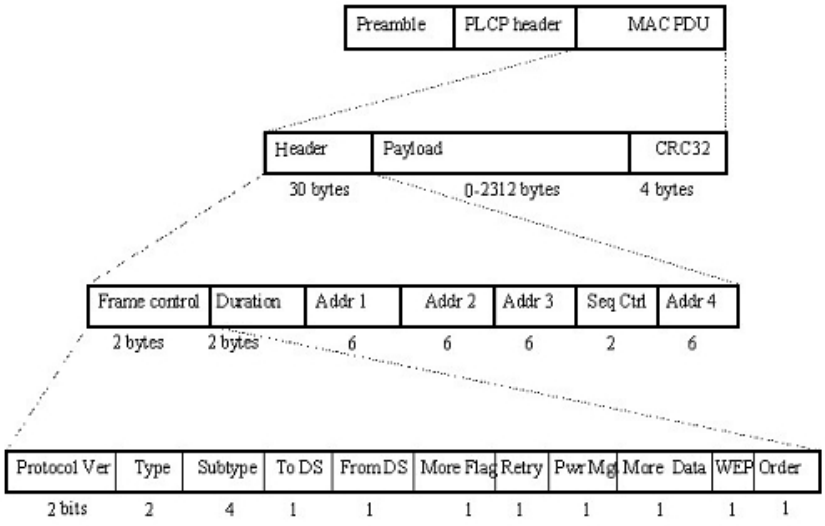

Figura 2.6: Formato de marco IEEE 802.11

- Una cabecera MAC abarca control de marcos, duración, dirección e información de control de secuencia, y para marcos de datos con calidad de servicio, información de control de calidad de servicio.
- Un cuerpo de marco de longitud variable, el cual contiene información específica de el tipo de marco y subtipo.
- Un *FCS*, el cual contiene un IEEE 32-bit CRC.

Formato general del marco

El formato de un marco MAC abarca un grupo de campos. La figura 2.7 muestra el formato de marcos MAC. Los primeros tres campos (Control de marco, Duración/ID, y Dirección 1) y el último campo (*FCS*) constituye el mínimo formato de marco. Los campos Dirección 2, Dirección 3, Control de secuencia, Dirección 4, Control de calidad de servicio y cuerpo del marco están presentes solo en ciertos tipos y subtipos de marco. El campo cuerpo del marco de de tamaño variable. El tamaño máximo del cuerpo del marco es determinado por el máximo tamaño del *MSDU* (2304 octetos).

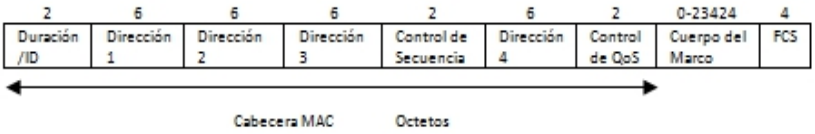

Figura 2.7: Formato de Marcos MAC

Tipos de marcos y mensajes del IEEE 802.11

Tres tipos de marcos *MAC (MPDUs)* atraviesan una *WLAN* (Red de Área Local Inalámbrica): control, datos, y manejo. Un marco del MAC tiene hasta cuatro, pero generalmente tres, campos de dirección. Cada campo de dirección es el mismo formato que una dirección *MAC* de IEEE 802. Se utilizan los cinco tipos de dirección siguientes:

- El identificador de *BSS (BSSID)* - Identifica el *AP* de una infraestructura *BSS*. Para un *IBSS* (red *ad hoc*) esto es un número al azar localmente-administrado.
- Dirección de destino (*DA*) - Identifica a receptores finales del marco.
- Dirección de fuente (*SA*) - Identifica la fuente inicial del marco.
- Dirección del receptor (*RA*) - Identifica al receptor inmediato *AP* en el *DS* inalámbrico.
- Dirección del transmisor (*TA*) - Identifica el *AP* que transmitió el marco sobre el *DS* inalámbrico.

2.2.8. Estándares IEEE 802.11 a/b/g/n

En la actualidad la mayoría de productos son de la especificación b o g , sin embargo ya se ha ratificado el estándar IEEE 802.11n que sobrepasa la velocidad de 600 Mbps. Actualmente ya existen varios productos que cumplen el estándar N con un máximo de 300 Mbps (80-100 estables).

La tabla 2.8 compara las características Wi-Fi en la evolución de los diversos estándares IEEE 802.11. El estándar IEEE 802.11n hace uso simultáneo de ambas bandas, 2,4 Ghz y 5,4 Ghz. Todas las versiones de 802.11xx, aportan la ventaja de ser compatibles entre sí, de

Parámetro Wi-Fi	Protocolos IEEE 802.11			
	802.11a	802.11b	802.11g	802.11n
Frecuencia de operación	5.3 GHz y 5.8 GHz	2.4 GHz		2.4 GHz o 5 GHz
Rango de señal promedio	~30 a 35 m			~60 a 70 m
Ancho de banda disponible por señal	~20 a 22 MHz			20 o 40 MHz
Tasa de transferencia (Max.)	54 Mbps	11 Mbps	54 Mbps	248 Mbps (2 fuentes)
Rendimiento típico para la máxima tasa de transferencia	18 a 22 Mbps	6 Mbps	18 a 22 Mbps	74 Mbps
Técnica de modulación	OFDM	CCK o DSSS	OFDM	OFDM usando MIMO y CB
Canales	36, 40, 44, 48, 52, 56, 60, 64, 149, 153, 157, 161	1-11		3 canales sin traslape en la banda de frecuencia ISM a 2.4 GHz 12 UNII canales en la banda de frecuencia de 5GHz sin traslape con o sin CB
Consideraciones especiales	Señales con frecuencia las alta tienen más problemas con obstrucciones físicas	2.4 GHz sujeto a interferencia con productos Bluetooth, teléfonos inalámbricos, microondas, radares, controles remotos, redes ZigBee, etc.		

Figura 2.8: Comparación de redes IEEE 802.11

forma que el usuario no necesitará nada más que su adaptador *WiFi* integrado para poder conectarse a la red. IEEE utiliza la técnica de modulación *OFDM* al igual que IEEE 802.11a e IEEE 802.11g. Presenta un mejor rendimiento y una velocidad bastante considerable en comparacion con sus antecesores según los datos descritos en la tabla.

Para el desarrollo del proyecto se trabaja con los estándares IEEE 802.11b e IEEE 802.11g.

2.3. IEEE 802.15.4 (WSN)

El estándar IEEE 802.15.4 fue desarrollado para Redes Inalámbricas de Área Personal *(WPANs - Wireless Personal Area Networks)*. Las *WPAN's* son redes inalámbricas de bajo costo y sin muchos requerimientos de rendimiento, donde se transporta información a cor-

tas distancias entre dispositivos. Las bondades principales de este tipo de redes son su facilidad de instalación, transferencia de datos confiable, costo extremadamente bajo y vida de batería razonable; manteniendo un protocolo simple y flexible [25] [26].

Las características principales del estándar son las siguientes:

- Tasas de transferencia de 250 Kb/s, 100 Kb/s, 40 Kb/s, y 20 Kb/s.
- Topología de red estrella o punto-a-punto.
- Direccionamiento corto de 16 bits o extendido de 64 bits.
- Asignación opcional de ranuras de tiempo garantizadas *(GTS - Guaranteed time slots)*.
- Protocolo de Acceso Múltiple con Detección de Portadora y Evasión de Colisiones *(CSMA-CA, Carrier Sense Multiple Access with Collision Avoidance)* para acceder al medio físico. El estándar IEEE 802.15.4 difiere del estándar IEEE 802.11 en que cuenta con 2 modalidades: ranurada (con *"beacons"*) y no ranurada (*"sin beacons"*) las cuales se explican en el apartado de acceso al canal.
- Bajo consumo de energía.
- Detección de energía (*ED*).
- Indicación de calidad de enlace (*LQI*).
- Un total de 49 canales disponibles, de los cuales 16 canales están en la banda de los 2450 MHz, 30 canales en la banda de los 915 MHz, y 3 canales en la banda de los 868 MHz.
- Redes complejas de más de 65534 dispositivos.

2.3.1. Componentes de una *WPAN* IEEE 802.15.4

Una *WPAN* se puede componer de 2 tipos de dispositivos: un dispositivo de función completa *(FFD, Full Function Device)* y un dispositivo de función reducida *(RFD, Reduced*

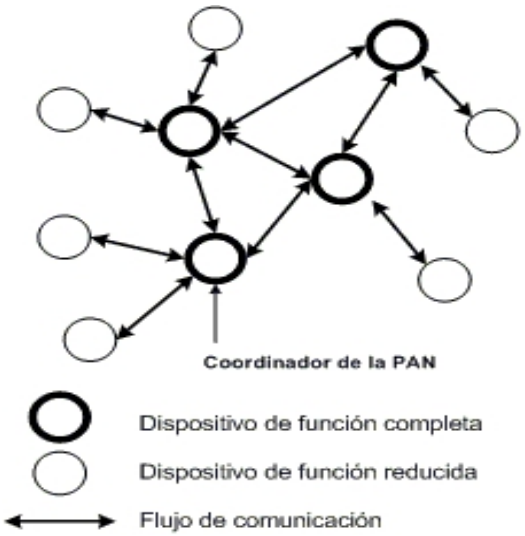

Figura 2.9: Dispositivos presentes en una LR-WPAN

Function Device), ver Figura 2.9. El *FFD* puede funcionar en uno de dos modos: como el coordinador (indispensable para una *PAN*) de la *PAN* (*Personal Area Network* ó Red de Área Personal) o como un dispositivo. Un *FFD* puede comunicarse con *RFD's* u otros *FFD's* además de poder iniciar y configurar una nueva *PAN* y solicitar la asignación de ranuras de tiempo garantizadas (*GTS's*) para la transmisión de datos, mientras que un *RFD* sólo puede comunicarse con un *FFD* y no tiene prestaciones para iniciar una *PAN* o solicitar la asignación de *GTS's*.

Particularmente un *RFD* está previsto para usos extremadamente simples, no puede ser coordinador de la *PAN* y puede asociarse solamente a un solo *FFD* a la vez. Por lo tanto, el *RFD* se puede poner en ejecución usando recursos y capacidades de memoria mínimos.

2.3.2. Arquitectura del IEEE 802.15.4

Éste estándar define una arquitectura basada en el modelo de referencia para la Interconexión de Sistemas Abiertos *(OSI, Open Systems Interconection)* [30]. Dicha arquitectura abarca la capa física (*PHY*), que contiene el transmisor-receptor de radiofrecuencia (*RF*) junto con su mecanismo de control de bajo nivel y una subcapa de control de acceso al medio

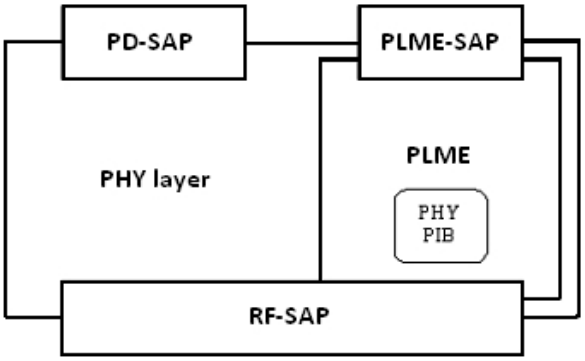

Figura 2.10: Modelo de referencia de la capa física

(*MAC*) que proporciona el acceso al canal físico para todas las transferencias. La Figura 2.11 muestra la arquitectura del IEEE 802.15.4.

Capa física (*PHY*)

La capa física define las características físicas y eléctricas de la red. La tarea básica de la capa física es la transmisión y recepción de datos. Las especificaciones para la sensibilidad de recepción y potencia de transmisión se encuentran en la capa física.

La figura 2.10 muestra el modelo de referencia de la capa física, el cual muestra los diferentes segmentos que la conforman. Tiene comunicación con capas adyacentes por medio de los puertos *PD-SAP* (*PayLoad Data - Service Access Point* ó Punto de acceso de servicio - Carga de datos), *RF-SAP* (Radio Frecuencia - *SAP*) y *PLME-SAP* (*Physical layer Management Entity - SAP*). Por un lado se comunica al exterior por medio del punto de acceso al servicio de radio frecuencia y a las capas superiores por medio de los puertos para la comunicación de datos y directivas de administracion.

La capa física es también responsable de las siguientes tareas:

- Habilitar/deshabilitar el transceptor de radio.
- Indicación de calidad de enlace (*LQI*) para paquetes recibidos.
- Detección de energía (*ED*) dentro del canal activo.
- Clear channel assessment (*CCA*).

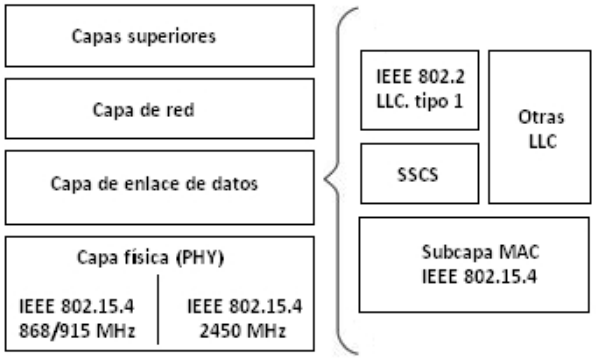

Figura 2.11: Relación del IEEE 802.15.4 con el modelo OSI

La subcapa de control de enlace lógico IEEE 802.2 (*LLC, Logical Link Control*) puede tener acceso a la subcapa *MAC* a través de la subcapa de convergencia de servicio específico (*SSCS, Service Specific Convergence Sublayer*). La *SSCS* existe conceptualmente sobre la parte común de la subcapa *MAC* del estándar IEEE 802.15.4. Esta subcapa proporciona una interfaz entre una instancia de la subcapa LLC del IEEE 802.2 y la subcapa *MAC* del IEEE 802.15.4. Por otra parte, la capa física del IEEE 802.15.4 puede operar en bandas de frecuencia y velocidades de transmisión de datos de 2450 MHz a 250 kb/seg, 915 MHz a 40 kb/seg y 868 MHz a 20 kb/seg con 16 canales en la banda de 2450 MHz, 10 canales en la banda de 915 MHz y 1 canal en la banda de 868 Mhz.

Capa *MAC*

La capa *MAC* define como múltiples dispositivos con IEEE 802.15.4 operando sobre la misma área, compartirán el espacio. Esto incluye la coordinación entre los transceptores al enlace de radio compartido y la calendarización y ruteo de marcos de datos.

Existen funciones embebidas de asociación y desasociación en la capa *MAC*. Esas funciones soportan la autoconfiguración y características *peer-to-peer* de una red con el estándar.

Subcapa *MAC*. La subcapa *MAC* IEEE 802.15.4 es responsable de brindar los siguientes servicios:

- Reconocimientos de entrega de trama (*ACK*): Éste servicio garantiza que la transmisión de datos sea confiable.

- Reinicio de la subcapa *MAC*: Servicio con el cual se pueden ajustar los valores almacenados en el *MAC-PIB* (La capa *MAC*, igual que la capa física, tiene sus propias constantes y atributos. Los atributos son almacenados en la *MAC PAN Information Base* ó Base de información de la PAN MAC a los valores establecidos por defecto, lo cual suele ser necesario antes de iniciar una PAN o antes de iniciar una tentativa de asociación.

- Exploraciones de canal: Este servicio es utilizado para detectar otros dispositivos o *PAN's* que se encuentren dentro del espacio de operación. Existen cuatro tipos de exploración: exploración para la detección de energía, exploración activa, exploración pasiva y exploración huérfana.

- Inicialización de una *PAN*: Este servicio es ofrecido, por un dispositivo de función completa (*FFD*), para iniciar las gestiones necesarias al iniciar una nueva PAN. Antes de iniciar una *PAN* será necesario realizar una exploración activa para reunir información necesaria para elegir un identificador de la *PAN* apropiado.

- Asociación: En base a los resultados de una exploración pasiva, un dispositivo podrá elegir una PAN con la cual asociarse enviando una solicitud al coordinador.

- Disociación: Este servicio hace referencia a la capacidad de todo dispositivo para abandonar una *PAN* o a la capacidad de todo coordinador para notificar a un dispositivo asociado abandonar la *PAN*.

- Transmisión de datos: Este servicio es ofrecido por todo dispositivo para poder enviar datos hacia otros dispositivos asociados a la *PAN*.

- Manejo de *GTS's*: Cuando un dispositivo *FFD* necesite un ancho de banda dedicado para la transmisión de datos, puede solicitar a su coordinador que le asigne una ranura de tiempo garantizado (*GTS*). De la misma forma un coordinador puede desasignar *GTS's* y notificar a un dispositivo dicha desasignación.

- Notificación de huérfanos y realineación de dispositivos: Este servicio es utilizado por los dispositivos que han quedado disociados de una *PAN* en calidad de huérfanos y que mediante una exploración huérfana solicitan la realineación a un coordinador. Cuando un coordinador recibe una petición de realineación por parte de un dispositivo, tendrá que buscar si existe un historial que avale la anterior asociación de dicho dispositivo para volver a asociar el dispositivo a su *PAN*.

- Seguridad: Este servicio es el encargado de ofrecer el control de acceso, la integridad, el cifrado de datos y la frescura secuencial según lo solicite la capa superior a la MAC de cualquier dispositivo.

- Información de la *PAN*: Este servicio otorga las capacidades suficientes para que un dispositivo (*FFD o RFD*) puedan escribir o leer información almacenada en la base de información de la *PAN* en la *MAC (MAC-PIB)*.

- Sincronización: Este servicio permite que todo dispositivo que esté funcionando en una PAN habilitada con *"superframes"* pueda solicitar la sincronización con el *"beacon"* o con el coordinador a través de la transmisión de datos pendientes en una *PAN* habilitada *sin "beacons"*.

La subcapa MAC incluye conceptualmente una entidad de manejo llamada *MLME (MAC Sublayer Management Entity)*. Esta entidad proporciona las interfaces de servicio a través de los cuales las funciones de administración de capa pueden ser invocadas. La *MLME* es también responsable de mantener una base de información de los objetos manejados que pertenecen a la subcapa *MAC*. Esta base de información se refiere como la *PIB* (*PAN Information Base* ó Base de información de la PAN) de la subcapa MAC.

La subcapa *MAC* proporciona dos tipos de servicios hacia las capas superiores, mismos que se acceden a través de dos Puntos de Acceso a Servicio *(SAP, Service Access Point)*, como se muestra en la Figura 2.12: el servicio de datos de la *MAC* ó *SAP* de datos accedido a través de la Parte Común de la Subcapa *MAC (MCPS, MAC Common Part Sublayer)* y

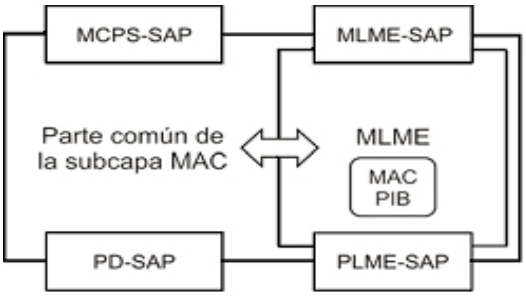

Figura 2.12: Modelo de la SubCapa *MAC*

el servicio de manejo de la *MAC*, accedido a través de la Entidad de Manejo de la subcapa *MAC (MLMESAP, MAC Sublayer Management Entity-SAP).*

La subcapa *MAC* tiene acceso a los servicios de la capa física (*PHY*) a través de dos puntos de acceso a servicios proporcionados por la *PHY*: el servicio de datos de *PHY*, accedido por el *SAP* de datos de *PHY (PD-SAP, PHY Data-SAP)* y el servicio de administración de *PHY*, a través del *SAP* de la entidad de manejo *(PLME-SAP, Physical Layer Management Entity-SAP).* Ver figura 2.12.

2.3.3. Estructura de las tramas *MAC*

El formato general de las tramas *MAC* se diseñó para ser muy flexible y que se ajustara a las necesidades de las diferentes aplicaciones con diversas topologías de red. La trama de la Unidad de Datos del Protocolo *MAC* (*MPDU, MAC Protocol Data Unit*), se compone de un encabezado *MAC (MHR, MAC Header)*, una Unidad de Servicio de Datos *MAC (MSDU, MAC Service Data Unit)* y termina con una verificación de la trama *(MFR, MAC Footer)*, ver Figura 2.13.

El campo *"Frame Control"* indica el tipo de trama a trasmitir, especifica el formato, el campo de dirección, controla los mensajes de reconocimiento ó *ACK's* y contiene un bit llamado *"Security enabled"* que indicará si la carga útil de la trama ha sido asegurada.

El campo *"Data Sequence Number"* contiene la verificación de la integridad de la trama MAC. El campo *"Address information"* puede contener información del remitente y/o

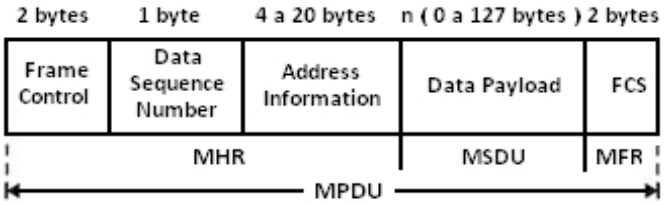

Figura 2.13: Formato general de la trama *MAC*

del destinatario. El campo *"Data Payload"* es la carga útil y su contenido dependerá del tipo de trama. El estándar define cuatro tipos diferentes de tramas MAC: trama de *"beacon"*, trama de datos, tramas *ACK* y tramas de comandos. Sólo las tramas de datos y de *"beacon"* contienen información que proviene de capas superiores; las tramas *ACK* y la de comandos de la *MAC* se originan en la propia subcapa *MAC*. El campo *FCS (Frame Check Sequency)* es un campo de verificación de 16 bits *CRC (Cyclic Redundancy Check)*.

2.3.4. Canales

De las 3 bandas de frecuencia en *ISM (Industrial, Scientific and Medical)*, solo la banda de 2.4 GHz opera alrededor del mundo. La banda de 868 MHz solo opera en EEUU y la banda de 915 MHz es solo para norte y Sudamérica.

Acceso al canal

Una *PAN* podrá ser configurada en una de dos formas dependiendo el mecanismo de acceso al medio físico que implemente: sin *"beacons"* o con *"beacons"*. Una red sin *"beacons"* utilizará el mecanismo *CSMA-CA* (en su versión no ranurada) es decir, implementará la contención para acceder al medio.

Estas redes trabajan de la siguiente forma. Cuando algún nodo desea transmitir, la red primero revisa si otro nodo se encuentra transmitiendo sobre el mismo canal. Si es el caso, el intento de acceso al canal se tiene que posponer, o finalmente indicar una falla de conexión después de varios intentos fallidos.

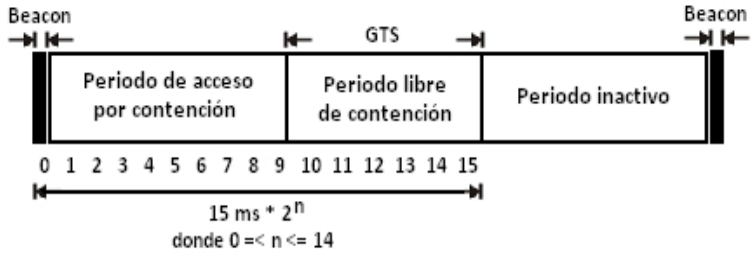

Figura 2.14: Estructura del *superframe*

La trama de *acknowledgment* confirma si una transmisión previa no utiliza los mecanismos de *CSMA* dado que estos se mandan inmediatamente después de cada paquete de información.

Por otro lado, una *PAN* configurada con *"beacons"* utilizará una estructura denominada *"superframe"*, mostrada en la Figura 2.14, la cual se implementa mediante una portadora dividida en intervalos de tiempo ofreciendo un Periodo de Acceso por Contención *(CAP, Contention Access Period)* y un Periodo Libre de Contención *(CFP, Contention Free Period)*. El *"superframe"* estará limitado por la transmisión de *"beacons"* en intervalos definidos y puede tener una porción activa y una porción inactiva.

- *Beacon* de red: Transmitido por el coordinador de la *PAN*. Contiene información de la red, la estructura del *"superframe"* y notificación de datos pendientes.
- Periodo de contencion: Acceso por cualquier dispositivo usando CSMA-CA ranurado.
- Ranuras de tiempo garantizado: Reservado para dispositivos que requieren ancho de banda garantizada.
- Periodo inactivo: Periodo en el cual un dispositivo puede pasar a un estado de bajo consumo de energia.

La porción activa se divide en 16 ranuras de tiempo, independientes a la duración de cada *"superframe"* y se compone de tres porciones: un *"beacon"*, el *CAP* y el *CFP*. Para to-

das las transmisiones dentro del *CAP* se utilizará el mecanismo *CSMA-CA* ranurado y en ese periodo serán transmitidos todos los comandos de la *MAC*. Por otra parte el *CFP* estará compuesto por todos los *GTS's* asignados por el coordinador de la *PAN* y ninguna transmisión dentro de ese periodo utilizará el mecanismo *CSMA-CA* para acceder al canal. Tanto el *"beacon"*, como los *ACK´s* serán transmitidos, sin el uso de *CSMA-CA*.

En una red con *beacons*, cualquier dispositivo que desee transmitir durante el periodo de acceso de contención, espera a que empiece la siguiente ranura de tiempo y después determina si algún otro dispositivo se encuentra transmitiendo en la misma ranura de tiempo. Si algún otro dispositivo se encuentra transmitiendo en esta, el dispositivo se repliega a un número aleatorio de ranuras o indica un fallo en la conexión después de varios intentos. Además en una red con *beacons*, las tramas de acknowledgment no utilizan *CSMA*.

Una función importante del *MAC* es la confirmación de recepciones exitosas de *frames* de algún dispositivo. Las recepciones exitosas y las validaciones de datos o comandos *MAC* se confirman por medio de mensajes de reconocimiento o *ACK´s*. Si el dispositivo de recepción no es capaz de recibir la información en ese momento por algún motivo, el receptor no manda ningún *ACK*. El campo de control en el frame indica si se espera un *ACK* o no. El frame que contiene al *ACK* se manda de regreso inmediatamente después de que se hace una validación exitosa del frame de entrada. Los marcos o paquetes mandados por el coordinador del *PAN* y los mensajes de *ACK´s* nunca son respondidos con algún *ACK*.

2.3.5. Servicios de seguridad

La arquitectura de seguridad del Modelo de Referencia OSI [29] define los siguientes servicios de seguridad para proteger las comunicaciones de los usuarios en las redes: autenticación verifica la supuesta identidad de un usuario o sistema; control de acceso protege los recursos del sistema contra su utilización no autorizada; confidencialidad protege los datos contra revelaciones no autorizadas; integridad protege los datos contra modificaciones no autorizadas; no rechazo protege contra el remitente de un mensaje que niega serlo o contra el receptor de un mensaje que niega haberlo recibido.

Servicios de seguridad definidos por el IEEE 802.15.4

En este estándar la subcapa *MAC* es la responsable de proporcionar los servicios de seguridad, cuando es requerido por las capas superiores. Las capas superiores proporcionan todo el material necesario para proporcionar los servicios de seguridad.

La administración de las llaves, la autenticación del dispositivo y la protección de refresco están fuera del alcance del estándar. Los mecanismos de seguridad en el estándar son mecanismos de llave simétrica. La seguridad proporcionada por estos mecanismos asume que las llaves han sido generadas, transmitidas y almacenadas de manera segura. Los servicios de seguridad definidos por el IEEE 802.15.4 son los siguientes:

- Control de acceso: Para proporcionar este servicio, un dispositivo mantendrá una lista de dispositivos de los cuales espere recibir tramas, la estructura es denominada Lista de Control de Acceso *(ACL, Acces Control List)* y está situada dentro del *MAC-PIB*.
- Cifrado de datos: Protege los datos contra la lectura de entidades no autorizadas.
- Integridad de la trama: Protege los datos contra manipulaciones no autorizadas, utiliza un código de integridad del mensaje.
- Refresco secuencial: Utilizado para rechazar las tramas repetidas y proporciona una evidencia de que los datos recibidos son más recientes que los últimos datos recibidos por un dispositivo, pero no proporcionan un sentido determinante del tiempo.

  Este estándar implementa los servicios de seguridad a través de siete mecanismos denominados *"Suites de seguridad"*. Para todas las *"Suites de seguridad"* el algoritmo de cifrado es el Estándar Avanzado de Cifrado *(AES)*. El nombre de cada *"Suite"* indica el modo de operación de *AES* y la longitud del código de integridad.

2.4. Comparación de los estándares IEEE 802.15.4 y IEEE 802.11

A continuación se realiza una comparación entre las tecnologías IEEE 802.15.4 e IEEE 802.11. Dado que los parámetros a considerar son muchos, es importante tener presente las características de cada uno de los estándares. Se comparan potencias de transmisión, tipos de modulación, canales, entre otros.

La tecnología con menor tasa de transmisión es la IEEE 802.15.4, la cual fue diseñada para ahorrar energía. Ambos estándares manejan diferentes velocidades, esto con el fin de adaptarse mejor a las necesidades del mercado.

Un parámetro importante es la potencia que consumen, donde observamos que la tecnología IEEE 802.15.4 consume mucho menos potencia que *WiFi*.

Otros parámetros importantes y que son necesarios considerar al momento de decidirnos por una tecnología (o rechazarla) son los parámetros que describen la complejidad de operación, los costos, interferencia y la compatibilidad con dispositivos extranjeros o que utilizan otras tecnologías.

La figura 2.15 muestra las principales características comparables entre IEEE 802.15.4 e IEEE 802.11.

Gracias a que entre los objetivos de los grupos de trabajo de la IEEE que desarrollaron estos estándares, estaba el ahorro de energía y el ahorro de inversión en licencia para transmitir sobre canales de *RF*, podemos contar con estándares que trabajan sobre bandas libres y por lo tanto no tenemos que pagar para utilizar canales de *RF*. Sin embargo el estándar IEEE 802.15.4 es mucho más simple que cualquiera de los demás estándares y por lo tanto es más fácil de implementar.

El estándar IEEE 802.11 con sus tres variables más importantes que trabajan en la banda de los 2.4 GHz. transmite con potencias muy elevadas con las que no sería apropiado mantener una *WPAN* (*Wireless Personal Area Network* ó Red de Área Personal). Más bien estas tecnologías son complementarias a las *WPAN's* ya que es posible conectar una *WPAN*

Parámetro Wi-Fi	Protocolos IEEE 802.11				IEEE 802.15.4
	802.11a	802.11b	802.11g	802.11n	
Frecuencia de operación	5.3 GHz y 5.8 GHz	2.4 GHz		2.4 GHz o 5 GHz	2.4 GHz
Rango de señal promedio	~30 a 35 m			~60 a 70 m	10 a 20 m
Ancho de banda disponible por señal	~20 a 22 MHz			20 o 40 MHz	5 MHz
Tasa de transferencia (Max.)	54 Mbps	11 Mbps	54 Mbps	248 Mbps (2 fuentes)	20, 40 o 250 Kbps
Rendimiento típico para la máxima tasa de transferencia	18 a 22 Mbps	6 Mbps	18 a 22 Mbps	74 Mbps	
Técnica de modulación	OFDM	CCK o DSSS	OFDM	OFDM usando MIMO y CB	BPSK
Canales	36, 40, 44, 48, 52, 56, 60, 64, 149, 153, 157, 161	1-11		3 canales sin traslape en la banda de frecuencia ISM a 2.4 GHz 12 UNII canales en la banda de frecuencia de 5GHz sin traslape con o sin CB	49 canales disponibles, de los cuales 16 canales están en la banda de los 2450 MHz, 30 canales en la banda de los 915 MHz, y 3 canales en la banda de los 868 MHz
Consideraciones especiales	Señales con frecuencia las alta tienen más problemas con obstrucciones físicas	2.4 GHz sujeto a interferencia con productos Bluetooth, teléfonos inalámbricos, microondas, radares, controles remotos, redes ZigBee, etc.			2.4 GHz sujeto a interferencia con productos Bluetooth, teléfonos inalámbricos, microondas, radares, controles remotos, redes ZigBee, etc.

Figura 2.15: Comparación de los estándares IEEE 802.11 e IEEE 802.15.4

a una *WLAN* (*Wireless Local Area Network ó Red de Área Local Inalámbrica*) por medio de una interfaz apropiada.

2.5. Resumen

Se describieron los estándares utilizados en el desarrollo del proyecto, el IEEE 802.11 y el IEEE 802.15.4, iniciando por una descripción general de ambos estándares, primeramente el estándar IEEE 802.11 y posteriormente el IEEE 802.15.4. Ventajas, aplicaciones en sistemas embebidos que son el tipo de dispositivos que nos interesa debido al proyecto, así como la arquitectura del estándar, seguridad, entre otros. Finalmente se hace una comparación de ambos estandares donde se muestran las ventajas del IEEE 802.15.4 que si bien es un estándar que cuenta con muy pocos recursos en cuanto a velocidad, capacidad de transmisión, entre otros, en comparación con *WiFi*, ofrece también la ventaja de requerir mucha menos energía que este último, permitiendo desplegar una amplia red de dispositivos capaces de captar múltiples datos. Se muestran también las bondades del estándar IEEE 802.11, un estándar ampliamente conocido y desplegado en una infinidad de aplicaciones de la vida cotidiana y mucho más complejas. Teniendo un panorama de estos dos estándares, tenemos la claridad que necesitamos para afirmar que el enlace entre ambos es una mezcla poderosa en cuanto a alcance y capacidad de manejo de información vital.

Capítulo 3

Puentes

3.1. Introducción

La transmisión de información ha sido siempre un importante tema dentro del área de sistemas digitales, siempre se mantiene la constante búsqueda de formas de poder llevar la información de un punto a otro de forma rápida, económica y transmitiendo la mayor cantidad de información posible en un determinado espacio de tiempo, sin embargo la transmisión de información entre diferentes esquemas o tecnologías de red es crucial, ya que la heterogeneidad de redes crece de una manera acelerada y la interconexión de redes de diferente tipo se hace indispensable.

De acuerdo al estándar ISO (*Open Systems Interconnection* - Interconexión de sistemas abiertos), dos o más subredes son interconectadas usando equipos llamados sistemas intermediarios, los cuales su función primaria es transmitir información selectiva de una subred a otra y realizar conversión de protocolos donde sea necesario. Un puente provee el medio para la interconexión de dos redes físicamente diferentes, las cuales difieren en las primeras dos capas del modelo OSI (Física y Enlace de Datos) [6].

3.2. Definición

Un puente [27] [4] es un dispositivo que comunica redes de distinta naturaleza (topologías y protocolos), por lo que opera a nivel 2 de la capa de enlace de datos (Nivel 2 del modelo de referencia OSI). Por encima, completamente ignorado queda el nivel 3 (de

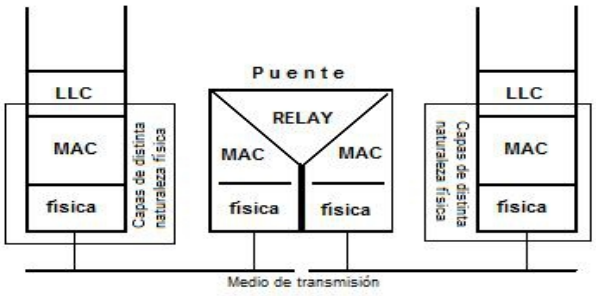

Figura 3.1: Estructura básica de un puente

protocolo de red), por lo que no importa que los protocolos usados sean distintos (ej. IP e IPX). El uso típico que se le ha dado en el pasado a los puentes es para conectar una *ethernet* (IEEE 802.3) con una *Token Ring* (IEEE 802.4).

Su función principal es la adaptación de la información de una red a otra, para que sea compatible entre ellas. Es capaz de analizar y filtrar información, limitando ó controlando el tráfico en una misma red. De esta forma, se reduce el tráfico de la red.

Cuando las subredes difieren en protocolos de sus capas más altas, especialmente en la capa de aplicación, o cuando las funciones de comunicación de las ultimas 3 capas no son suficientes para el acoplamiento, el sistema intermediario llamado en este caso *Gateway*, contiene todas las capas de las redes implicadas y convierte mensajes de aplicación entre los formatos apropiados [6]. La figura 3.1 muestra la representación básica de un puente. En esta figura se observa que el puente interconecta 2 redes IEEE 802 transmitiendo y filtrando marcos entre cada una de las *MAC´s* de dichas redes.

La generalizada interconexión entre redes provoca la necesidad de contar con sistemas que permitan dicha conexión, es por eso que se requiere del desarrollo de dispositivos que permitan estas conexiones como lo es el caso del desarrollo del presente proyecto: puente WiFi-WSN.

3.3. Operación de un Puente

Los elementos principales de la operación de un Puente son los siguientes:

- Retransmisión y filtrado de marcos.
- Mantenimiento de la información requerida para toma de decisiones de filtrado y retransmisión de marcos.
- Mantenimiento de lo anterior.

3.4. Retransmisión

Un Puente entrega marcos de datos de usuario individuales entre dispositivos conectados a sus puertos. Las funciones que soportan la retransmisión de marcos y mantenimiento en la calidad de servicio son las siguientes:

- Recepción de marcos.
- Descarte de marcos recibidos con error.
- Descarte de marcos si el tipo de marco no es un marco de datos de usuario.
- Regeneración de prioridad de usuario.
- Descarte de marcos para suprimir ciclos en la topología física de la red.
- Descarte de marcos en el excedente de medida del servicio de unidad de datos.
- Reenvio de marcos transmitidos a otros puertos del puente.
- Selección de clases de tráfico, siguiendo la aplicación de filtrado de información.
- Colas de espera de marcos por clases de trafico.
- Descarte de marcos para asegurar que no se exceda el máximo de retardo.
- Selección de paquetes en cola para transmisión.
- Selección de prioridad de acceso.

- Mapeo del servicio de unidades de datos y recalculo de la secuencia de Chequeo de Marcos.
- Transmisión de marcos.

3.5. Filtrado y entrega de información

Un puente filtra marcos. Por ejemplo un puente no debería de enviar marcos recibidos de un puerto de un puente a otro puerto de ese mismo puente, de esta forma se prevendría la duplicación de marcos y también permite el control administrativo de los recursos de la red. Las funciones que soportan el uso y mantenimiento de información para este propósito son las siguientes:

- Cálculo distribuido y configuración del estado del puerto para cada puerto en la red.
- Configuración administrativa del *MAC* o del estado del puerto del puente.
- Configuración administrativa de los parámetros de protocolos en árbol (son protocolos de capa 2 que exploran constantemente la red, de forma que cualquier fallo o adición en un enlace, *switch* ó *bridge* es detectado al instante). Un puente también filtra marcos para reducir el tráfico en algunas partes de la red entre la fuente y el destino. Las funciones que permiten el uso y mantenimiento de información para este propósito son las siguientes:
  - Configuración permanente de direcciones reservadas.
  - Configuración explícita de filtrado de información estático.
  - Aprendizaje automático de filtrado de información dinámico.
  - Adición y remoción de filtrado dinámico de la información.

Un marco apresura la transmisión de marcos generados por servicios críticos o sensitivos a tiempo. La función que suporta el uso y mantenimiento de información para este propósito

es la siguiente: Configuración explicita de información de clase de tráfico asociada con los puertos del puente.

3.6. Arquitectura del Puente

Un Puente es modelado abarcando lo siguiente:

- Una entidad *MAC* de retransmisión que interconecte los puertos del puente.
- Al menos dos puertos.
- Entidades de capas superiores.

La entidad de retransmisión de *MAC* maneja funciones independientes de métodos de acceso al medio para retransmisión de marcos entre puertos del puente, filtrado de paquetes, y aprendizaje de filtrado de información. Este usa el servicio de subcapa interna proveída por entidades *MAC* separadas de cada puerto.

Cada puerto del puente transmite y recibe marcos a y de la red a la cual está unido. Una entidad *MAC* individual permanentemente asociada con el puerto, provee el servicio de subcapa interno usado para la transmisión y recepción de marcos. La entidad *MAC* maneja todas las funciones dependientes de métodos de acceso al medio.

La figura 3.2 muestra un ejemplo de la topología física de una red de área local. Las *LAN's* (*Local Area Networks* ó Redes de Área Local) están interconectadas por puentes *MAC*, donde cada puerto de un puente se conecta a una *LAN* simple.

3.7. Recepción de marcos

La entidad *MAC* individual de cada puerto del Puente examina todos los marcos transmitidos en la Red de Area Local (*LAN - Local Area Network* por sus siglas en inglés).

Los marcos con bits erroneos son descartados por la entidad *MAC*.

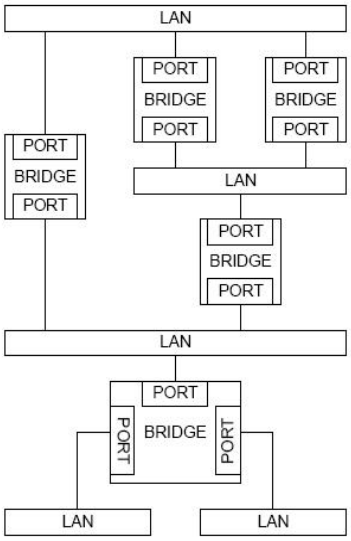

Figura 3.2: Red de área local puenteada

3.8. Transmisión de marcos

La entidad *MAC* individual para cada puerto del Puente transmite marcos recibidos por la entidad de retransmision *MAC*.

Los marcos retransmitidos son sometidos a transmisión por el proceso de reenvio.

Cada marco es transmitido siguiendo los procedimientos *MAC* de la tecnología de red específica, por ejemplo IEEE 802.3 e IEEE 802.4.

En este trabajo los marcos se transmitirán utilizando IEEE 802.15.4 e IEEE 802.11b.

3.9. Administración del puente

El puente cuenta con una serie de funciones que permiten su funcionamiento y manejo. Las funciones de administración del puente son suministradas por el Control de acceso al medio (*MAC*), de acuerdo con los conceptos y principios del marco de trabajo de OSI.

En cuando al puente WiFi-WSN, toda la administración se lleva a cabo dentro del módulo *Rabbit*, los elementos que permiten esta administración se describen en el capitulo 5 (Implementación).

3.9.1. Funciones de administración

Las funciones de administración consisten en brindar facilidades que soporten la planeación, organización, supervisión, control, protección y seguridad de los recursos de comunicación. Esas funciones pueden ser categorizadas como configuración, fallas, rendimiento, seguridad y administración de seguridad. Se describen a continuación:

- Administración de la configuración: permite la identificación de recursos de comunicación, inicialización y reconfiguración. Provee además parámetros operacionales, así como el establecimiento y descubrimiento de relaciones entre recursos. Las funciones de la administración de la configuración son las siguientes:
  - La identificación de todos los puentes que juntos forman una Red de Área Local Puenteada y sus respectivas locaciones.
  - La habilidad para reinicializar de forma remota puentes específicos.
  - La habilidad para controlar la prioridad de que puentes transmiten marcos en determinado momento.
  - La habilidad para forzar una configuración especifica.
  - La habilidad para controlar la propagación de marcos con un grupo MAC específico a ciertas partes de la Red de Área Local Puenteada.
- Administración de Fallas: permite la prevención de fallas, detección, diagnóstico y corrección. La habilidad para identificar y corregir el malfuncionamiento del puente, incluyendo errores de registro y reportes.
- Administración del rendimiento: permite la evaluación del comportamiento de los recursos de comunicación y de la efectividad de las actividades de comunicación; la habilidad de obtener estadísticas y análisis del rendimiento y tráfico del puente. Métricas específicas incluyen: utilización de la red, reenvio de marcos y descarte de marcos para puertos individuales.

- Administración de la seguridad: Permite la protección de recursos.

3.10. Identificación única del puente

Una dirección *MAC* única de 48 bits debe ser asignada a cada Puente. La dirección del puente puede ser la dirección *MAC* de un puerto del puente, donde se recomienda que sea el número mas bajo de la numeraciones de los puertos del puente (ej. Puerto 1).

3.11. Puentes WiFi-WSN

Existe una incontable cantidad de puentes de diferente tipo, sin embargo en el área de redes de de sensores, el desarrollo de puentes que permitan interconectar éste tipo de redes a otra tecnología inalámbrica es mínimo. Uno de los trabajos finalizados de este tipo, que es el que más se apega al proyecto presentado en ésta tesis y del cual se tomaron algunos conceptos de diseño y desarrollo es el que se presenta a continuación.

El puente fué desarrollado en la Universidad Autónoma de Baja California, Unidad Tijuana en el año 2006 [8].

Éste puente interconecta el estándar IEEE 802.11b y una red de sensores inalámbrica. El puente está compuesto principalmente por un microcontrolador PIC18F452, una tarjeta de red de computadora tipo PCMCIA IEEE 802.11b y un puerto RS-232 para la conexión de sensores u otros módulos. En el microcontrolador se implementó software embebido (*firmware*) para el manejo de la tarjeta *WiFi* y los protocolos estándares *IP y UDP*. Éste software también maneja un protocolo simple sobre el puerto serie para la comunicación con un módulo sensor. Como una muestra de su funcionalidad se presenta una aplicación de prueba donde un asistente personal digital (*PDA*) se conecta al puente para tomar información de su sensor.

El dispositivo prototipo está conformado por una parte de la circuitería (hardware) que contiene los dispositivos necesarios para llevar acabo la interfaz al medio físico y por otra parte la conforma el programa embebido (firmware) que se encarga de manejar a las señales de control y manejo de protocolos.

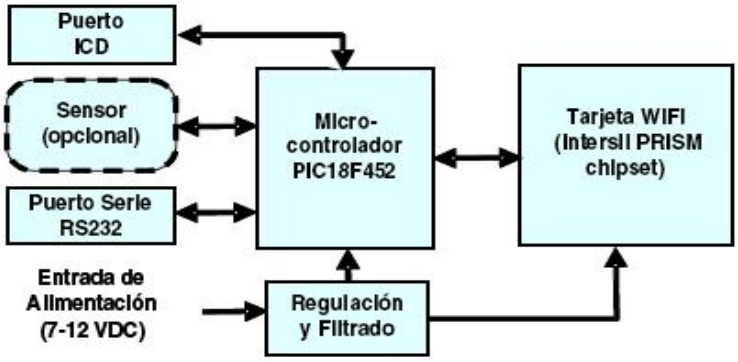

Figura 3.3: Diagrama a bloques del puente

3.11.1. Circuitería (Hardware)

Las principales características definidas en el diseño del puente son:

- a) El bajo consumo de potencia.
- b) Una interfaz de software para abstraer la complejidad de protocolos y manejo de la tarjeta *PCMCIA (Personal Computer Memory Card International Association)*.
- c) Que sea adaptable hardware/software en su interfaz para diferentes redes de sensores.

La 3.3 muestra un diagrama a bloques del puente diseñado en el cual se pueden ver los diferentes bloques que lo componen y que se describen de forma breve a continuación.

El microcontrolador del puente es un PIC18F452 de la compañía Microchip, es de bajo consumo de energía y de arquitectura tipo RISC de hasta 10 MIPS de velocidad. Sin embargo tiene limitaciones de memoria puesto que tiene 32 Kbytes de memoria de programa tipo flash, y solo 1536 bytes de SRAM. El microcontrolador es la parte central del puente, opera a 2 MIPS con un resonador de 8 MHz, éste se dedica a la operación de la tarjeta de red inalámbrica, enviando y recibiendo paquetes, codificado acorde a los protocolos implementados, así como llevar acabo el proceso de recepción y transmisión del puerto serie.

Este microcontrolador posee un conjunto de 3 líneas especiales que conforman un puerto *ICD (In-Circuit Debugger)* para su depuración y reprogramación.

Además contiene internamente una *EEPROM (Electrically-Erasable Programmable Read-Only Memory)* la cual se utiliza para almacenar parámetros de configuración del puente. Estos parámetros son la dirección IP asignada, el modo de operación (Infraestructura o *Ad-hoc*) y el nombre de identificación de la red inalámbrica *SSID (Service Set IDentifier)*. El bloque del circuito regulador y filtrado, mantiene un voltaje estable y libre de ruido para la buena operación del sistema. Consta de un circuito integrado LM1086CT-5.0 que es un regulador de voltaje de 5V de alta eficiencia y un conjunto capacitores que operan como filtro. La tarjeta de red inalámbrica es cualquier PCMCIA tipo II que use el *chipset* PRISM de la compañía Intersil. El puente utiliza una tarjeta Netgear MA401. Ésta tarjeta tiene la función de conectar el puente al medio físico inalámbricamente bajo el estándar IEEE 802.11b.

3.11.2. Firmware (software embebido)

El *firmware* en el microcontrolador lleva a cabo las operaciones de bajo nivel a la tarjeta de red inalámbrica. Estas operaciones son la escritura y lectura acorde al estándar PCMCIA Tipo II y a los registros de inicialización y operación del *chipset* PRISM de la compañía Intersil.

El firmware fue implementado en lenguaje *ANSI C* y está estructurado en dos módulos principales. La codificación está organizada de manera que cada uno de los módulos está por separado e independiente para facilitar la depuración e incorporación de módulos nuevos a la estructura. El primer módulo tiene la finalidad de abstraer operación de lectura/escritura de bajo nivel a la tarjeta. El otro módulo lleva acabo la codificación y decodificación de paquetes para el manejo de los protocolos IP (*Internet Protocol*) y UDP (*User Datagram Protocol*) que son enviados o recibidos por la tarjeta *WiFi*. Solo se han implementado dichos protocolos debido a la limitante de memoria SRAM del microcontrolador.

Éste módulo abstrae el uso de los protocolos en forma de Sockets estándar tipo Berkeley y la implementación se basa en los *RFC's* de los protocolos IP y UDP. Una herramienta de soporte utilizada en el desarrollo del firmware fue el paquete *Ethereal* que es un analizador de protocolos y fue utilizado para la validación de los paquetes codificados por el mi-

Figura 3.4: Puente funcionando

crocontrolador.

3.11.3. Aplicación de prueba

Antes del desarrollo de la aplicación se realizaron pruebas de verificación y depuración de los paquetes IP y UDP, esto mediante una computadora con tarjeta de red inalámbrica y con el soporte del programa *Ethereal* confirmando la información obtenida de los *RFC's* correspondientes a estos protocolos.

Posteriormente se utilizó una *PDA* con tarjeta interna IEEE 802.11b validando nuevamente y enlazándose al puente. Para fines prácticos se le conectó un sensor de temperatura y humedad STH11 de la compañía Sensirion a través del puerto RS-232, quedando abierto para incorporarle cualquier otro tipo de sensor.

El dispositivo puente fué cargado con firmware que funciona como servidor UDP, y constantemente está escuchando y analizando peticiones en paquetes tipo UDP para darles servicio si se requiere. (ver Figura 3.4).

En este caso experimental las peticiones están relacionadas con el valor de temperatura de humedad del sensor que ha sido incluido.

La *PDA* ejecuta una aplicación desarrollada en lenguaje Java la cual conecta a la *PDA* en forma inalámbrica al puente mediante un socket tipo UDP. Se seleccionó Java debido a que las aplicaciones implementadas pueden ser portadas a otras plataformas con un míni-

mo o ningún esfuerzo, incluso la aplicación realizada trabaja en Linux o Windows indistintamente. Una vez conectada la *PDA* realiza peticiones de los valores de temperatura y humedad para ser desplegados en tiempo real en pantalla.

El prototipo es indispensable en aplicaciones donde se requiere enlazar ambas redes y manipular la información a través de dispositivos inalámbricos estándar sin necesidad de adquirir productos de marca propietaria. El diseño funciona acorde a las necesidades, se ha validado y verificado en la comunicación con dispositivos utilizando el protocolo estándar UDP. El puente actual es una versión de prueba debido al tamaño, aunque ideal para experimentos de campo e investigaciones; pero se desea compactarlo y seguir trabajando para mejorarlo.

El *firmware* desarrollado ha sido estructurado de forma que los cambios en este pueden llevarse acabo fácilmente, sin embargo existen algunos puntos que deben ser mejorados en el futuro, ésto en respuesta a mediciones y análisis de posibles aplicaciones.

Es necesario bajar el consumo de potencia, debido a que la tarjeta PCMCIA opera a 5V y en ocasiones puede llegar a consumir hasta 400 mA. Para lograr esto se desea emigrar al estándar *CF (Compact Flash)* que opera a bajo voltaje de 3.3V, voltaje en el que el microcontrolador puede también operar. Además, el protocolo de comunicación UDP es insuficiente para algunas aplicaciones y en un futuro se trabajará en la implementación del protocolo TCP (Transmission Control Protocol) para conexiones más robustas.

También se planea en un futuro cercano adaptarle una cubierta a prueba de agua y el diseño de una fuente de alimentación híbrida (baterías y celda solar) para las aplicaciones a la intemperie y finalmente poder ser utilizado de forma cotidiana.

El proyecto puente WiFi-WSN ofrece varias ventajas con respecto a este desarrollado en la UABC-Tijuana, principalmente en cuanto a diseño, prestaciones y confiabilidad.

El dispositivo *Rabbit 5600* es más robusto que el PIC18F452 y el entorno de hardware donde se implementa; además de contar con más memoría y ser mas veloz, cuenta con mas puertos de comunicación e implementa el protocolo TCP que es más seguro que el UDP que se utiliza en este proyecto.

El proyecto antes mencionado no implementa una red de sensores sin embargo tiene disponible el puerto serial al cual se puede acoplar un nodo coordinador de una WSN. El puente WiFi-WSN a diferencia de este ejemplo, implementa una red de sensores, y los datos obtenidos a través de esta es posible accesarlos por medio de *WiFi* y una interfaz diseñada para dicho fin que se encuentra en un servidor embebido disponible en el módulo Rabbit.

3.12. Resumen

La necesidad de cooperación entre un rango diverso de diferentes tecnologías de información es ampliamente aceptado. Existen fundamentalmente 2 caminos para lograr esta cooperación:

1. Definición de estándares y protocolos, procesos ó modelos que pueden ser incorporados directamente dentro de un sistema. 2. Permitir la definición de modelos autónomos independientes y construir puentes que permitan afinar las diferencias.

Idealmente se prefieren los estándares porque reducen la interconexión y mantenimiento del flujo de trabajo a largo plazo, sin embargo en la práctica un estándar universal que contemple todos los aspectos de un sistema es impráctico y es inevitable el uso de alternativas que permitan un flujo efectivo de información.

La cooperación entre sistemas incompatibles puede ser implementado modificando directamente los sistemas para lograr la compatibilidad. Sin embargo el desarrollo y uso de puentes es el método más utilizado.

La información descrita en éste capítulo permite primero definir de manera clara y precisa el concepto de un puente. Se describen los elementos principales que intervienen en la operación de un puente, la forma en que se retransmiten y entregan los marcos, el filtrado y entrega de información. Se describe también la arquitectura y administración de un puente y dentro de este último las funciones que permiten dicha administración. Finalmente se presenta un proyecto muy similar al desarrollado en cuanto a concepto y diseño.

Teniendo esto como base se tomaron muchos de los conceptos y elementos que forman un puente y se aplicaron al proyecto objeto de esta tesis.

Capítulo 4

Hardware y Software utilizado

4.1. Introducción

En esta parte se muestra la infraestructura utilizada para el desarrollo del proyecto de la presente tesis, conformada por los dispositivos físicos *Rabbit RCM5600W* y *PAN802154HAR00*, el software *Dynamic C*, *Freescale BeeKit* y *CodeWarrior*.

4.2. Seleccion de Hardware y Software

La selección de todos los componentes presentes en este proyecto se hizo con mucho cuidado debido a que el objetivo principal tiene un enfoque en el área de la salud, la cual es una de las más delicadas y en donde el más mínimo error en alguna lectura, transferencia de algun dato ó falla general del sistema podría ocasionar un daño irreversible en el paciente.

Es por eso que se cuidaron varios aspectos en la elección de los componentes para el desarrollo del proyecto. Los aspectos que se tomaron en cuenta son los siguientes:

- Rendimiento: Se requiere un rendimiento óptimo, libre de congestión y fallas por lo cual se eligió el dispositivo *Rabbit* que cuenta con la suficiente velocidad, capacidad y conectividad requerida.

- Confiabilidad: Se requiere de una transmisión confiable donde se tenga la seguridad de que se transferirá de forma fiel el 100 % de los datos recolectados por la red de sensores. El dispositivo implementa el protocolo *TCP* el cual es ampliamente reconocido

y utilizado en los sistemas de comunicación y siempre ha demostrado una excelente confiabilidad.

- Precio: El precio tanto del dispositivo Rabbit como de los sensores empleados en la *WSN* es mínimo en comparación con otros dispositivos similares y aunque si existen dispositivos de la misma clase mas económicos e incluso mas pequeños, sacrifican rendimiento y conectividad, lo cual no es conveniente para el proyecto.

Cabe mencionar que el puente prototipo funciona de dos modos: el modo de programación y el modo independiente. Inicialmente se trabajó con el dispositivo en modo de programación. Una vez desarrollada la programación necesaria se graba el software en la memoria flash del dispositivo rabbit para que de esta forma sea posible su funcionamiento en modo independiente, sin embargo este es un detalle que a futuro se podría optimizar ya que el modo independiente requiere de una conexión a corriente.

4.3. Rabbit RCM5600W

4.3.1. Introducción

EL módulo *MiniCore RCM5600W* [17] [18] provee un módulo compacto de tipo *mini PCI express* con funcionalidad *WiFi* ó IEEE 802.11 b/g el cual permite diseñar proyectos de bajo costo y bajo consumo de energía basados en *WiFi* para sistemas embebidos. La figura 4.1 muestra el módulo.

El *kit* de desarrollo *RCM5600W Deluxe Kit* incluye un completo sistema de desarrollo llamado *Dynamic C*. El *kit* de desarrollo también contiene una tarjeta de interfaz con conexión *USB* que permite evaluar el *RCM5600W* y una tarjeta de prototipos para el desarrollo de aplicaciones. Permite también escribir y probar software para los módulos *RCM5600W*, incluyendo aplicaciones *WiFi*. La figura 4.2 muestra el *kit* de desarrollo.

El *RCM5600W* tiene un procesador *Rabbit 5000* operando a más de 73.73 MHz, memoria *flash*, dos relojes (oscilador principal y reloj en tiempo real) y la circuitería necesaria para

Figura 4.1: Módulo *Rabbit MiniCore RCM5600W*

Figura 4.2: *Kit* de desarrollo *RCM5600W*

resetear y manejar el *Rabbit 5000*. El módulo se monta sobre una tarjeta madre por medio de un socket *mini PCI Express 52-pin*. El *RCM5600* recibe su energía (+3.3V) de la tarjeta madre sobre la cual se monta. El *RCM5600W* se puede conectar con otros dispositivos digitales *CMOS*-compatible a través de la tarjeta madre.

4.3.2. Características

- Tamaño: 1.20"x 2.00"x 0.40"(30 mm x 51 mm x 10 mm)
- Microprocesador: *Rabbit 5000* corriendo a 73.73 MHz
- Más de 35 líneas de E/S de propósito general, cada una de ellas configurables con más

de cuatro funciones alternadas.

- Líneas de E/S de 3.3 V .
- Seis puertos seriales *CMOS*-Compatible - Cuatro puertos son configurables como un *Clocked Serial Port (SPI)*, y dos puertos son configurables como puertos SDLC/HDLC.
- Transceptor *Airoha single-chip* IEEE 802.11b/g.
- El bus externo de E/S puede ser configurado para 8 líneas de datos, 8 líneas de direcciones (compartidas con líneas paralelas de E/S), así como lectura/escritura E/S.
- 1MB SRAM y 1MB de memoria *flash* serial.
- Reloj en tiempo real soportado por una batería.
- Supervisor *Watchdog*

4.3.3. Programación

El *RCM5600W* es programado a través de un conector *USB* en la tarjeta madre, usando un cable *USB*. Puede también ser programado remotamente usando el *Remote Program Update library* con *Dinamic C* v.10.54 o posterior.

4.3.4. Ventajas

- Rápida ejecución y programación utilizando ingeniería llamada *"ready-to-run/ready-to-program"*
- Precio competitivo comparado al *WLNG-EK-DP003* y otros.
- Programación en lenguaje C de fácil desarrollo y depuración.
- *Rabbit Field Utility* para descargar archivos .bin de *Dynamic C* compilados.

- Gran capacidad de memoria que permite programas con cientos o miles de líneas de código, así como suficiente capacidad de almacenamiento de datos.

4.4. Dynamic C

Dynamic C [14] es un sistema de desarrollo integrado para la escritura de software embebido. Es diseñado para el desarrollo con controladores *Rabbit* y otros controladores basados en microprocesadores *Rabbit*. Integra las siguientes funciones de desarrollo:

- Edición.
- Compilación .
- Enlace.
- Carga.
- Depuración.

Cuenta con un editor de texto donde los programas se desarrollan y pueden ser ejecutados y depurados interactivamente a nivel de código fuente o código máquina. Soporta también programación en lenguaje ensamblador, sin necesidad de dejar el sistema de desarrollo. Se pueden mezclar los lenguajes C y ensamblador.

Permite el uso de comandos *printf*, expresiones *watch*, *break-points* y *stack tracing*.

Provee extensiones al lenguaje C (como variables compartidas y protegidas) que soportan el desarrollo de sistemas embebidos en tiempo real; soporta también multitarea cooperativa y *preemptiva*.

Contiene muchas librerías de función en código fuente, esas librerías soportan programación en tiempo real, E/S a nivel máquina, y provee funciones estándar de cadena y matemáticas.

4.4.1. Velocidad

Dynamic C compila directamente a memoria. Las funciones y librerías son compiladas, enlazadas y descargadas. En una PC rápida, Dynamic C puede cargar 30,000 *bytes* de código en 5 segundos a una tasa de transferencia de 115,200 bps.

4.4.2. Mejoras y diferencias de Dynamic C

Dynamic C difiere de la tradicional programación en C corriendo en una PC o bajo UNIX ya que no es posible usar el lenguaje estándar C en un ambiente embebido sin hacer adaptaciones. El lenguaje C estándar hace muchas asunciones que no aplican en sistemas embebidos. Por ejemplo el lenguaje estándar C asume que un sistema operativo está presente y que un programa inicia con un entorno limpio, considerando que un sistema embebido puede tener memoria respaldada por una batería y puede conservar datos a través de ciertos ciclos. *Rabbit* ha extendido el lenguaje C a una amplia variedad de áreas.

Mejoras de Dynamic C

- *Dynamic C* 10.54 introduce la actualización remota de *firmware* para algunos tipos de tarjetas.
- *Function Chaining* permite que segmentos especiales de código sean embebidos dentro de una o más funciones. Cuando una función cadena se ejecuta, todos los segmentos ligados a esa cadena se ejecutan. Permite al software realizar inicialización, recuperación de datos u otro tipo de tareas.
- *Costatements* permite simular procesos cooperativos y paralelos.
- *Cofunctions* permite que procesos cooperativos sean simulados en un solo programa.
- Soporta código ensamblador embebido y código ensamblador *stand-alone*.

- *Dynamic C* tiene palabras clave que ayudan a proteger datos compartidos entre diferentes contextos (compartidos) o almacenados en memoria *battery-backed* (protegidos).

- Tiene características que permiten al programador utilizar el máximo de memoria *(xmem: extended memory). Dynamic C* es cuidadoso en el manejo de memoria, pero existen instancias donde el programador querrá tener el control de ésta. *Dynamic C* tiene palabras clave y directivas para ayudar a poner el código y datos en el lugar apropiado como: *root, xmem,* y *·memmap* para código y *far* para datos.

Diferencias de Dynamic C

Si una variable es explícitamente inicializada en una declaración (ej. Int x = 0;), esta es almacenada en memoria *flash (EEPROM)* y no puede ser cambiada por una sentencia de asignación. Así, una declaración puede generar un *warning* que puede ser suprimido usando la palabra *const: const int x = 0;* . Para inicializar variables estáticas en RAM estático (SRAM) debemos usar secciones *#GLOBAL_INIT*. Otros compiladores de C automáticamente inicializaran todas las variables estáticas a cero que no son explícitamente inicializadas antes de entrar a la función principal. Los programas en Dynamic C no hacen esto porque en un sistema embebido podríamos desear preservar los datos en memoria RAM.

Los numerosos archivos include encontrados en programas típicos de C no son usados porque *Dynamic C* tiene un sistema de librerías que automáticamente provee prototipos de función e información de conector similar al compilador antes que el programa de usuario sea compilado. Esto es realizado mediante la directiva *#use*.

4.5. PAN802154HAR00

El módulo *Panasonic PAN802154* [20] es un dispositivo de comunicación de baja tasa de transferencia y bajo consumo de energía, basado en la plataforma de desarrollo de *freescale ZigBee* llamada *SARD (Sensor Application Reference Design)*. Ésta opera en la banda ISM

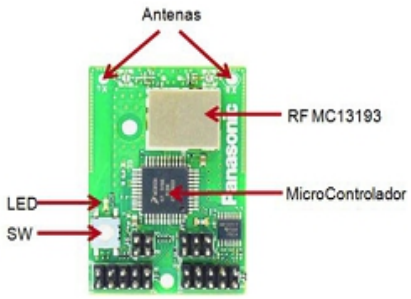

Figura 4.3: Módulo *PAN802154*

2.4 GHz y cumple totalmente con el estándar IEEE 802.15.4. El módulo *PAN802154* puede ser programado con las capas física y *MAC* del estándar, así como la capa de protocolo de *ZigBee*.

El módulo incorpora el transceptor *MC13193* [21], el microcontrolador *GT60*, el cual cuenta con un microprocesador de 8 bits, Memoria *flash* de 60 kb, 4Kb de memoria RAM y un convertidor Analógico-Digital. Cuenta también con una licencia para utilizar el software de la pila de protocolo *ZigBee* de *Freescale*. Y tiene una interfaz *RS-232 IC* y dos antenas impresas sobre la placa.

El módulo *PAN802154* cumple totalmente con los requerimientos actuales para la banda *ISM* 2.4 GHz.

La figura 4.3 muestra una imagen real del módulo y sus principales componentes.

Las características [19] del dispositivo son:

- Soporte completo *ZigBee*, IEEE 802.15.4 o aplicaciones *Simple MAC*.
- Banda de 2.4GHz *ISM*.
- 16 canales, espaciados a 5MHz,
- Tasa de transferencia de más de 250kbps.
- Puerto RS-232; 2 entradas analógicas seleccionables a un convertidor de 10bit AD.
- 8 puertos digitales de ES.

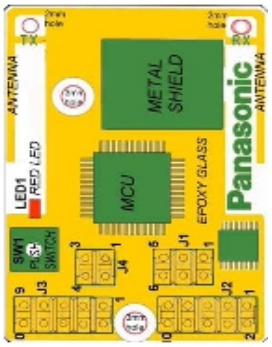

Figura 4.4: Conectores del módulo *PAN802154*

- 1 *Switch* y 1 LED para control y monitoreo.
- Salida nominal de potencia: 0dBm.
- Sensibilidad de recepción: -92dBm a tasa de pérdida de paquetes de 1.0
  - 2.2 VDC a 3.4 VDC sin uso de RS-232.
  - 3.0 VDC a 3.4 VDC usando RS-232.

4.5.1. Componentes y descripciones

La figura 4.4 muestra un diagrama donde se observan todos los conectores con los que cuenta el módulo *PAN802154*.

J1: Conector de programación/Depurador.

J2: Conector de puerto serial El conector J2 es una conector de puerto serial para la conexión a un puerto serial de PC. El conector contiene una línea de transmisión a nivel RS-232, una línea de recepción de datos y una línea *GND*.

J3: Conector de propósito general El conector J3 es usado como conector de propósito general de entrada/salida. Un pin de energía y otro GND permiten la conexión a componentes externos.

J4: Conector Power/GND El conector J4 es el conector de energía para el *PAN802154*.

4.6. BeeKit

BeeKit [22] es una herramienta de configuración de software usada para producir proyectos completos para CodeWarrior listos para ser importados por el ambiente de desarrollo de *CodeWarrior (CodeWarrior Development Studio Environment).*

BeeKit incluye múltiples plantillas de aplicaciones *(SMAC - Simple Media Access Control, MAC* - estándar IEEE 802.15.4 y *BeeStack* que es la solución que sirve como punto de partida para el desarrollo de aplicaciones del dispositivo. Los proyectos demo incluidos son aplicaciones totalmente funcionales. Esos proyectos pueden ser desplegados directamente al dispositivo sin necesidad de ninguna modificación.

La interfaz de usuario de *BeeKit* extrae su información del código base que contiene toda la información necesaria para construir una aplicación de dispositivo (componentes de software, código fuente, librerías, definiciones de plataforma de hardware, directivas de enlace y compilación, entre otros.) *BeeKit* se compone de las siguientes partes que en conjunto forman la base para el desarrollo de aplicaciones:

- *Codebase*: Es un repositorio de código fuente, archivos de configuración y de generación de reglas. De este repositorio los demos, plantillas y otras aplicaciones son generadas.
- Solución: Es un grupo de proyectos que están ligados a un directorio en específico a través de un archivo. Cada proyecto generara sus propios directorios.
- Proyecto: Es un grupo específico de archivos que son exportados desde BeeKit, para crear un proyecto de *CodeWarrior* en forma de un archivo XML.
- Archivo XML: *BeeKit* genera un archivo XML listo para ser importado en *CodeWarrior* en el que se contiene todo lo necesario para que *CodeWarrior* cree el proyecto.

4.7. CodeWarrior

CodeWarrior [23] [24] *Development Studio* es un completo Ambiente de Desarrollo Integrado (IDE) que provee un entorno de trabajo altamente visual y automatizado y permite acelerar el desarrollo de aplicaciones embebidas complejas.

CodeWarrior permite crear uno de los códigos más optimizados en el mercado gracias a los compiladores ANSI C/C++ y*Compact* C++. Estos compiladores tienen gran ventaja al trabajar con arquitecturas HCS08 o HCS12 con más de 60 estrategias de optimización avanzada especificamente diseñadas para maximizar el rendimieno y reducir el tamaño de código. De esta forma se aprovechan al máximo las capacidades y se reducen los costos.

CodeWarrior cuenta con un simulador, que provee una herramienta poderosa para el desarrollo de prototipos en el cual podemos detectar y reparar errores, siguiendo paso a paso el comportamiento en tiempo real de nuestros programas.

4.8. Resumen

En este capítulo se analizan a detalle dos componentes principales en la implementación del puente:

1. El módulo *Rabbit RCM5600W* y toda la tecnología que esta alrededor de este, software de desarrollo, módulos web, programación, ventajas, desventajas, características que lo definen como un dispositivo poderoso con el cual se pueden desarrollar prototipos con una rapidez notable y a un costo accesible.

2. El módulo *PAN802154HAR00*, así como su ambiente de desarrollo el cual incluye un módulo IDE de desarrollo: *BeeKit* el cual es una herramienta de configuración de software a base de plantillas que son importadas al ambiente de desarrollo de *CodeWarrior*, en el cual sobre una base establecida, permite el diseño de aplicaciones totalmente funcionales.

Primeramente se trabajó en el área de sensores utilizando las herramientas de desarrollo *Bee Kit* y *Code Warrior* para programar los módulos *PAN802154HAR00*, se utilizaron plantillas prediseñadas en base al estándar IEEE 802.15.4, las cuales cuentan con la fun-

cionalidad básica para el despliegue de una red de sensores, posteriormente se modificaron estas plantillas para darles la funcionalidad requerida en este proyecto. Luego se realizaron diversas pruebas y una vez funcionando esta parte del proyecto se prosiguió con el trabajo en el módulo *Rabbit RCM5600*.

En el módulo *Rabbit RCM5600* se realizó la programación necesaria para realizar el puente entre éste y el nodo coordinador de la red de sensores por medio de Dynamic C, que ofrece un entorno amigable de desarrollo, muy similar a los distintos entornos de desarrollo de C. Se diseñó tambien una interfaz de administración del puente utilizando el servidor embebido de este mismo módulo *(RCM5600)*.

Capítulo 5

Implementación

5.1. Introducción

Con el rápido desarrollo de sistemas de hardware embebidos, el desarrollo de puentes se vuelve cada vez más importante y se ofrecen soluciones de desarrollo rápido y confiable.

El diseño de un puente es muy importante en el desarrollo de plataformas *e-salud* y existen muchos aspectos que se deben de tomar en cuenta en dicho diseño.

En algunos casos, como el manejo de desastres, reconocimiento del campo de batalla e instalaciones de seguridad, el diseño de un puente es importante debido a que es grande el número de sensores y trabajan con una batería pequeña.

En algunos otros casos el puente tiene que comunicarse con diferentes redes de sensores para adquisición de datos, hacer funciones de *routing* y otras aplicaciones; puede tener comunicación con protocolos TCP/IP como IPV4 e IPV6.

Para sistemas de tipo plataformas *e-salud*, el reconocimiento del estado de salud de pacientes es muy importante y al mismo tiempo difícil de llevar a cabo, es por eso que para ésta área en particular se requiere de un diseño delicado [7].

5.2. Diseño e implementación

A continuación se muestran los 3 pasos básicos para el desarrollo del puente WiFi-WSN:

- Selección, desarrollo e implementación del hardware.

Figura 5.1: Arquitectura de la plataforma *e-salud*

- Implementación de la funcionalidad del puente la cual consiste en el diseño y desarrollo de las funciones basicas de intercambio de datos entre los ambientes IEEE 802.11 e IEEE 802.15.4.

- Implementación de las aplicaciones. Una vez finalizado el trabajo que brinda la funcionalidad básica, se desarrolló la aplicación que permite la administración del puente, así como la configuración para que funcione en forma independiente de una computadora (modo standalone).

La arquitectura de la plataforma *e-salud* donde se pretende que opere el puente aqui diseñado se muestra en la figura 5.1.

La plataforma *e-salud* consiste básicamente de 3 partes:

- La primera parte es el objeto a monitorizar (red inalámbrica de sensores desplegada sobre los pacientes).

- La segunda parte es el puente entre la *WSN* y el destino final de la información.

- La tercera parte es el destino final de la información, la cual puede ser manipulada por los cuidadores.

Figura 5.2: Arquitectura abstracta de la plataforma *e-salud*

5.2.1. Requerimientos

Los requerimientos del diseño del puente son los siguientes:

- Proveer un mecanismo de conexión entre la estación base de la red de sensores y el puerto de interconexión del dispositivo *Rabbit*.
- Proveer conexión *WLAN* para dispositivos inalámbricos de distancia corta como *PDAs* y *laptops*.
- Recibir y ejecutar comandos.
- Proveer una interfaz de manejo para controlar el flujo de los requerimientos antes mencionados.
- Proveer suficiente rapidez en el procesador, memoria y puertos externos adecuados.

5.2.2. Hardware y Software

Para el diseño de hardware, esos requerimientos pueden ser agrupados dentro de tres diferentes categorías, estas se reflejan en la figura 5.2.

El módulo *Rabbit MiniCore RCM5600W* utilizado en el proyecto cuenta con un grupo de subsistemas que en conjunto forman dicho módulo y que están presentes en el desarrollo del prototipo. La figura 5.3 muestra estos módulos que son parte del *RCM5600W*.

Los puertos del microprocesador *Rabbit 5000* usados en el *RCM5600W* son configurables, y los valores que vienen de fábrica pueden ser configurados. La figura 5.4 muestra los puertos con los que cuentan los módulos del *RCM5600W*.

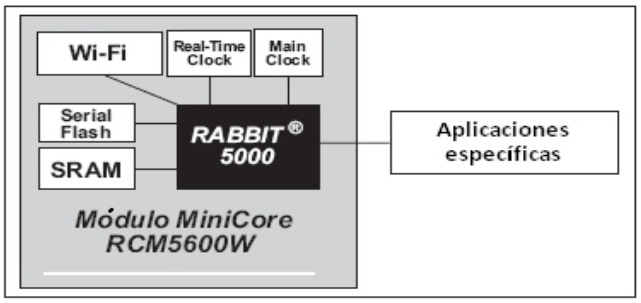

Figura 5.3: Subsistemas RCM5600W

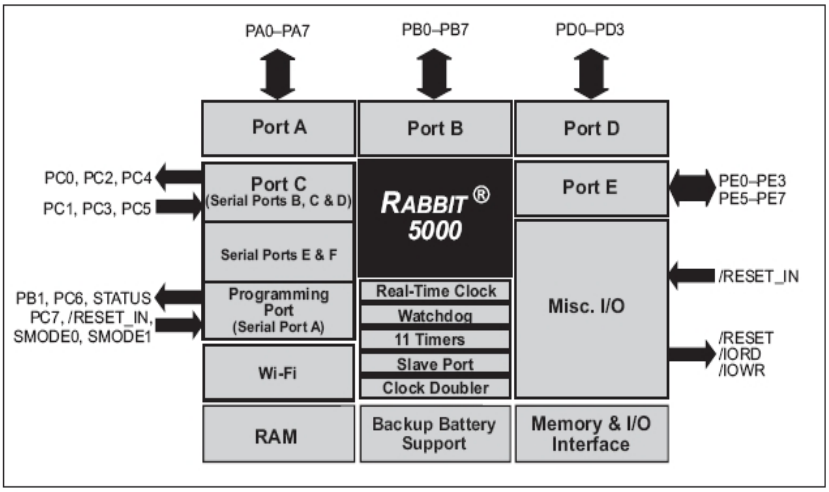

Figura 5.4: Puertos del *Rabbit 5000*

Se hizo uso de los puertos C y D para la conexión serial con el módulo coordinador de la *WSN*.

Por otro lado tenemos el diagrama a bloques del módulo *PAN802154HAR00*, que se muestra en la figura 5.5. Se observan los puertos con los que cuenta, incluyendo el RS232C que utilizamos para la comunicación serial con el dispositivo Rabbit. De igual forma el tranceptor *MC13192* para la comunicación con el destino final de los datos adquiridos.

5.2.3. Comunicación serial

La tarjeta *RCM5600W* no tiene ningún tipo de convertidor serial directamente en la tarjeta. Sin embargo se puede incorporar una interfaz serial o *Ethernet* sobre la tarjeta

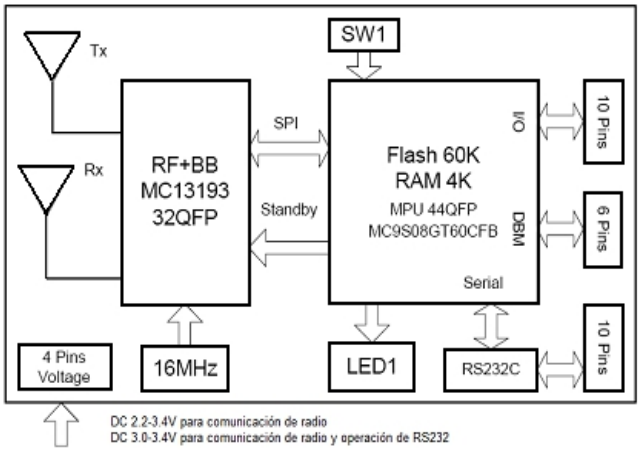

Figura 5.5: Diagrama a bloques del módulo *PAN802154HAR00*

RCM5600W. El *kit* de desarrollo cuenta con una tarjeta de comunicación serial, la cual tiene un transceptor *RS-232* y la tarjeta de interfaz tiene conectores *USB*.

5.2.4. Puertos Seriales

El módulo cuenta con 6 puertos seriales (A, B, C, D, E y F). Todos los puertos pueden operar en modo asíncrono arriba de la tasa estándar de baudios del sistema de reloj, dividido entre 8. Un puerto asíncrono puede manejar 7 u 8 bits. Un esquema de direcciones de 9 bits, donde un bit adicional es enviado para marcar el primer byte de un mensaje también es soportado.

El puerto serial A lo utilizamos como puerto de programación, pero puede ser usado también en el modo asíncrono o como puerto serial sobre reloj, una vez que el desarrollo de la aplicación ha sido completado y el *RCM5600W* esté operando en el Modo *"Run"*.

El puerto serial B, es compartido por el módulo flash del *RCM5600W* y por el convertidor analógico digital en el circuito *WiFi*.

Los puertos seriales C y D pueden también ser operados en el modo serial con reloj, los utilizamos como puertos de comunicación con el módulo coordinador de la red de sensores. En este modo una línea sincroniza los datos de entrada y salida con el reloj.

Los puertos seriales E y F también pueden ser configurados como *SDLC* (Control de

Serial Port A	TXA	PC6, PC7
	RXA	PC7,PE7
	SCLKA	PB1
Serial Port B	TXB	PC4, PC5
	RXB	PC5, PE5
	SCLKB	PB0
Serial Port C	TXC	PC2, PC3
	RXC	PC3, PD3, PE3
	SCLKC	PD2, PE2, PE7, PC7

Serial Port D	TXD	PC0, PC1
	RXD	PC1, PD1, PE1
	SCLKD	PD0, PD3, PE0, PE3, PC3
Serial Port E	TXE	PE6, PC6
	RXE	PE7, PC7
	RCLKE	PE5, PC5
	TCLKE	PE4, PC4
Serial Port F	TXF	PD2, PE2, PC2
	RXF	PD3, PE3, PC3
	RCLKF	PD1, PE1, PC1
	TCLKF	PD0, PE0, PC0

Figura 5.6: Puertos seriales y pines de reloj del *Rabbit 5000*

enlace de datos síncrono - *Synchronous Data Link Control*) / *HDLC* (Control de enlace de datos de alto nivel - *High-Level Data Link Control*). El protocolo *IrDA* también es soportado en formato *SDLC* por estos dos puertos. Los puertos E y F deben ser configurados antes de ser utilizados. Esta configuración se realiza de la siguiente manera:

- #define SERE_TXPORT PEDR
- #define SERE_RXPORT PEDR
- #define SERF_TXPORT PFDR
- #define SERF_RXPORT PFDR

La tabla 5.6 muestra los posibles pines paralelos para los puertos seriales y sus relojes.

5.2.5. Comunicación inalámbrica - *WiFi*

La figura 5.7 muestra el diagrama de bloques funcional para los circuitos de *WiFi*.

La transmisión *WiFi* es controlada por el chip *Rabbit 5000*, el cual contiene el control de acceso al medio (*MAC*). El *Rabbit 5000* implementa la funcionalidad en banda base *MAC* IEEE 802.11b/g y controla el tranceptor *airoha AL2236*. El código de programa es almacenado en la memoria flash y es cargado en la SRAM para su ejecución.

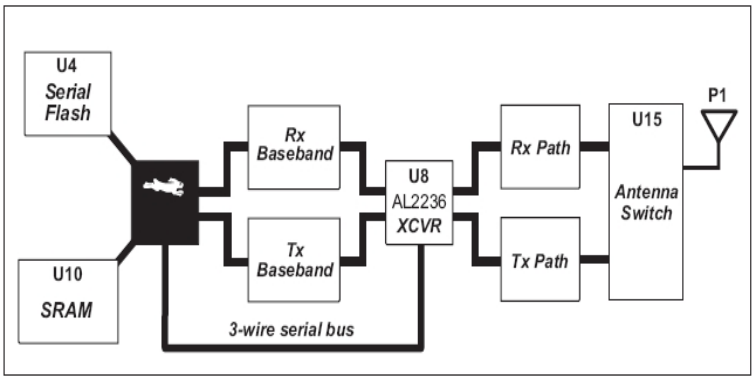

Figura 5.7: Diagrama de bloques de *WiFi* del *RCM5600W*

La interfaz de datos entre el procesador y el transceptor consiste de dos convertidores, *D/A y A/D*. La conversión la hacen a una tasa de 40Mhz.

El *AL2236* es un chip transceptor con un amplificador de potencia integrado para la banda Industrial, científica y médica *(ISM)*. Este es configurado y controlado por el *Rabbit 5000*.

El *AL2236* puede transmitir y recibir datos a más de 11Mbits/s en el modo IEEE 802.11b y arriba de 54 Mbits/s en el modo IEEE 802.11g.

Soporta los canales del 1-13(2.401 Ghz a 2.472 Ghz). El canal 4 no se usa.

5.3. Desarrollo

Se define el ambiente de la red donde se desplegará el puente y unido a este la red de sensores, para ello se reliza la siguiente configuración:

- CONFIG_TCP Es el tipo de configuración de red TCP/IP que vamos a tener, 1 configuración estática y 2 DHCP o configuración dinámica, para nuestro caso asignamos el valor 1.

- _IP_PRIMARIA_ESTATICA: Es la dirección IP estática que le asignamos al dispositivo *Rabbit*, para nuestro caso asignamos la dirección "1.1.1.10".

- _MASCARA_PRIMARIA: Es la máscara para el dispositivo Rabbit, para nuestro caso asignamos la dirección "255.255.255.0".

- PUERTA_ENLACE: Es la dirección que asignamos al *Gateway* en caso de usarlo, en nuestro caso si lo requerimos y le asignamos el valor "1.1.1.1".

- MY_NAMESERVER: Si requerimos del servicio de nombres de dominio habilitamos esta variable, en nuestro caso no lo requerimos.

- IFC_WIFI_SSID: Es el servicio de identificación de red o nombre de la red a la cual se va a conectar el dispositivo Rabbit o cualquier otro dispositivo, se le dio el nombre *"ap-puente-wifi-wsn"*.

- IFC_WIFI_ROAM_ENABLE: Habilita o deshabilita el roaming, en nuestro caso lo habilitamos asignando el valor 1.

- IFC_WIFI_ROAM_BEACON_MISS: Configura el numero de beacons que son perdidos continuamente para escanear y encontrar un mejor punto de acceso y luego asociarse a el, en nuestro caso asignamos el valor 20.

- IFC_WIFI_MODE: Especifica la arquitectura de la red para la red inalámbrica, en nuestro caso asignamos IFPARAM_WIFI_INFRASTRUCTURE.

- IFC_WIFI_REGION: Configura el rango del canal y máxima potencia para la región seleccionada, nosotros usamos IFPARAM_WIFI_REGION_AMERICAS con los canales 1-11.

- IFC_WIFI_ENCRYPTION: Controla el tipo de encriptación usada, en nuestro caso no utilizamos ningún tipo de encriptación, lo dejamos abierto IFPARAM_WIFI_ENCR_NONE.

- RED: Esta opción se definió para el tipo de red de sensores que usaremos, el tipo es estrella y la llamamos *"star"*.

Se definieron 3 buffers para intercambio de datos entre puertos seriales y puertos TCP los cuales corresponden a los estándares IEEE 802.11 e IEEE 802.15.4, para lo cual se definió uno de tamaño 1024 (TCP) y otros 2 (e/s) de 127 (serial) cada uno.

Se habilitaron 2 puertos serie del dispositivo *Rabbit 5600w*, los cuales pueden ser configurados desde la página web de configuración del puente; la configuración inicial de ambos puertos es la siguiente:

- Puerto Local TCP: 4567 y 4568 (C y D respectivamente).
- Bits por segundo: 19200
- Bits de datos: 7
- Paridad: None
- Bits de parada: 1

El módulo cuenta con servidores HTTP y FTP, así como con clientes HTTP, FTP, TFTP, SMTP, POP3 y servicios como *telnet* y una consola de propósito general. Para el desarrollo del proyecto se hace uso del servidor *HTTP*, el cual elimina la necesidad de programación de *CGI's* dando libertad para usar una amplia variedad de herramientas de diseño web. Dicho servidor es llamado *RabbitWeb* y usa un lenguaje de tipo script llamado zhtml, el cual es muy similar a html tanto en sintaxis como en facilidad.

Permite además del uso del lenguaje *Dynamic C*, el cual incluye directivas de compilación que pueden ser añadidas a la aplicación llamando al servidor HTTP. Esto permite diseñar una interfaz web con interconectividad (entre web y hardware) casi transparente al programador.

Mediante esta propiedad del módulo *Rabbit*, se diseñó una interfaz que permite modificar la configuración del módulo y obtener una monitorización de las estadísticas de envío/recepción de paquetes entre el módulo y el coordinador de la red de sensores quienes forman el puente.

Se hace uso de las dos librerías principales de TCP/IP definiendolas:

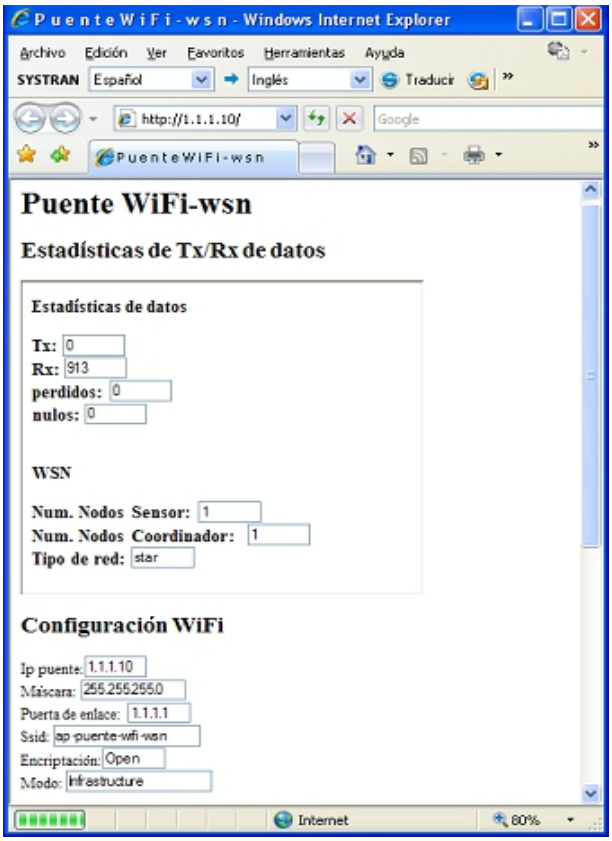

Figura 5.8: Interfaz de configuración/monitorización del puente

- #use "dcrtcp.lib"
- #use "http.lib"

También de la librería propia del modelo Rabbit que usamos:

- #use "rcm56xxw.lib"

Se diseñó la interfaz principal, que es llamada config.zhtml y otra página llamada datos.zhtml la cual refresca los datos monitorizados de la red de sensores aproximadamente cada segundo.

En las figuras 5.8 y 5.9 se muestran los valores configurables / monitorizados en la interfaz.

5.3.1. Estadísticas de datos

La interfaz cuenta con una serie de estadísticas que miden la cantidad de datos que son transmitidos y recibidos por el puente. Tambien registra la cantidad de paquetes perdidos y nulos. De esta manera se conoce el rendimiento del puente. La figura 5.8 muestra esto.

- Tx es el número de paquetes transmitidos del puente hacia el nodo coordinador de la wsn.
- Rx es el número de paquetes transmitidos del nodo coordinador de la wsn hacia el puente.
- Perdidos es el número de paquetes perdidos durante una sesión de transmisión, ya sea por interferencia, pérdida de potencia o desconexión de motes.
- Nulos es el número de paquetes transmitidos que carecen de carga útil.

5.3.2. WSN

En esta parte se registran todos los eventos que suceden en una red de sensores, se registra cada nodo tanto sensor como coordinador que se conecta y desconecta de la red. La figura 5.8 muestra esto.

- Núm. de Nodos Sensor es el número de nodos transmitiendo simultáneamente al nodo coordinador del puente.
- Núm. de Nodos Coordinador es el número de nodos coordinador capaces de recibir datos de la red, el puente únicamente permite la detección de un solo nodo coordinador.
- Tipo de red es el tipo de red de la red de sensores.

5.3.3. Configuración WiFi

Esta sección de la interfaz detecta la configuración actual de *WiFi* del puente. La figura 5.9 muestra esto.

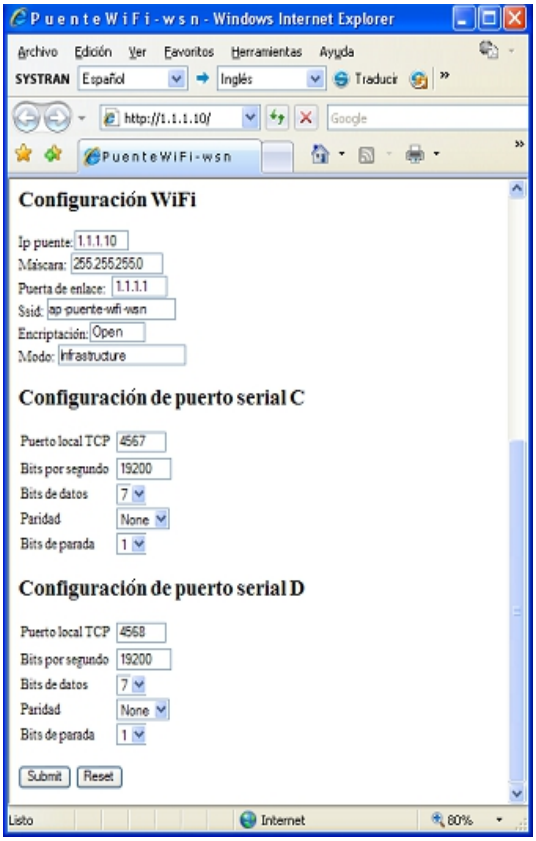

Figura 5.9: Interfaz de configuración/monitorización del puente

- Ip puente es la dirección asignada al puente.
- Máscara es la dirección asignada a la máscara de subred.
- Puerta de enlace es la puerta de enlace perteneciente a la red.
- *SSID* es el servicio de identificación de red al cual se conectará el puente y cualquier otro dispositivo.
- Modo es el modo de funcionamiento de la red, en nuestro caso siempre será de tipo infraestructura.

5.3.4. Configuración de puerto serial C y D

Esta sección describe la configuración de los puertos seriales C y D del dispositivo *Rabbit* donde es posible realizar la interfaz con el nodo coordinador de la red de sensores. La figura 5.9 muestra esto.

- Puerto local TCP al conectarnos via telnet socket lo haremos a alguno de estos dos puertos, asignando los valores 4567 y 4568 respectivamente.
- Bits por segundo es la tasa de transmisión entre el nodo coordinador y el dispositivo rabbit, ambos son configurados con el valor 19200, siendo posible cambiar esta velocidad siempre y cuando sea en ambas partes.
- Bits de datos es el número de bits de datos en la transmisión, también puede ser cambiado.
- Paridad en nuestro caso no requerimos de paridad.
- Bits de parada es el número de bits de parada.

Ambas páginas son copiadas a la memoria del rabbit usando la función #ximport y posteriormente asociadas al servidor web.

Se definió un socket que se asocia con el puerto serial que se elija y punteros para acceder a este: ABRIR, CERRAR, LEER, ESCRIBIR, DEFINIRBITDEDATOS y DEFINIRPARIDAD.

5.3.5. Funcionalidad del puente

La figura 5.10 muestra el diagrama de flujo de la función principal que consiste en una máquina de estados (INICIALIZACION, ESCUCHAR Y PROCESO); todo esto en conjunto forma la mayor parte del firmware que le da el funcionamiento de transmisión/recepción al puente.

A continuación se muestran las funciones que permiten el funcionamiento del puente:

- Función ReiniciarSocket: Aborta el socket actual, reinicia la máquina de estados y reabre el socket.
- Función ReiniciarPuertoSerial: Cierra y reabre el puerto serial seleccionado.
- Función ActualizarPuertoSerial: Esta función es llamada cuando se actualiza un puerto serial desde la página de configuración. Se determina cual de los 2 puertos se cambio y reinicia con los nuevos parámetros.
- Función AbrirSerial: Sirve para inicializar el puerto serial y dejarlo listo para su uso, se inicializan todos los parámetros: número de bits de datos, stop bits y paridad.
- Función InicializarSerial-Wifi: Inicializa la máquina de estados TCP a WiFi.
- Función Principal: Inicializa la pila TCP/IP, el servidor HTTP y la máquina de estados TCP-WiFi.

El funcionamiento de la máquina de estados es el siguiente:

Primeramente se inicializa la interfaz serial-WiFi quedando en modo de espera (estado Inicializar Serial/WiFi) hasta que se transmita algún paquete ya sea de WiFi a serial o

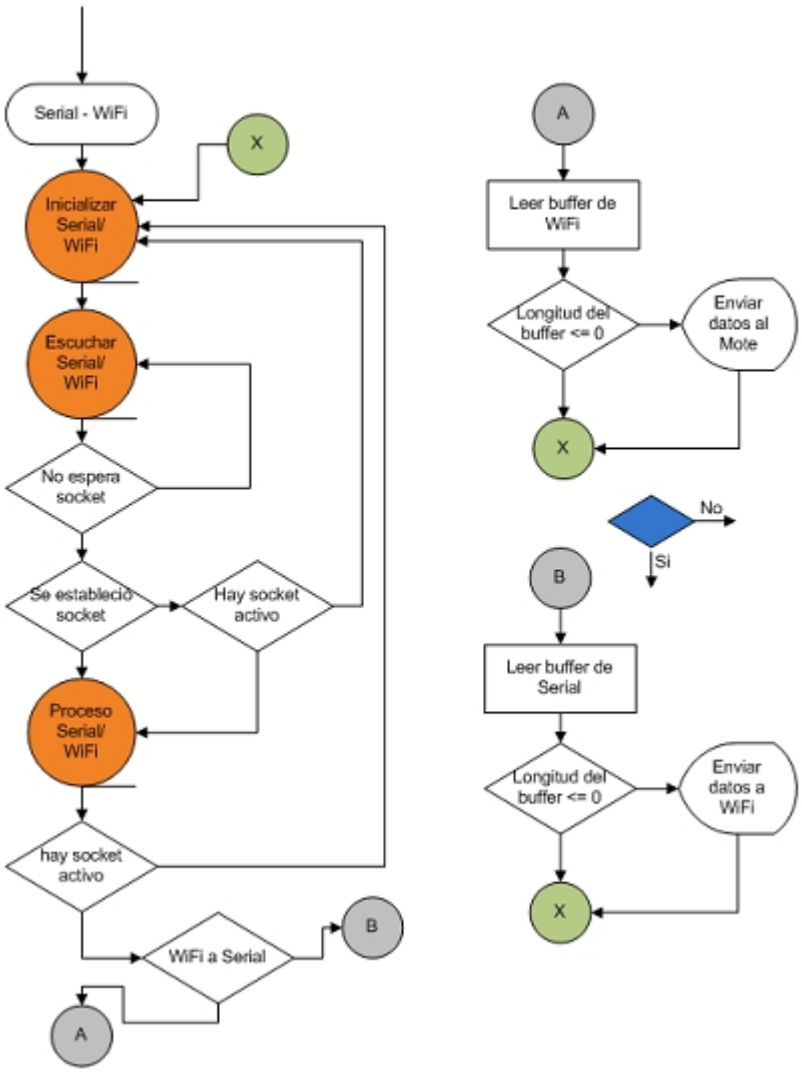

Figura 5.10: Funcionamiento de la máquina de estados necesaria para el intercambio de paquetes entre estándares (802.11 y 802.15.4)

de serial a WiFi. Se espera la creación de un socket, una vez establecido entra al proceso serial/WiFi, si no existe ningún socket o se pierde la conexión, se regresa al estado Inicializar serial/WiFi. En el estado Proceso Serial/WiFi vuelve a verificar si se cuenta con algún socket activo, si es asi, verifica la procedencia del paquete, si es de WiFi a Serial o viceversa. Si el dato proviene de WiFi se verifica la longitud del buffer, y si ésta es mayor a 0 se envía el paquete al mote. Si el dato proviene del nodo coordinador de la red de sensores, de igual forma se verifica si la longitud del buffer es mayor a 0, si es así se envían los datos a WiFi. Una vez ocurrido cualquiera de estos dos eventos, la máquina de estados regresa al estado inicial.

Otra parte de la funcionalidad la forman la interfaz y el módulo coordinador de la red de sensores.

Transmisión de paquetes

El diagrama 5.11 muestra el procedimiento de envío de paquetes entre cada uno de los nodos sensores y el nodo coordinador de la red de sensores.

A continuación se describe el funcionamiento del procedimiento que permite la transmision de paquetes del nodo sensor hacia el nodo coordinador.

El programa espera la ocurrencia de un evento (Únicamente se cuenta con el evento presionar el botón del mote). si el evento es presionar el botón, verifica si el estado es inicializar, si es asi, realiza las siguientes acciones para preparar al mote:

- Inicializa el estándar IEEE 802.15.4
- Inicializa el Temporizador del mote
- Inicializa el puerto serial
- Inicializa el teclado (existen motes con más de un botón, en nuestro caso solo contamos con uno)

Una vez preparado el mote para transmitir, establece el estado a modo de escucha. Y regresa a verificar el estado en el que se encuentra, primero verifica si es Inicializar, si no

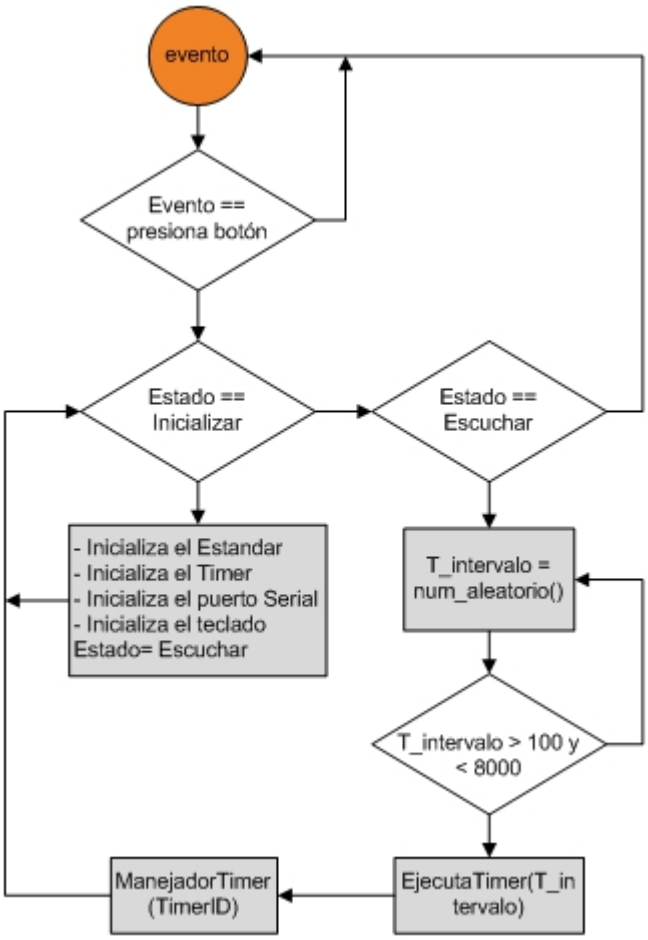

Figura 5.11: Procedimiento para transmitir paquetes en tiempos aleatorios de los nodos sensores al nodo coordinador

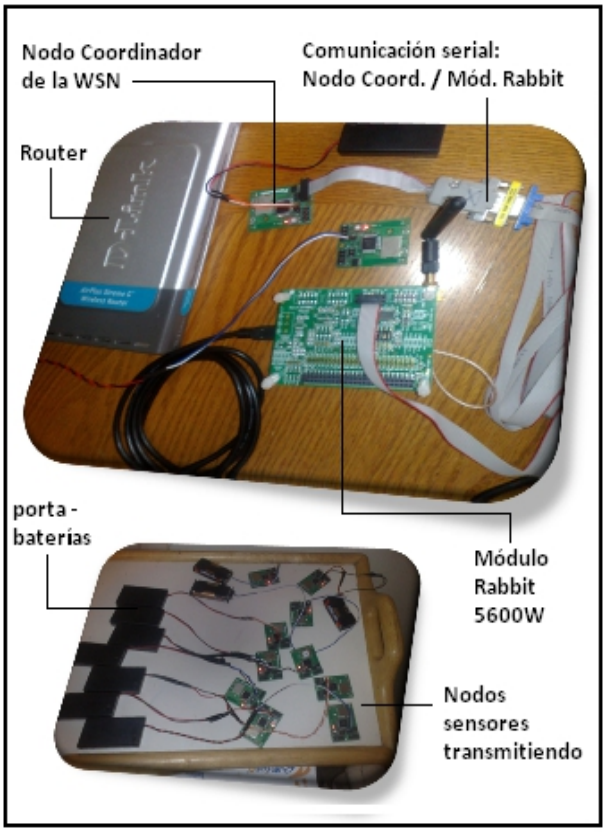

Figura 5.12: Implementación del proyecto

es Inicializar como es el caso, verifica si es escuchar, si es escuchar una variable llamada T-intervalo toma un valor, ya sea que lo asignemos directamente o como se presenta: se asigna un valor aleatorio entre 100 y 8000 milisegundos. Una vez asignado el tiempo de transmisión del paquete se ejecuta el temporizador con el valor asignado a la variable T-intervalo y se ejecuta la función que controla al temporizador y regresa a verificar el estado. De aquí en adelante el estado siempre será escucha, a menos que se rompa algún punto de la conexión.

La figura 5.12 muestra la implementación de todos los elementos presentes en el desarrollo del proyecto y funcionando.

Observamos primeramente que módulo rabbit esta conectado al nodo coordinador por medio del cable serial. Ambos cables coinciden en un conector serial macho-macho y de esta manera se logra la unión, la figura 5.13 muestra la configuración que se debe de seguir al conectar el conector serial que viene del módulo rabbit y el conector serial del módulo coor-

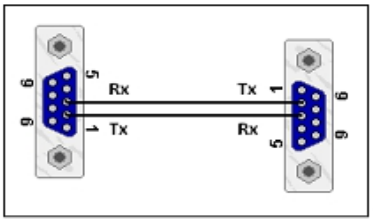

Figura 5.13: Conexión serial entre el módulo *Rabbit* y el nodo coordinador de la *WSN*

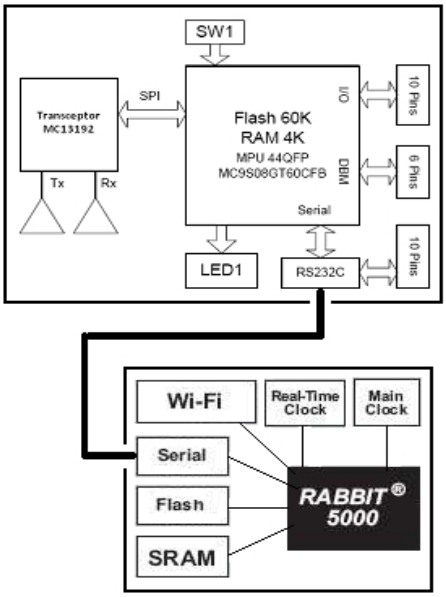

Figura 5.14: Diagrama de bloques de la conexión entre el módulo *Rabbit* y el nodo coordinador de la *WSN*

dinador.

La figura 5.14 muestra la interconexión del módulo *PAN802154HAR00* y el módulo Rabbit mostrado en sus diagramas de bloques.

Primero se desarrolló el modelo en una computadora con herramientas de desarrollo proporcionadas bajo el ambiente *Rabbit*, posteriormente se depuró y finalmente se genera un archivo imagen que se graba directamente en el dispositivo *Rabbit* de manera que pueda funcionar de forma independiente.

5.4. Resumen

En este capítulo se presentaron primeramente los módulos de hardware de cada uno de los dispositivos utilizados, los requerimientos y arquitectura del proyecto, se explicó a detalle una de las partes fundamentales del proyecto: Primero la comunicación serial e inalámbrica con la que cuentan los módulos *PAN802154HAR00* y el módulo rabbit5600w y posteriormente la conexión entre dichos dispositivos. Se explicó de igual forma el desarrollo del software implementado en el puente, tanto el desarrollado en el dispositivo *Rabbit* como en el nodo coordnador de la *WSN*.

Se muestra finalmente la puesta en funcionamiento del puente y la red de sensores, indicando todos los pasos que llevan a esto.

Capítulo 6

Resultados

6.1. Introducción

En esta parte se muestran los resultados obtenidos de la implementación y puesta a punto del Puente desarrollado. Se realizaron diversas mediciones estableciendo una serie de métricas las cuales muestran el desempeño general del puente.

Se define una serie de escenarios y se realizan las pruebas para cada uno de ellos. Los resultados se presentan mediante gráficas que muestran el número y porcentaje de paquetes perdidos, de esta forma se conoce el comportamiento y por ende el desempeño de los diferentes esquemas de medición.

Los escenarios y resultados para cada escenario se muestran a continuación:

6.2. Escenarios

Se establecieron escenarios que cubren todos los puntos críticos del puente y se realizaron las mediciones.

Se plantearon 3 escenarios principales:

- 3 escenarios (1-A, 1-B y 1-C) para medir el rendimiento y confiabilidad en la transmisión de paquetes de la red de sensores inalámbrica al puente o dicho de otra forma de WSN a WiFi.

Escenario 1-A

- El primero de los cinco escenarios establece una red de sensores inalámbrica de tipo estrella con 1 nodo coordinador y nodos sensores que van de 1 a 10.

  Se enviaron paquetes desde una red de sensores inalámbrica tipo estrella a una computadora por medio de la interfaz serial donde cada nodo sensor en tiempos aleatorios que oscilan entre 100 y 8000 milisegundos. El nodo coordinador recibe los paquetes y los retransmite a una computadora por medio de una interfaz serial.

Resultados para el Escenario 1-A:

La figura 6.1 y la figura 6.2 muestran los resultados de pérdida y porcentaje de pérdida de paquetes respectivamente para las distancias a 1,5, y 10 metros en un espacio abierto libre de obstáculos.

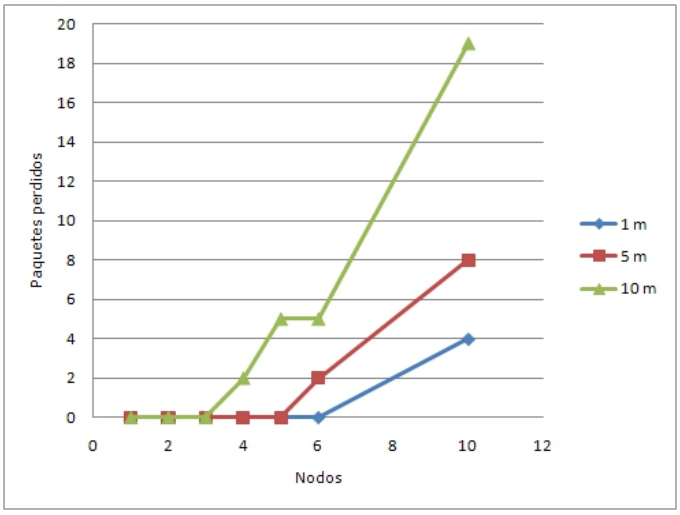

Figura 6.1: Pérdida de paquetes a 1, 5 y 10 metros en espacio abierto

Para esta parte del escenario, el mayor porcentaje de pérdida se observa en la distancia de 10 metros y 10 nodos con 19 paquetes perdidos de un total de 5000, dando un porcentaje de 0.0038 %. Todos los demás esquemas muestran un porcentaje de pérdida aun menor a este.

La figura 6.3 y la figura 6.4 muestran los resultados de pérdida y porcentaje de pérdida de paquetes respectivamente para una distancia de 15 metros con obstáculos presentes (básicamente paredes).

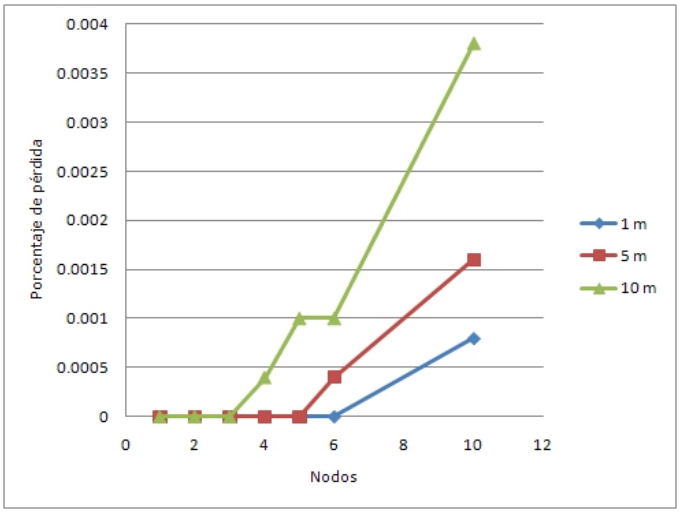

Figura 6.2: Porcentaje de pérdida de paquetes a 1, 5 y 10 metros en espacio abierto

En esta parte del escenario, el mayor porcentaje de pérdida es de 0.0104 % con 52 paquetes perdidos de 5000. A menor cantidad de nodos, menor pérdida.

Las pruebas realizadas para este primer escenario muestran que la pérdida de paquetes es mínima en comparación con el porcentaje de paquetes entregados, teniendo un porcentaje de pérdida dentro del 1 % tanto en las distancias de 1, 5 y 10 metros como en la distancia de 15 metros, siendo en esta última ligeramente alto el porcentaje de pérdida, pero dentro del rango mínimo.

Escenario 1-B

- El segundo escenario establece la misma red de sensores inalámbrica tipo estrella con un nodo coordinador y el mismo número de nodos sensores enviando paquetes en tiempos aleatorios entre 100 y 8000 milisegundos. Ahora el nodo coordinador establece comunicación con el dispositivo Rabbit 5600w mediante una interfaz serial y retransmite los paquetes recibidos de la red de sensores a este.

Resultados para el Escenario 1-B:

La figura 6.5 y la figura 6.6 muestran los resultados de pérdida y porcentaje de pérdida de paquetes respectivamente para las distancias a 1,5, y 10 metros en un espacio abierto libre de obstáculos.

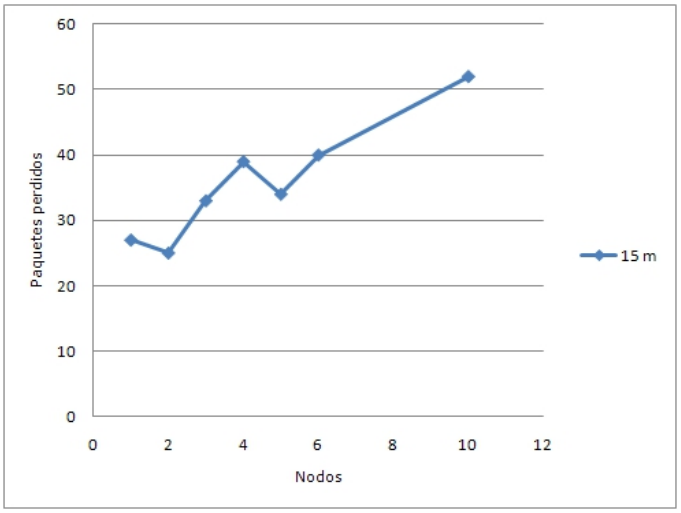

Figura 6.3: Pérdida de paquetes a 15 metros en espacio cerrado con obstáculos

Para esta parte del escenario, el mayor porcentaje de pérdida se observa en la distancia de 10 metros y 10 nodos con 18 paquetes perdidos de un total de 5000, dando un porcentaje de 0.0036 %. Todos los demás esquemas muestran un porcentaje de pérdida aún menor a este.

La figura 6.7 y la figura 6.8 muestran los resultados de pérdida y porcentaje de pérdida de paquetes respectivamente para una distancia de 15 metros con obstáculos presentes (básicamente paredes).

En esta parte del escenario, el mayor porcentaje de pérdida es de 0.0114 % con 57 de 5000 paquetes perdidos.

Las pruebas realizadas para este segundo escenario no muestran grandes diferencias con respecto al esquema de mediciones anterior, muestran que la pérdida de paquetes es minima en comparación con el porcentaje de paquetes entregados, teniendo un porcentaje de pérdida dentro del 1 % tanto en las distancias de 1, 5 y 10 metros como en la distancia de 15 metros, siendo en esta última ligeramente alto el porcentaje de perdida, pero dentro del rango minimo. Ambos esquemas muestran una pérdida no significativa para fines de comparación. Por lo tanto podemos afirmar que las irregularidades o diferencias que se observan son puramente

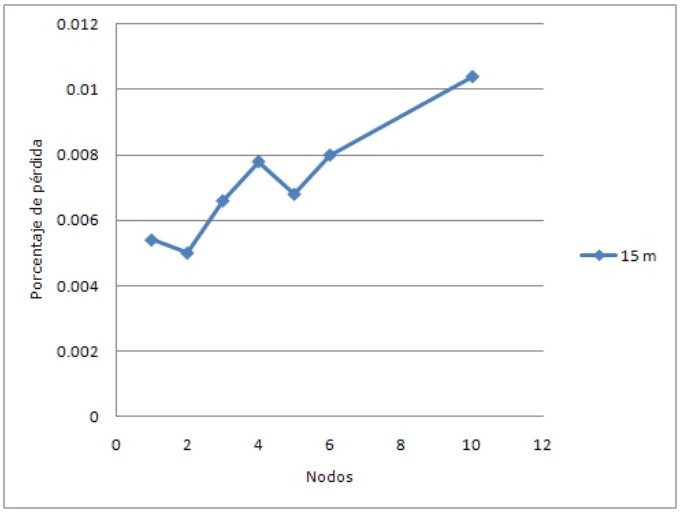

Figura 6.4: Porcentaje de pérdida de paquetes a 15 metros en espacio cerrado con obstáculos

debido a la red de sensores, siendo indistinto si se transmite a la terminal de una computadora o a la terminal *WiFi* del dispositivo *Rabbit*.

En estos dos primeros escenarios se monta una red en la que se va incrementando el número de nodos que se adhieren a esta, siguiendo el siguiente patrón: 1 nodo, 2 nodos, 3 nodos, 4 nodos, 5 nodos, 6 nodos, y finalmente 10 nodos.

En cada uno de estos dos escenarios se realizan pruebas a 1 metro, 5 metros, 10 metros y 15 metros, las pruebas a 1, 5 y 10 metros se realizan en un espacio abierto sin obstáculos y la última prueba de 15 metros, se realiza en un espacio cerrado con obstáculos. Cada una de estas 4 pruebas se realiza con 1, 2, 3, 4, 5, 6 y 10 nodos.

Se envían 5000 paquetes por cada nodo sensor, formando el siguiente esquema de series de prueba:

Envío de 5000 paquetes con 1 (, 2, 3, 4, 5, 6, 10) sensor(es) a 1 (, 5, 10, 15) metro(s) de distancia entre el nodo coordinador y el(los) nodo(s) sensor(es).

Escenario 1-C

- El tercer escenario establece una red de 4 nodos sensores formando una red de tipo estrella, Se enviaron 5 series de 10000 paquetes cada una a una velocidad fija de 4, 8, 16, 32 y 64 paquetes por segundo, por lo que cada nodo envía 1, 2, 4, 8,

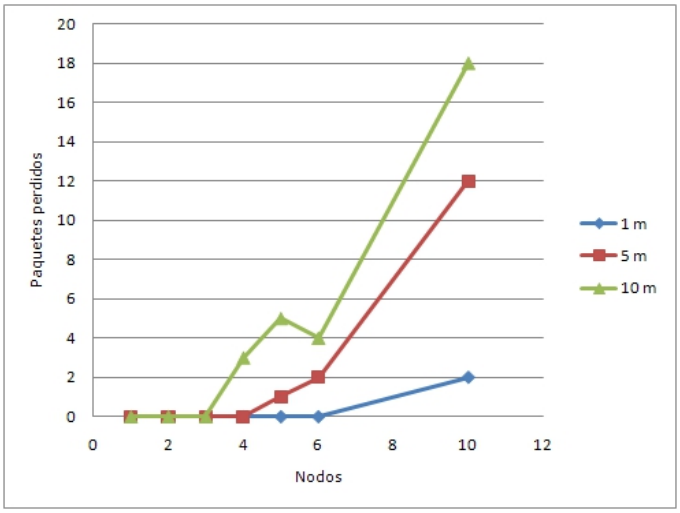

Figura 6.5: Pérdida de paquetes a 1, 5 y 10 metros en espacio abierto

16 paquetes por segundo respectivamente cada uno. Este envio se realiza desde la red de sensores hacia el dispositivo Rabbit 5600w por medio de la interfaz serial a una distancia de 1 metro cada nodo. La distancia fue igual para estos 5 esquemas. Se establecen así esquemas segun la velocidad de transmisión y se envían 10000 paquetes por cada uno de estos; es decir, para el primer esquema cada uno de los 4 nodos envia 1 paquete por segundo. Este esquema es igual para los siguientes, doblando unicamente su velocidad hasta llegar a 64 paquetes por segundo con el cual cada nodo enviara 16 paquetes por segundo.

Resultados para el escenario 1-C:

La figura 6.9 muestra los resultados de pérdida de paquetes respectivamente para una distancia de 1 metro en un espacio abierto libre de obstáculos.

- 2 escenarios (2-A y 2-B) para medir el rendimiento y confiabilidad en la transmisión de paquetes del puente hacia la red de sensores inalámbrica o en otras palabras de WiFi a WSN.

Escenario 2-A

- Este escenario a diferencia de los anteriores invierte la transmisión de paquetes

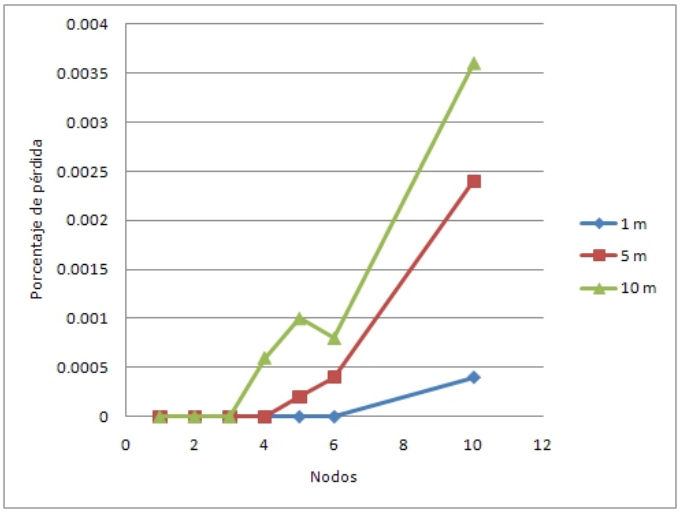

Figura 6.6: Porcentaje de pérdida de paquetes a 1, 5 y 10 metros en espacio abierto

estableciendo comunicación entre el puente y el puerto serial de la computadora, siendo la dirección de transmisión de paquetes de *WiFi* a serial. En este escenario se envían 4 series de 2104 paquetes cada una a velocidades de 1.51 paquetes por segundo, 5.43 paq. por seg., 10.95 paq. por seg. y 15.58 paq. por seg., y se calcula el promedio en pérdida de paquetes. Se mide la transmisión máxima de paquetes en un espacio de tiempo que el puente es capaz de enviar.

Resultados para el Escenario 2-A:

No se observa perdida alguna para una velocidad maxima de 15.58 paquetes por segundo, sin embargo al incrementar esta velocidad a 17.23 se observa pérdida como lo muestra el siguiente escenario.

Escenario 2-B

- Éste escenario es similar al inmediato anterior solo que aquí se envian menos paquetes a una velocidad mayor, esto con el fin de conocer la máxima capacidad de envio donde no exista pérdida. En este escenario se envian 2 series de 217.7 paquetes cada una, la primera a una velocidad de 17.23 paquetes por segundo y la segunda dobla la velocidad a 34.46 paquetes por segundo.

Resultados para el Escenario 2-B:

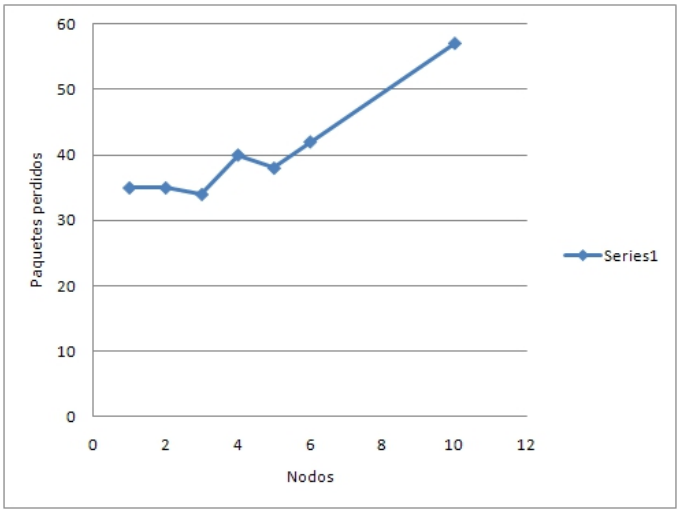

Figura 6.7: Pérdida de paquetes a 15 metros en espacio cerrado con obstáculos

En este escenario se envian 2 series de 217 paquetes cada una a velocidades de 17.23 paquetes por segundo y 34.46 paquetes por segundo. Las figura 6.10 muestra el número de paquetes perdidos.

En este escenario se observa una pérdida minima a una velocidad de 17.23 paquetes por segundo; se observa que se pierden 8 paquetes por cada 217 que se envian, teniendo un porcentaje de pérdida del 0.96 %. Esta perdida se incrementa significativamente al incrementar (doblar) la velocidad a 34.46 paquetes por segundo teniendo una pérdida de 61.60 paquetes de 217 que se envian con un porcentaje del 28.3 %, el cual se considera alto.

- 1 escenario para medir de forma global el rendimiento y confiabilidad en la transmsion de paquetes de la red de sensores inalambrica de *WiFi* a *WSN* en una ventana de tiempo.

Escenario 3

En el escenario 3, se programan cada uno de los motes para que transmitan paquetes a una velocidad fija de 16 paquetes por segundo. Se monitorizan las transmisiones y retransmisiones y se obtienen resultados que indican el rendimiento durante un lapso de 1 minuto o dicho de otra forma se establece una ventana de 1 minuto en la cual el(los)

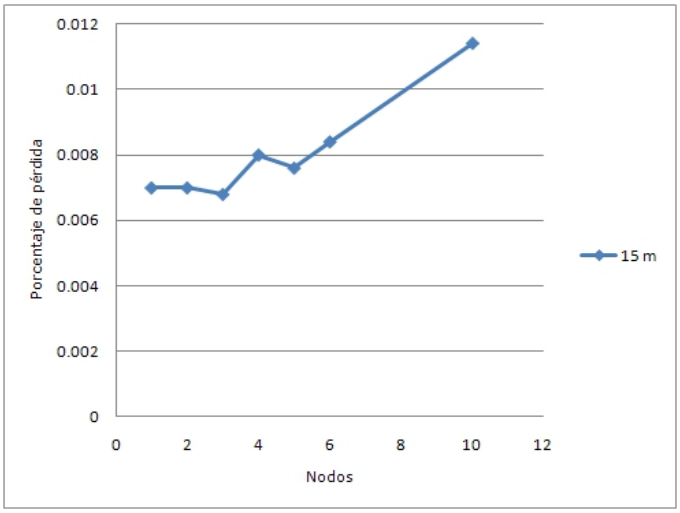

Figura 6.8: Porcentaje de pérdida de paquetes a 15 metros en espacio cerrado con obstáculos

nodo(s) implicado(s) transmite(n) paquetes. Por lo tanto se tienen 4 esquemas de prueba, el primero con 1 nodo transmitiendo 16 paquetes por segundo que durante un minuto transmite un total de 960 paquetes, el segundo esquema son 2 nodos transmitiendo de igual forma 16 paquetes por segundo cada uno para tener un total de 1920 paquetes en 1 minuto, el tercero son 3 nodos que al final envian un total de 2880 paquetes en un minuto y finalmente 4 nodos con un total de 3840 transmisiones / retransmisiones.

Resultados para el escenario 3:

En este escenario se establecieron 4 esquemas iguales que varian unicamente por el total de nodos presentes en la red de sensores; al final se monitorizaron el total de transmisiones y retransmisiones. La gráfica 6.11 muestra los resultados de esta prueba.

Se observa que mientras se incrementa el número de nodos, se incrementa el numero de retransmisión de paquetes, en las pruebas se observa que la mayoría de estas retransmisiones son debido a colisiones entre paquetes de diferente mote.

6.3. Análisis de los resultados

Se midió el total de pérdida de paquetes y el porcentaje de pérdida de paquetes, de esta forma se puede medir el rendimiento y confiabilidad.

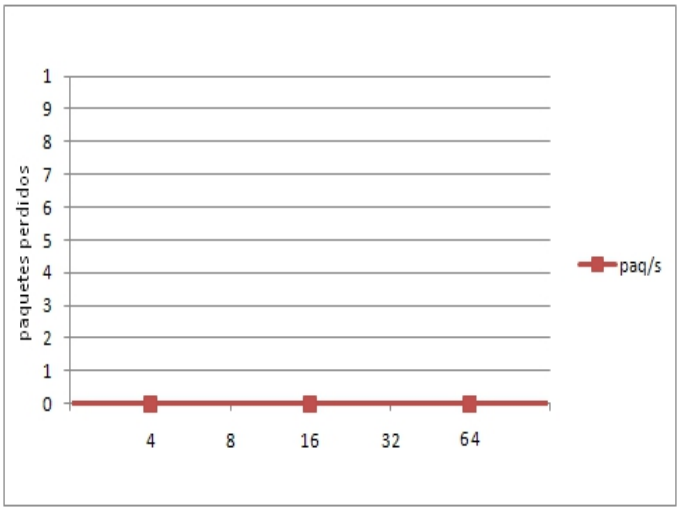

Figura 6.9: Pérdida de paquetes a 1 metro en espacio abierto sin obstáculos

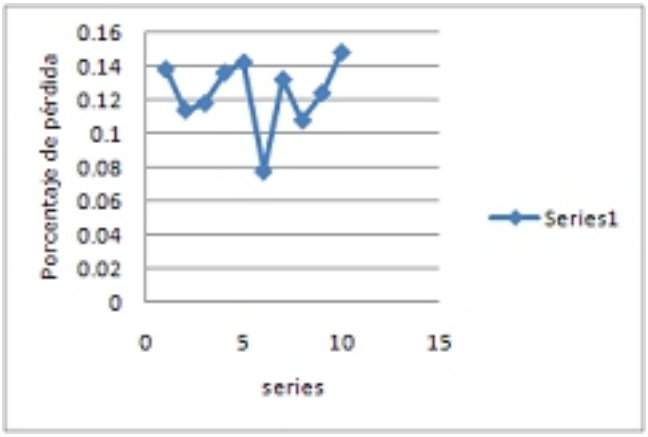

Figura 6.10: Pérdida de paquetes de *WiFi* a serial

Los resultados obtenidos con las pruebas realizadas muestran que el puente es eficiente en la comunicación de datos de WSN a WiFi siendo el desempeño de la red totalmente transparente al puente en este aspecto. La pérdida de paquetes se debe principalemente a las pérdidas inherentes o propias de la red de sensores, que aunque se observa que el desempeño es excelente aun con incremento de nodos, obstáculos y distancias, se tiene una pérdida mínima.

Otro punto que se evaluo del puente es la transmisión de paquetes con dirección *WiFi* a *WSN*. En este punto se observa una pérdida del 12.38 % en comparación con el 1 % de *WSN* a *WiFi*, la cual se esperaba por la conversión y diferencia de marcos entre estándares.

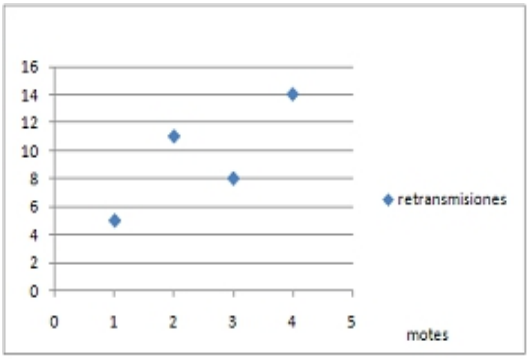

Figura 6.11: Retransmisión de paquetes de WiFi a serial

Para el propósito original del desarrollo del puente, (el cual es formar parte de la plataforma *e-salud*), se puede afirmar que también es eficiente, ya que la mayor carga de de datos es soportada por la dirección *WSN* a *WiFi*; la plataforma planteada despliega un determinado número de pacientes con un determinado número de sensores transmitiendo información al puente, en otras palabras *WSN* a *WiFi*, y aunque se requiere el envío de paquetes de configuración por parte de la aplicación, pasando por *WiFi* y llegando a cada uno de los nodos sensores de los pacientes, este es mínimo.

6.4. Resumen

En este capítulo se muestran los resultados obtenidos en base a pruebas realizadas en diferenes escenarios. Se definieron básicamente 6 escenarios que miden el rendimiento en la transferencia de paquetes que realiza el puente entre los dos estándares (IEEE 802.15.4 e IEEE 802.11).

Los resultados muestran un excelente rendimiento y permiten establecer una plataforma para el desarrollo de sistemas que requieran intercambiar paquetes entre ambos estándares.

Capítulo 7

Conclusión

7.1. Conclusión

En este trabajo de tesis se muestran las ventajas de trabajar con redes inalámbricas para la implementación de plataformas e-salud principalmente. Estas tecnologías que son IEEE 802.15.4 e IEEE 802.11 básicamente cuentan con ventajas que al combinarlas para desarrollar dispositivos de interconexión de tecnologías de distinto tipo, ofrecen herramientas muy poderosas y útiles para desplazar datos entre ambientes heterogéneos.

Se muestra la motivación principal para el desarrollo de este proyecto, así como los alcances del mismo, los cuales ofrecen una solución inteligente para el despliegue de redes inalámbricas de sensores, así como la captación y procesamiento de los datos obtenidos.

Se observan también los resultados de diversas pruebas aplicadas al puente desarrollado que muestran el excelente desempeño de éste. Esto permite promover el puente para su implementación en un escenario real que no necesariamente tiene que ser una plataforma de e-salud.

Glosario

ACK (ACKNOWLEDGEMENT): Acuse o reconocimiento es un mensaje que se envía para confirmar que un mensaje o un conjunto de mensajes han llegado.

CSMA/CA (Carrier Sense Multiple Access/Collision Avoidance): Acceso múltiple por detección de portadora con evasión de colisiones es un protocolo de control de redes de bajo nivel que permite que múltiples estaciones utilicen un mismo medio de transmisión.

FFD: Full Function Device son dispositivos con funcionalidad completa capaces de funcionar como nodos coordinador dentro de una red de sensores inalámbrica.

IEEE: *Institute of Electrical and Electronics Engineers*, Instituto de Ingenieros Electricos y Electronicos. Es una institución profesional mundial encargada de estandarizar.

ISO: *International Organization for Standardization*, Organización Internacíonal para la Estandarización. Organismo encargado de promover el desarrollo de normas internacionales para industrias, comercios y las comunicaciones.

MAC: *Medium Access Control*, Control de Acceso al Medio. Es una de las subcapa de la capa de enlace de datos del modelo OSI y se encarga de controlar el acceso al medio de los nodos, este determina quien y cuando transmitira el nodo.

OSI: *Open System Interconnection*, Modelo de Referencia de Interconexión de Sistemas Abiertos. Modelo de red Creado por la ISO para definir arquitecturas de interconexión de sistemas de comunicación.

RFD (Reduced Function Device): Son dispositivos con funcionalida limitadas que en una red de sensores únicamente pueden tomar el rol de nodos sensores.

RFID (Radio Frecuency ID): Identificador de Radio Frecuencia. Es un dispositivo pequeño de Radio Frecuencia que sirve para obtener información de un artículo, animal o persona a través de peticiones, el funcionamiento es muy similar a los códigos de barras.

Sink: Estación Base de una red de sensores, es el dispositivo por medio del cual pasa toda la información que se esta transmitiendo en la red.

Nodo coordinador: Nodo conformado por la unión de un nodo con protocolo 802.11 y un mote con protocolo 802.15.4 que recolecta la información de todos los motes.

Mote: Nodo sensor que utiliza protocolo de comunicación 802.15.4. Celda: área dentro de la cual se encuentra un nodo coordinador y una determinada cantidad de motes.

Repositorio central: Gateway que recibe la información recolectada por cada uno de los nodos coordinadores.

TCP/IP (Transmision Control Protocol/Internet Protocol): Protocolo de Control de Transmisión/Protocolo de Internet es un conjunto de protocolos de red en los que se basa Internet y que permiten la transmisión de datos entre redes de computadoras.

Wi-Fi: Wireless Fidelity es una marca de la Wi-Fi Alliance, la organización comercial que adopta, prueba y certifica que los equipos cumplen los estándares IEEE 802.11 relacionados a redes inalámbricas de área local.

WSN (Wireless Sensor Networks) Redes de sensores inalámbricas son redes compuestas por nodos sensores capaces de monitorizar determinada área y enviar datos de forma inalámbrica a un nodo coordinador.

Bibliografía

[1] Lantronix, Inc. *How Device Servers Modernize Health Care Information Systems* The Right Rx for Upgrading Today s Hospitals 167 Technology Dr. Irvine, CA (p: II)

[2] Ian F. Akyildiz, Weilian Su, Yogesh Sankarasubramaniam and Erdal Cayirci *A Survey on Sensor Networks* IEEE Communications Magazine, Agosto, 2002. (p: 2)

[3] Ilker Demirkol, Cem Ersoy and Fatih Alagoz, *MAC Protocols for Wireless Sensor Networks: A survey* IEEE Communications Magazine, Abril, 2006. pp. 115-121. (p: 6)

[4] Ben Crawford *Gateway Design and Implementation*, United Kingdom, Septiembre 1995 (p: 40)

[5] Ian F. Akyildiz, Tomasso Melodia, Kaushik R. Chowdury *A survey on wireless multimedia sensor networks*, United States, 2006 Elsevier B.V. (p: 3)

[6] Miroslav Sveda, Roman Trachlik *ZigBee-to-Internet Interconection Architectures*, Proceedings of the Second International Workshop on Mobile Communications and Learning MCL 2007, Saint Luce, Martinique, MQ, IEEE CS, 2007, p. 6 (p: 41)

[7] Sangim Ahn and Kiwon Chong *Building a bridge for Heterogeneous Sensor Networks* IEEE Computer Society Washington, DC, USA, 2006, pp. 121 - 126 (p: 65)

[8] Víctor Hugo Romero Corral y Leocundo Aguilar Noriega *Gateway para Redes de Sensores Inalámbricas y Redes 802.11b* Facultad de Ciencias Químicas e Ingeniería, Universidad Autónoma de Baja California, Tijuana, Baja California (p: 47)

[9] *An Introduction to Wi-Fi*®, Digi International Inc. © 2007-2008 (p: 13)

[10] Technology Review *Ten Emerging Technologies that will change the world.*, February 2003 (p: 9)

[11] Ian F. Akyildiz, Xudong Wang, Kiyon, Inc. *A Survey on Wireless Mesh Networks*, Georgia Institute of Technology IEEE Radio Communications, Septiembre 2005 (p: 6)

[12] Stefano M. Faccin, Carl Wijting and Jarkko Kneckt, Ameya Damle. *Mesh WLan Networks: Concept and System Design*, IEEE Wireless Communications, Mag. Abril 2006 (p: 5)

[13] http://www.radioptica.com/Radio/wsn.asp?pag=2 (p: 2)

[14] Dynamic C: Integrated C development system for rabbit 4000 and 5000 microprocessors user‘s manual. (p: 57)

[15] http://fiji.eecs.harvard.edu/CodeBlue (p: 3)

[16] http://en.wikipedia.org/wiki/Wireless-Sensor-Networks (p: 1)

[17] MiniCore RCM5600W - C-Programmable Wi-Fi Core Module OEM Userś Manual 2009 Digi International Inc. (p: 54)

[18] www.rabbit.com (p: 54)

[19] Panasonic 2.4 GHz Low Power Module for the IEEE802.15.4 Standard Application Notes, Pags 6. (p: 42) (p: 60)

[20] Panasonic Industrial Company ZigBee Comm Module IEEE802.15.4, Performance Speci

cations, Summary, Doc Numero: PANASONICCOFS, Rev 0, Pags 2, 2005. (p: 42) (p: 59)

[21] Freescale Semiconductor, MC13192 2.4 GHz Low Power Transceiver for the IEEE 802.15.4 Standard, Rev. 3.2, Pags 24, 2005. (p: 45) (p: 60)

[22] Freescale Semiconductor BeeKit Wireless Connectivity Toolkit, Userś Guide, BKWCTKUG Rev. 1.9, Pags 47, Sep. 2009. (p: 46) (p: 62)

[23] http://www.freescale.com/codewarrior . (p: 51) (p: 63)

[24] Freescale Semiconductor Codewarrior Development Tools, Document Number: CODEWARRIORPRDS REV 0, 2007. (p: 51) (p: 63)

[25] IEEE Computer Society IEEE Std 802.15.4, Revision of IEEE Std 802.15.4- 2006, Septiembre, 2006. pags 323. (p: 22) (p: 26)

[26] Modelo UML del estándar IEEE 802.15.4 Septiembre, 2006. pags 323. (p: 22) (p: 26)

[27] IEEE Computer Society IEEE Std 802.1D, Revision of IEEE Std 802.1D- 2004, Junio, 2004. (p: 40)

[28] IEEE Computer Society IEEE Std 802.11, Revision of IEEE Std 802.11- 1999, Junio, 2007. (p: 22) (p: 13)

[29] Information processing systems, Open Systems Interconnection, Basic Reference Model, Part 2: Security Architecture ISO 7498-2, International Standards (p: 35)

[30] ISO/IEC 7498-1. Information technology, Open Systems Interconnection, Basic Reference Model: The Basic Model, ISO/IEC 7498-1, International Standards Organization, 1994, (p: 27)

[31] www.forrester.com/ (p: 14)

[32] NIST-1451, 2008 (p: 8)

[33] Artaud et al, 2004 (p: 8)

[34] IEEE P1451.5 Project, 2008 (p: 8)