Universidad Autónoma de Baja California

Facultad de Ingeniería, Arquitectura y Diseño

Maestría y Doctorado en Ciencias e Ingeniería

Protocolo de transporte para el control de congestión, bajo diseño Cross-Layer, consciente del QoS de las aplicaciones en redes inalámbricas IEEE 802.15.4

TESIS

que para obtener el grado de DOCTOR EN CIENCIAS

Presenta:

RAYMUNDO BUENROSTRO MARISCAL

Director de Tesis:

DR. JUAN IVÁN NIETO HIPÓLITO

Ensenada, Baja California, México Julio de 2014

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA

FACULTAD DE INGENIERÍA, ARQUITECTURA Y DISEÑO

Protocolo de transporte para el control de congestión, bajo diseño Cross-Layer, consciente del QoS de las aplicaciones en redes inalámbricas IEEE 802.15.4

TESIS

Que para obtener el grado de Doctor en Ciencias presenta:

Raymundo Buenrostro Mariscal

Aprobada por:

Dr. Juan Iván Nieto Hipólito
Director de tesis

Dr. Juan Antonio Guerrero Ibáñez
Miembro del comité

Dra. Mabel Vázquez Briseño
Miembro del comité

Dra. Larysa Burtseva
Miembro del comité

Dr. Juan de Dios Sánchez López
Miembro del comité

Ensenada Baja California, México, Julio de 2014

RESUMEN de la Tesis de Raymundo Buenrostro Mariscal, presentada como requisito parcial para la obtención del grado de DOCTOR EN CIENCIAS. Ensenada, Baja California, México. Julio 2014.

Protocolo de transporte para el control de congestión, bajo diseño Cross-Layer, consciente del QoS de las aplicaciones en redes inalámbricas IEEE 802.15.4

Resumen aprobado por:

Dr. Juan Iván Nieto Hipólito
Director de tesis

La utilización de las redes inalámbricas de sensores (WSNs) ha evolucionado rápidamente; hoy se proponen para ser usadas en el entorno de las personas. Por ejemplo: la vigilancia de la salud de forma remota, donde las personas pueden portar múltiples dispositivos sensores que recolectan sus signos vitales y los transmiten inalámbricamente a un equipo final con el propósito de analizarlos. Bajo este escenario, las WSNs tienen un gran potencial para transformar la manera en la que se pueden prestar los servicios de salud. Sin embargo, la adopción práctica de este escenario, requiere que las WSNs superen el desafío técnico de proveer *una comunicación de datos eficiente* por la red. Ya que la pérdida de información o la entrega tardía de la misma, pueden repercutir en la salud de las personas.
El problema de la *congestión de la red* que se presenta en las WSNs, afecta fuertemente este desafío; debido a que la congestión se traduce en pérdida de datos y en el incremento del tiempo para entregarlos. Impactando negativamente el desempeño global de la red y la calidad de servicio (QoS), que ésta puede ofrecer a las aplicaciones. Para resolver este problema se propone el protocolo de transporte para el control de congestión (*PCCQ*). Diseñado para controlar la congestión de forma distribuida, en cada nodo de la red; dinámica, de acuerdo al grado de congestión experimentado por el nodo y a la QoS de la aplicación. *PCCQ* está compuesto por cuatro módulos integrados bajo el diseño Cross-Layer: *Priorización de datos, Detección, Notificación y Resolución de la congestión.* En consecuencia, *PCCQ* puede contribuir a lograr el desafío técnico planteado, permitiendo materializar los beneficios potenciales de las WSNs en la vigilancia de la salud de forma remota.

Palabras clave: Congestión, redes inalámbricas de sensores, Cross-Layer, IEEE 802.15.4, calidad de servicio, supervisión de la salud.

DEDICATORIA

A mi esposa **Liva** e hijas **Karla** y **Ximena**:
Gracias por su amor, apoyo y confianza que siempre me han brindado y que fue fundamental para la culminación de este trabajo.

A mis padres:
Con amor y respeto les dedico este trabajo; gracias por su comprensión y apoyo que siempre me han brindado.

AGRADECIMIENTOS

Agradezco a la Universidad Autónoma de Baja California y a la Universidad de Colima por el apoyo que me brindaron durante la realización de mis estudios y del trabajo de tesis.

Al Dr. Juan Iván Nieto Hipólito, por la confianza y apoyo para la realización del presente trabajo; pero sobre todo por su amistad.

Al Dr. Juan Antonio Guerrero Ibáñez, por su apoyo y consejos para llevar acabo el desarrollo de la investigación. Especialmente agradezco su apoyo personal por atender las cuestiones laborales durante mi ausencia.

A los miembros del comité de tesis, por sus aportaciones y comentarios que enriquecieron este trabajo de tesis.

A todos mis profesores y amigos que de una u otra manera colaboraron a la conclusión de mi trabajo.

ÍNDICE GENERAL

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

CAPÍTULO 1

INTRODUCCIÓN

Las redes inalámbricas de sensores (WSNs, Wireless Sensor Network) se han convertido en un paradigma para la recolección de datos de forma remota y representan el siguiente paso en las comunicaciones inalámbrica "portables" [Lorincz et al., 2004], [Hanson et al., 2009], ya que ofrecen múltiples ventajas sobre las redes tradicionales de comunicación, tales como: bajo costo de implementación, mínimo mantenimiento, fácil despliegue, bajo consumo de energía, dispositivos de tamaño reducido, movilidad, entre otras (Sec.2.1). Esto ha convertido a las WSNs en una de las tecnologías más prometedoras para su uso en el entorno de las personas. Por ejemplo, en el campo del entretenimiento, acondicionamiento físico y para el cuidado de la salud (Healthcare). Al respecto del cuidado de la salud, el uso de las WSNs para habilitar la vigilancia de la salud de forma remota, es una de los campos más estudiados a la fecha [Darwish and Hassanien, 2011]. Este escenario contempla múltiple dispositivos sensores colocados en, cerca de, o dentro de las personas, los cuales tiene la capacidad de recolectar sus signos vitales y transmitirlos inalámbricamente a un equipo final para su análisis.

Bajo este escenario, las WSNs tienen un enorme potencial para transformar la forma en

que las personas interactúan con, y se benefician, de las tecnologías de la información; sin embargo, para su adopción práctica debe superar desafíos técnicos y sociales.

Al respecto de los desafíos técnicos, uno de ellos es proveer una ***comunicación eficiente*** entre las diferentes entidades de la red; por ejemplo, entre los nodos sensores y el nodo final de la red (o Sink). Este desafío se define como la capacidad de la red para soportar ciertos parámetros de transmisión de datos: retardo de transmisión, paquetes perdidos y paquetes entregados al destino. El valor que alcanzan estos parámetros definen el nivel de calidad de servicio de la red (QoS, Quality of Service); por lo tanto, entre mayor nivel de QoS pueda ofrecer una red, significa que ésta puede cubrir una mayor cantidad de aplicaciones.

Este desafío se vuelve sumamente importante en el cuidado de la salud, ya que son aplicaciones de carácter crítico, y cualquier pérdida de información o su entrega fuera de tiempo puede repercutir en la salud de las personas. Además, dado que las personas pueden portar diferentes sensores, éstos generan diferentes tipos y volúmenes de datos, formando un tráfico heterogéneo. Lo anterior, demandan mejores y diferentes niveles de QoS de la WSN, para poder instrumentar este tipo de aplicaciones (Sec.2.1.2).

Por su parte, las WSNs tienen limitaciones importantes para cumplir con este desafío. Entre ellas, su naturaleza inalámbrica, que lo convierte en un medio de transmisión inestable y propenso a fallas [Tian et al., 2005]. Otra limitación es el medio de comunicación compartido que utilizan, donde todos los nodos contienden por el acceso al medio único de transmisión; lo cual requiere de métodos de acceso eficientes, de lo contrario pueden producirse fallas de transmisión. En ambos casos, se pueden producir perdida de datos o retraso en la entrega de los mismos, afectando la QoS que las redes pueden ofrecer y por ende el objetivo de una *comunicación eficiente*.

Aunado a las limitaciones de las WSNs, existe el problema de la *congestión de la red*, que incrementan las fallas de transmisión de las WSNs [Wang et al., 2007]. La congestión puede producirse en dos casos, cuando la capacidad de almacenamiento de un nodo es rebasada; y cuando la capacidad de comunicación de la red es superada. En el primer caso, el nodo recibe más paquetes de datos que los que puede manejar; por lo cual, cada nuevo paquete que recibe debe "tirarlo". Además, se incrementa el tiempo de envío de los paquetes que tiene almacenado. En el segundo caso, una red conges-

tionada puede provocar que un nodo no pueda transmitir, y si lo hace, tenga una alta probabilidad de *colisionar* con otras transmisiones, provocando que ambos paquetes se pierdan. Este problema se ejemplifica en el esquema de *causa-efecto* de la figura 1.1. Cuando la red presenta el problema de congestión, ya sea en el nodo o en la red, causa la pérdida de paquetes de datos de las diferentes aplicaciones. Lo cual, tiene un efecto negativo en el desempeño global de la red, y en consecuencia, en la QoS entregada a las aplicaciones; ya que incrementa la cantidad de paquetes perdidos, retrasa la entrega de datos y reduce la información entregada al nodo destino [Malindi, 2011].

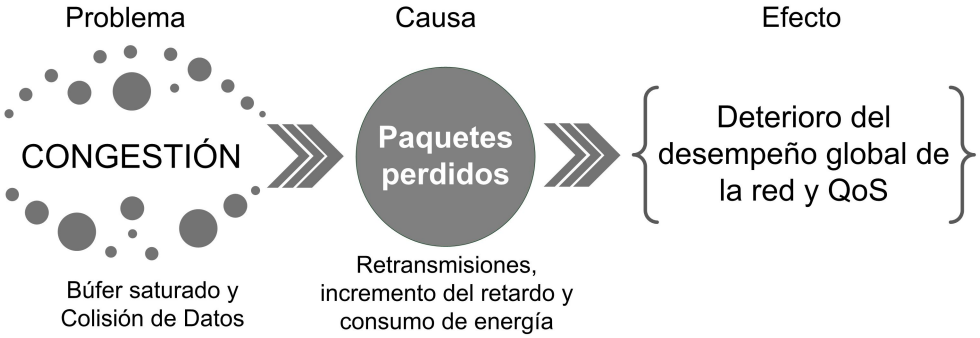

Figura 1.1: Efecto de la Congestión en las WSNs.

Por lo anterior, resolver el problema de la *congestión* es un reto obligado, si deseamos materializar los beneficios potenciales de las WSNs en la vigilancia de la salud de forma remota.

Una forma de afrontar el problema de la *congestión* es mediante la incorporación de mecanismos que impidan que se presente o que controlen sus efectos. Estos mecanismos de solución pueden estar integrados dentro de un protocolos de comunicaciones o distribuidos en varios protocolos. A la fecha existen diferentes propuestas de solución, principalmente para redes cableadas o inalámbricas convencionales. Sin embargo, como lo argumentan varios autores, estas propuestas no son una opción adecuada para las WSNs; ya que éstas tienen características únicas y diferentes a las redes mencionadas [Vuran and Akyildiz, 2010], [Rathnayaka et al., 2010], [min LIN et al., 2011]. Además, los protocolos propuestos para las WSNs, actualmente, en su mayoría están diseñados para aplicaciones con requerimientos de comunicaciones más sencillos y centrados en el ahorro de energía; que no consideran las necesidades de QoS de las aplicaciones y

mucho menos para el cuidado de la salud [Sharif et al., 2010], [Rahman et al., 2008b]. Diferentes investigaciones han mostrado que el diseño Cross-Layer mejora el desempeño de los protocolos de comunicaciones, mediante la interacción de diferentes capas de la "pila" de protocolos de red: capa de Transporte, de Red, de acceso al medio (MAC, Medium Access Control) y Física (correspondientes al modelo de referencia OSI). Esta capas pueden intercambiar información para trabajar con los mismos objetivos de optimización. A la fecha, los protocolos de control de congestión, propuestos para las WSNs, raramente consideran este método de diseño; o utilizan diferentes capas en su solución, obteniendo diferentes resultados de optimización [Fu et al., 2014], [Edirisinghe and Zaslavsky, 2013], [Sridevi and Usha, 2011].

Por lo anteriormente expuesto, nuestro objetivo es crear un protocolo para el control de congestión en redes WSNs, que considere los requerimientos de las aplicaciones críticas y heterogéneas, como el cuidado de la salud. Que además, utilice el diseño Cross-Layer en su operación, para optimizar los recursos de la red y que alcance el mayor desempeño posible en los niveles de QoS ofrecidos.

1.1 Planteamiento del Problema

Las aplicaciones críticas que utilizan una WSN, como las del cuidado de la salud, necesitan una red que haga una entrega confiable y oportuna de los datos de extremo a extremo. Para ello, es necesario proveer un buen desempeño de la red de comunicaciones, que alcance valores satisfactorios en los parámetros de transmisión de datos (retardo de comunicación, paquetes perdidos y paquetes entregados). Con lo cual, se logre el desafío de una *comunicación eficiente*, que alcance los mejores niveles de QoS. Sin embargo, existe en las redes de comunicaciones el problema de la *congestión*, que afectan el logro de los objetivos anteriores; sobre todo en las redes WSNs, como se mencionó en la introducción y se esquematiza en la figura 1.1. Por lo tanto, La problemática del trabajo de investigación es:

> El *control de la congestión* en redes WSNs con aplicaciones de diferente QoS y que considere un diseño Cross-Layer.

Bajo esta problemática se plantea las siguientes preguntas de investigación:

1. ¿Qué módulos deben formar el control de la congestión, que consideren el QoS de las aplicaciones?

2. ¿Qué capas de la "pila" de protocolos de red deben interactúar en el diseño Cross-Layer?

3. ¿Qué parámetros de éstas capas se deben compartir en el diseño Cross-layer para instrumentar la solución?

1.2 Objetivos

1.2.1 Objetivo general

Desarrollar un protocolo de Transporte para el control de congestión, basado en en el diseño Cross-Layer, para redes WSNs-IEEE 802.15.4, consciente de la QoS de las aplicaciones y del grado de congestión de la red.

1.2.2 Objetivos particulares

1. Establecer el marco de actuación del trabajo de investigación, que incluye el *diseño del escenario* (requerimientos) y la *determinación de las métricas de evaluación* para el análisis del desempeño del protocolo.

2. *Delimitar la información y funcionalidades* principales de la capa de transporte y subcapa MAC, que serán incluidas en la propuesta de diseño Cross-Layer.

3. *Desarrollar el protocolo* de transporte Cross-Layer bajo las especificaciones de diseño definidas y utilizando el estándar IEEE 802.15.4.

4. Desarrollar el marco de trabajo para *evaluar el desempeño* del protocolo de transporte, que incluya una red de sensores y el estándar IEEE 802.15.4.

1.3 Organización de la Tesis

Este trabajo de tesis se estructura en seis capítulos de la siguiente manera.
En el capítulo 1 se presenta la introducción al trabajo de investigación, la problemática que se pretende resolver, y los objetivos que persiguen con los resultados de este trabajo.
En el capítulo 2 se detallan los temas e información asociados a nuestra problemática y objetivos planteados, para construir el *marco teórico* que sustente el presente trabajo. Entre los temas se presenta, las WSNs y sus aplicaciones para el cuidado de la salud, el impacto de la *congestión* en la QoS y su control, las estrategias de diseño Cross-Layer y la descripción técnica del estándar IEEE 802.15.4.
En el capítulo 3 se hace un estudio del estado del arte de las soluciones propuestas al problema del *Control de Congestión* en las redes WSNs. Este estudio nos permitió crear una clasificación de los mecanismos utilizados para resolver este problema y proponer las *consideraciones y retos* más importantes que un protocolo de transporte debe cubrir para el control de la congestión en las redes WSNs.
En el capítulo 4 se presenta la propuesta para resolver la problemática del *control de congestión* en las redes WSNs, mediante el protocolo de Transporte *PCCQ*. Por ello, en este capítulo se describe la arquitectura de diseño, el modelo de sistema y los módulos que componen el protocolo *PCCQ*.
En el capítulo 5 se realiza la evaluación del desempeño del protocolo *PCCQ*. Para lo cual, se describe el escenario y los casos de estudio propuestos para la evaluación; así como la implementación de los diferentes elementos que integran la evaluación. Concluyendo con la presentación de los resultados de cada caso de estudio y su análisis.
Por último, en el capítulo 6 se presentan las conclusiones generales del trabajo de tesis desarrollado; enfatizando aquellos puntos que representan las aportaciones más importantes. Así mismo, se discute el trabajo futuro que puede ser desarrollado dentro del contexto de este trabajo.

CAPÍTULO 2

MARCO TEÓRICO

En este capítulo se presentan la temas e información asociados a nuestra problemática y objetivos; con la intención de exponer los conceptos, las teorías y el contexto donde se desarrolla nuestro trabajo. Que además, respalde y fundamente nuestra investigación. Así mismo, este capítulo busca detallar la problemática de la congestión en la redes WSNs, para el uso de aplicaciones médicas en la monitorización remota de pacientes. Específicamente, el efecto de la congestión en la calidad de servicio ofrecida en la comunicación de datos entre los dispositivos de la red.

2.1 Red Inalámbrica de Sensores (WSN)

Las Redes Inalámbricas de Sensores (WSNs, Wireless Sensor Networks) son una tecnología compuestas por dispositivos electrónicos capaces de recolectar información del entorno, procesarla localmente y transmitirla de forma inalámbrica hacia un punto final. Las WSNs se caracterizan por su facilidad de despliegue, bajo costo, mínimo mantenimiento y por contar con dispositivos pequeños de bajo consumo de energía [Fernández et al., 2009]. La figura 2.1 muestra algunos dispositivos usados en las redes WSN, en particular usados para el proyecto de aplicaciones médicas CodeBlue [Shnayder et al., 2005]. Estos dispositivos consisten en general de un módulo sensor, un procesador de datos, fuente de energía y componentes de comunicación.

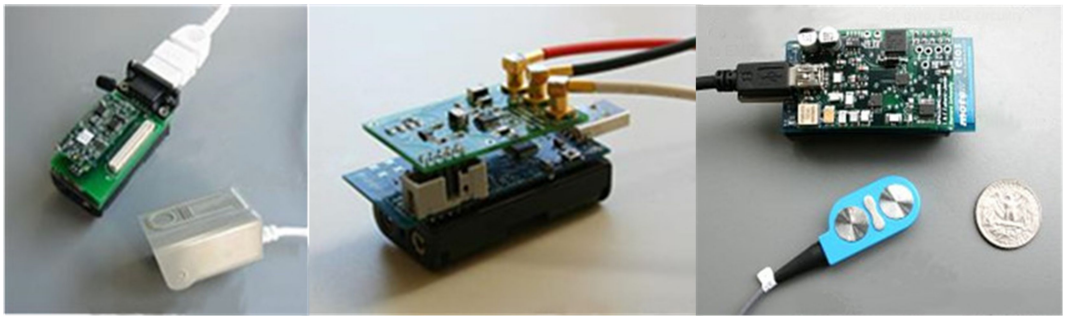

Figura 2.1: Dispositivos electrónicos de una WSN.

El escenario tradicional de una WSN esta compuesto por un sólo dispositivo final (llamado Sink) y múltiples dispositivos, que pueden ser fijos o móviles; desplegados como se presenta en la figura 2.2. El modelo seguido por las aplicaciones es el siguiente: realizar unas serie de mediciones de variables de interés y transferirlas, típicamente en varios saltos, a través de los dispositivos hasta el Sink. Adicionalmente, el Sink esta conectado a una red externa para extender la cobertura de la red y poder llevar la información hasta un centro de procesamiento de mayor capacidad.

Debido a que los dispositivos de las WSN tienen recursos limitados de energía, procesamiento y de almacenamiento; las aplicaciones y algoritmos que pueden ejecutar deben ser simples y conscientes de los recursos disponibles. Esto limita la posibilidad de utilizar los desarrollos existentes de las redes convencionales, como las Redes Inalámbricas de Area Local (WLANs, Wireless Local Area Networks) o redes cableadas, las cuales

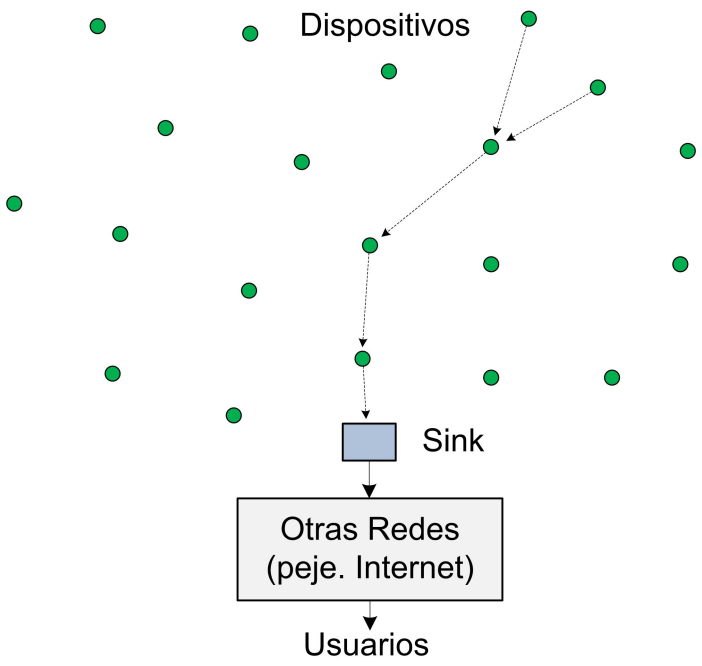

Figura 2.2: Escenario tradicional de una WSN.

consideran dispositivos de mayor capacidad.

Derivado de lo anterior, el proceso de estandarización en el ámbito de las WSN ha sido un proceso muy activo en los últimos años. Un resultado importante está representado por el estándar IEEE 802.15.4, que es un sistema de comunicaciones de corto alcance prevista para aplicaciones con necesidades de bajas tasas de datos y uso optimizado de los recursos en Redes inalámbricas de Área Personal (WPANs, Wireless Personal Area Networks) [LAN/MAN Standard, 2006]. Las principales características del estándar 802.15.4 se retoman en la Sección 2.5; ya que es seleccionado como base para desarrollar las WSNs en nuestro trabajo de investigación.

2.1.1 Aplicaciones de las WSN para la supervisión de la salud

Las WSNs pueden contar con dispositivos que incluyen varios tipos de sensores tales como: de movimiento, térmicos, visuales, acústicos, entre otros; los cuales son capaces de monitorear una amplia variedad de condiciones ambientales. Inicialmente las WSNs fueron diseñadas con propósitos militares para la monitorización de variables de campo

en tiempo real; por ejemplo, temperatura, movimiento, humedad, contaminantes, entre otras, necesarias para la toma de decisiones. Las WSNs han sido un tema de investigación muy activo en varias partes del mundo; especialmente con los avances tecnológicos de la microelectrónica y las comunicaciones inalámbricas. Lo cual ha facilitado el desarrollo de sensores más inteligentes, extremadamente pequeños y de bajo costo [Yick et al., 2008]. Lo anterior hace que las WSNs tengan un gran potencial y sean propuestas para aplicarse en diferentes escenarios [Akyildiz et al., 2002], [Castillo-Effer et al., 2004], [Abed et al., 2010], [Cervantes, 2014], tales como :

- Sistemas de monitorización ambiental
- Sistemas de seguridad y vigilancia
- Sistemas de automatización de edificios
- Sistemas de control de procesos industriales
- Sistemas inteligentes de transporte
- Sistemas de aplicación médica

Los escenarios de aplicación antes mencionados son sólo algunos de las posibilidades de utilización de las WSNs; sin embargo, existe un amplio abanico de posibilidades y cada día se siguen experimentando con nuevos servicios. Nuestro trabajo se centra en los escenarios de aplicación médica, específicamente para supervisar la salud de personas; por ello, a continuación se detalla este tipo de escenarios y sus requerimientos.

WSN para la supervisión de la salud

En los últimos años se ha incrementado los trabajos que exploran el uso de las WSNs en aplicaciones médicas para mejorar los servicios de salud y de vigilancia existentes; especialmente para los ancianos, niños y enfermos. El uso de las WSNs para supervisar el estado de salud de forma remota es una de las aplicaciones que más atención ha tenido en los últimos años [Egbogah and Fapojuwo, 2011], [Latré et al., 2011], [Rajba et al., 2013]. La arquitectura general para éste tipo de aplicaciones incluye un conjunto de dispositivos colocados en las personas para recuperar signos vitales, tales como: temperatura corporal, saturación de oxígeno, ritmo cardíaco, electrocardiograma (ECG,

Electrocardiogram), entre otros (Fig.2.3). Sensores de aceleración, para detectar desplazamientos y caídas, puede ser incluidos junto a estos signos vitales. La información recolectada es transmitida inalámbricamente a otros dispositivos concentradores, para retransmitirla hasta un punto central (Sink); donde puede ser almacenada para tener un historial de salud de cada persona, y/o analizada por personal médico. La figura 2.4 muestra la arquitectura típica de una WSN para este tipo de aplicaciones.

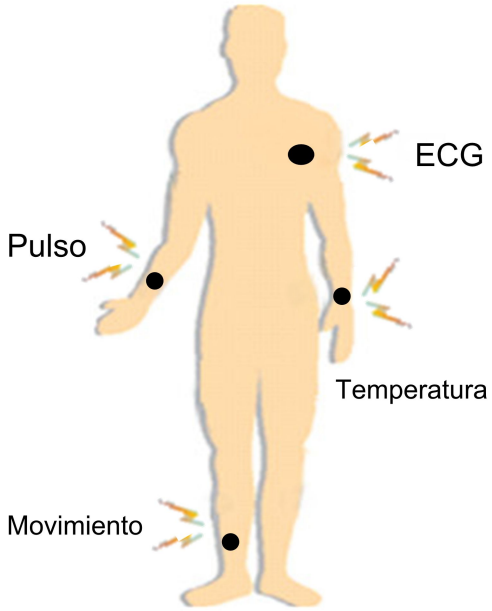

Figura 2.3: Dispositivos sensores en las personas.

Hay varios beneficios que se obtienen con estos sistemas de supervisión de la salud. Para empezar, la capacidad de monitoreo remoto es el principal beneficio de los sistemas de salud. Estos sistemas contribuyen a la prevención de la salud y a la atención a urgencias médicas. Con el seguimiento remoto, la identificación de las condiciones de emergencia para los pacientes en riesgo será más fácil; y las personas con diferentes grados de discapacidad cognitiva y física podrán tener una vida más independiente. Por ejemplo, para el cuidado de personas de edad avanzada, se puede tener vigilancia durante las 24 horas sin necesidad de hospitalizar el paciente o tener personal médico en casa. En lugar de ello, se puede utilizar sensores portátiles para monitorizar sus signos vitales y movimientos, para avisar al personal médico o familiares cuando se

presenten emergencias vía una red de cobertura amplia tipo Internet o telefonía celular (Fig.2.4) [Ameen et al., 2008] [Akyildiz et al., 2008]. Además, el monitoreo constante y permanente de parámetros críticos de las personas, permite identificar tempranamente enfermedades potenciales, o estudiar el comportamiento de pacientes enfermos para responder eficientemente a su tratamiento.

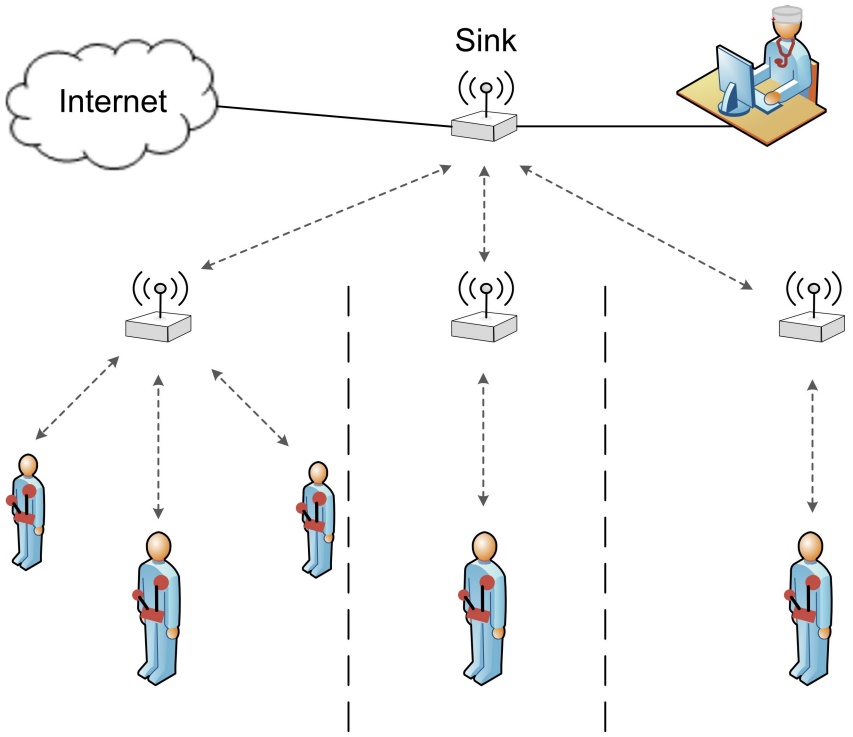

Figura 2.4: Escenario de una WSN para la supervisión de la salud.

Los beneficios mencionados son sólo algunos de los más evidentes. Sin embargo, existen muchos más. Con este fin se mencionan algunos proyectos destacados, que utilizan las WSNs en el área de la salud; a manera de referencias y para mostrar el nivel de interés en el uso de las WSNs:

- MobiHealth: es un proyecto Europeo desarrollado para monitorizar signos vitales basados en una red de área corporal (BAN, Body Area Network) y una plataforma de servicios de m-salud [Konstantas and Herzog, 2003].

- CustoMed: fue desarrollado por la Universidad de California. Su arquitectura

se basa en una red que soporta diferentes sensores portables, con capacidades de cómputo para ejecutar la detección de eventos, alertas y la comunicación con diferentes servicios de información médica [Jafari et al., 2005].

- Alarm-Net: fue desarrollado por la Universidad de Virginia. Alarm-Net es una red de monitorización de vida asistida y residencial para la investigación sanitaria inteligente; que permite un seguimiento continuo de los ancianos o de las personas que necesitan asistencia médica [Wood et al., 2008].

- CodeBlue: fue desarrollado por la Universidad de Harvard. Este proyecto explora las aplicaciones de la tecnología WSNs en una gama de aplicaciones médicas, incluyendo: pre-hospitalaria, atención hospitalaria de emergencia, respuesta a desastres, rehabilitación de pacientes, accidente cerebro-vascular, y la localización de personas [Shnayder et al., 2005].

- Wearable Health Monitoring Systems: es una arquitectura de WSN para la monitorización ambulatoria del estado de salud; desarrollado por la Universidad de Alabama. La WSN incluye varios sensores de movimiento que supervisan la actividad global del usuario y un sensor de ECG para supervisar la actividad del corazón [Otto et al., 2005].

Los proyecto mencionados ofrecen diferentes niveles y tipos de servicios; además, utilizan diferentes tecnologías tanto para el desarrollo de las WSNs como de las redes de cobertura amplia. Sin embargo, hay un esfuerzo significativo en los diferentes proyectos de investigación por la monitorización cardíaca. Especialmente los sistemas de medición de ECG móvil están ganando importancia [Alemdar and Ersoy, 2010].
La diversidad de aplicaciones puede ser tan amplia como las necesidades de atención lo requieran. Sin embargo, las capacidades de los dispositivos y sistemas que forman la WSN deben soportarlas. En consecuencia, las aplicaciones crean las necesidades de operación de las WSNs y estas necesidades definen la selección de los dispositivos, los protocolos de comunicación y las técnicas de transmisión. En la sección 2.1.2 se presentan los requisitos más importantes para las supervisión de la salud.

2.1.2 Requerimientos para la supervisión de la salud

Cuando se proponen las WSNs para la monitorización de pacientes de forma remota, el desempeño en la transmisión de los datos y los requerimientos de los dispositivos son mucho más importantes que en otras aplicaciones [Malindi, 2011], [Skorin-Kapov and Matijasevic, 2010], [Rahim et al., 2011]. Considerando que:

1. Los datos son utilizados para la atención médica personal, y esto lo vuelve una aplicación crítica. Ya que cualquier dato perdido o demorado puede tener serias implicaciones en la salud del paciente.

2. Como se expuso en la sección 2.1.1, las WSNs puede llevar diferente tipos de aplicaciones con diferentes requerimientos de comunicación. Por ejemplo, transmitir el valor de temperatura corporal necesita menos capacidad de la red y es más tolerable al retardo extremo a extremo, comparado con la señal ECG; la cual necesita enviar más datos por segundo para reconstruir la señal. Convirtiendo a las WSNs en redes con aplicaciones heterogéneas.

3. debido a que los dispositivos deben colocarse en las personas y su entorno, éstos deben cubrir ciertos requerimientos de tamaño, uso de energía y de mantenimiento.

4. Las personas necesitan movilidad e independencia durante los procesos de monitorización; por ello, los dispositivos y elementos de la WSN deben privilegiar ésta premisa ante los objetivos de monitorización.

Por lo tanto, cuando se diseñan y construyen WSNs para aplicaciones médicas, es necesario tomar en cuenta estas consideraciones para determinar las características de los dispositivos, y los requerimientos de transmisión de las señales fisiológicas por la red. Estas consideraciones son descritas a continuación.

Señales fisiológicos y sus requerimientos

Como se ha mencionado las aplicaciones crean las necesidades de operación de las WSNs. En este sentido, conocer el tipo de tráfico que generan las aplicaciones médicas es importante; ya que ésto determina el tipo de atención que los protocolos de comunicación

deben ofrecerles en su transporte dentro de la WSN. Para el caso de la supervisión de la salud, el tráfico es generado por los sensores que recolectan las señales fisiológicas de los pacientes. Para conocer el tipo de tráfico es necesario analizar ciertos factores tales como: tipo, cantidad y frecuencia de los datos que resultan de cada señal fisiológica. Existen diferentes señales fisiológicas que pueden ser monitorizadas (Sec.2.1.1), sin embargo, con el fin de entender las necesidades impuestas por la aplicación se selecciona la señal ECG como ejemplo de análisis (estudios similares se pueden hacer a cualquier señal fisiológica). Dichas señales se necesitan estudiar desde el punto de vista *eléctrico* y desde el punto de vista de la *aplicación médica*.

La señal ECG es el resultado del registro de las señales eléctricas que produce el corazón en el rango de frecuencia de 0.01 a 250 Hz (Fig.2.5a y Tabla 2.1) [Arnon et al., 2003]. Estas señales eléctricas son de tipo analógico, y son generadas continuamente; por lo tanto, deben ser periódicamente muestreadas, con el fin de digitalizarlas (Fig.2.5b). La frecuencia de muestreo y los métodos de digitalización juegan un papel importante en determinar las características del tráfico resultante [Cypher et al., 2006]. Por ejemplo, si las señales eléctricas del corazón se muestrean a 1250 muestras por segundo (más de 4 veces la frecuencia máxima) y cada muestra tiene un tamaño de 12 bits; entonces, se espera que el flujo de datos de la aplicación ECG genere un tráfico de 15000 bits por segundo (bps). Entonces los requerimientos de transmisión para la señal ECG, que la WSN debe soportar es de al menos 15 kbps. La Tabla 2.1 muestra los requerimientos de digitalización y de trasmisión de algunas señales fisiológicas [Latré et al., 2011], [Arnon et al., 2003]. Observe que las señales con más requerimientos de transmisión (o tasa de datos) son la Electromiografía (EMG) y el ECG. Por el contrario, la señal de temperatura corporal es la que menos bits por segundo requiere para su operación.

Las muestras digitalizadas deben agruparse o empaquetarse (Fig.2.5c) para ser transmitidas por el medio inalámbrico; función que le corresponde a los protocolos de comunicaciones utilizados en la WSN. Por ejemplo, para al caso del ECG y considerando paquetes de 80 bytes, resultarían 23.43 paquetes por segundo (p/s); los cuales deben ser enviados al Sink para reconstruir la señal ECG del paciente para su interpretación. Por el contrario, la señal de temperatura corporal sólo requiere del envío de un paquete de datos.

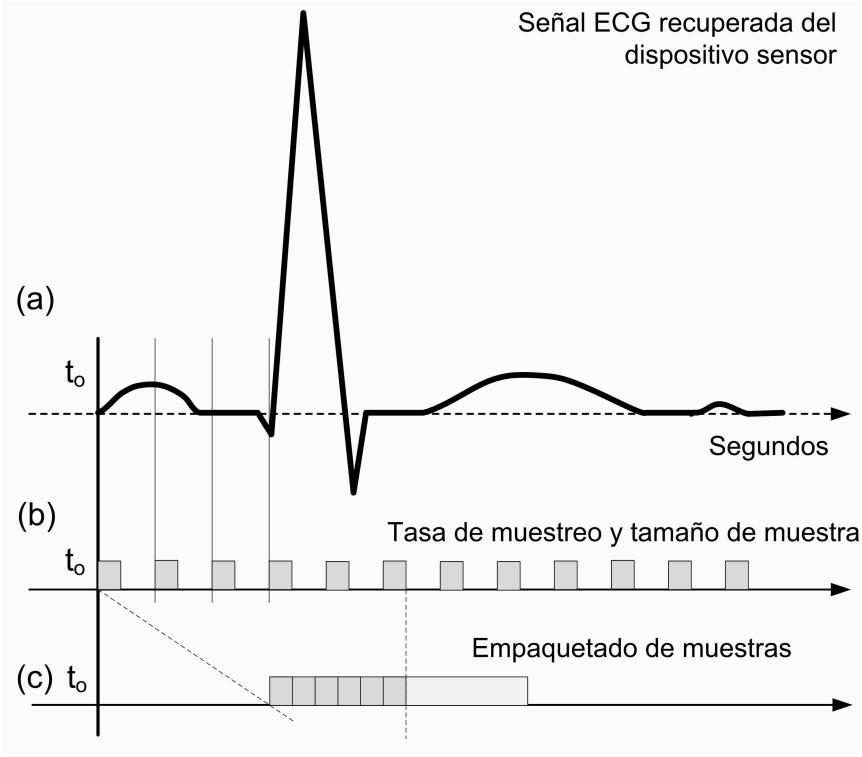

Figura 2.5: Señal ECG y su digitalización.

La alta heterogeneidad de las aplicaciones médicas provocan requerimientos de transmisión que varían fuertemente, desde un paquete hasta varios paquetes por segundo. Además, los datos pueden ser en "ráfagas", lo cual significa que existe momentos de alta transmisión; como el ECG, que envía más de 20 p/s en cada evento. Esta diferencia de requerimientos debe ser identificada por los protocolos de comunicaciones, con el fin de ofrecer mejores servicios de transmisión.

Desde el punto de vista de la aplicación, los sistemas tradicionales de supervisión de la salud requieren una monitorización permanente y periódica de las variables fisiológicas [Egbogah and Fapojuwo, 2011]. Por lo tanto, el tráfico se genera de igual forma durante todo el tiempo que incurre la supervisión médica. Lo anterior establece que la WSN debe manejar tráfico constantemente. Además, el tráfico que genera cada dispositivo de la red es dirigido hacia el Sink (Fig.2.4); lo cual hace que los dispositivos intermedios en el trayecto manejen una mayor cantidad de tráfico. Este tipo de comunicación se

Tabla 2.1: Tasa de datos generadas por las señales fisiológicas.

Señal Fisiológica	Ancho de Banda (Hz)	Muestreo (muestras/s)	Resolución (bits)	Tasa de datos (bps)
EMG	0 a 10,000	50,000	12	600,000
ECG	0.01 a 250	1250	12	15,000
Nivel de Glucosa	0 a 50	100	16	1600
F. Respiratoria	0.1 a 10	50	16	800
Ritmo cardíaco	0.4 a 5	25	24	600
Temperatura C.	0 a 1	5	16	80

conoce como flujo de tráfico "convergente". Esta forma de comunicación genera un "efecto embudo" que se va agravando conforme el tráfico es más cercano al Sink. El efecto embudo puede provocar que la WSN no pueda atender las demandas de transmisión de las aplicaciones en los niveles que lo necesitan.

Por el carácter crítico de los datos (como se menciono en la consideración 1) el usuario de la aplicación (personal médico o personal especializado) espera recibir los datos lo más rápido posible. Esto convierte a las aplicaciones de cuidado de la salud en aplicaciones en *tiempo real*. Por lo tanto, el retardo de transmisión de los datos hasta el Sink, es un importante parámetro que hay que tomar en cuenta. Sin embargo, en aplicaciones de cuidado de la salud, un sistema en tiempo real es considerado un sistema de tiempo real "relajado", en el cual un cierto nivel de latencia es permitida [Alemdar and Ersoy, 2010]. Ya que, identificar situaciones de emergencia como un ataque cardíaco o un desvanecimiento súbito en pocos segundos o incluso en minutos será suficiente para salvar vidas; considerando que sin ello, no serían identificadas en absoluto. Por lo tanto, la identificación en tiempo real para la toma de medidas en los sistemas de salud son algunos de los principales beneficios.

Como referencia de los niveles de *retardo* y *niveles de transmisión*, en el trabajo de Ikram se establece que el valor de retardo de transmisión de un ECG debe ser menor a 500 milisegundos (ms) y una tasa de transmisión mínima de 4 kbps [Ikram et al., 2007]. Por su parte, Gállego indica que una señal ECG genera una tasa de datos de 1 a 20 kbps y el retardo de transmisión debe ser menor a 1 segundo [Gallego et al., 2005].

En suma, los diferentes tipos de señales fisiológicas y el tipo de aplicación demandan diferentes requerimientos de transmisión, tanto de capacidad de envío como de calidad de la comunicación. Además, el carácter crítico de la aplicación exige reducir al máximo la demora en la entrega y evitar la perdida de datos (consideraciones 1 y 2, antes mencionadas). Los *requerimientos de comunicación* de las aplicaciones, son comúnmente llamados necesidades de Calidad de Servicio (QoS, Quality of Service) [Halsall, 2006] [Yigitel et al., 2011]. Las necesidades de QoS de cada aplicación deben ser atendidas por la red, con el fin de no afectar la calidad de los servicios. Desde el punto de vista de la red, el QoS se define como la capacidad de ésta para "soportar" los requerimientos de comunicación solicitados (QoS de la red).

Requerimientos de los nodos sensores

Considerando que los dispositivos que forman las WSNs deben trabajar en las personas y en su entorno (consideración 3), y que éstas necesitan independencia y movilidad (consideración 4); los dispositivos deben satisfacer ciertos requerimientos de diseño y operación, tales como [Darwish and Hassanien, 2011], [Rahim et al., 2011], [Ikram et al., 2007]:

- **Movilidad y portabilidad**: Para lograr la supervisión de la salud continua, los pacientes deben tener un cierto grado de confort, lo cual implica sistemas no-invasivos y discretos. Que permitan la movilidad e independencia de los pacientes para desplazarse libremente mientras son monitoreados. Esto exige dispositivos ligeros y de tamaño reducido. Además, la movilidad exige dispositivos con comunicación inalámbrica y de protocolos de comunicaciones que se adapten rápidamente a los cambios de la calidad del enlace.

- **Bajo consumo de potencia**: Bajos requerimientos de potencia para la operación de los dispositivos es muy importante, para ampliar la vida del dispositivo y evitar que el paciente tenga que cambiar el suministro de energía con frecuencia. Entonces, los componentes considerados en el diseño del nodo sensor deben consumir un mínimo de energía, en particular el módulo de comunicación inalámbrica.

- **Libre de mantenimiento**: uno de los objetivos de la supervisión de salud es el monitoreo remoto, por lo cual es necesario evitar dispositivos que necesiten de la presencia de personal técnico para su mantenimiento u operación. Dispositivos con mínimos niveles de mantenimiento y de fácil operación, son aspectos importante para implantar sistemas de salud realmente factibles.

Requerimientos de la red para la QoS

La alta heterogeneidad de las aplicaciones médicas y la naturaleza crítica de los datos recogidos, hacen que este tipo de aplicaciones tengan necesidades especiales de QoS en la comunicación de los datos por la red (Sec.2.1.2). Los requerimientos de QoS en la comunicación se pueden agrupar en tres rubros [Buenrostro et al., 2013]:

1. **Retardo en la comunicación**: En los ambientes de salud la disponibilidad de los datos es crítica, por ello se consideran de tiempo real. Entonces, las aplicaciones exigen de la red bajos tiempos de entrega de datos extremo a extremo. Además, la red debe asegurar los mejores tiempos de retardo a las aplicaciones más críticas; por ejemplo, la señal ECG requiere una mayor atención (mejor QoS) que el valor de temperatura corporal (Sec.2.1.2).

2. **Comunicaciones fiables**: Evitar la *perdida de datos* es un requisito en cualquier red de comunicaciones; sin embargo, en las redes para aplicaciones médicas es un requisito aún mayor. Ya que la pérdida de información puede tener serias implicaciones en la salud del paciente.

3. **Desempeño en la comunicación**: La cantidad de datos que una red puede entregar al destino es conocido como *desempeño de la red* (o Throughput). Un seguimiento eficiente de la salud exige un desempeño de la red del cien por ciento. Sin embargo, ésto se considera ideal, ya que las redes tienen capacidades limites que pueden ser rebasadas. Por ello, los protocolos de comunicación deben priorizar el tráfico de acuerdo a su importancia, con el fin de satisfacer los requerimientos de comunicación cuando sus capacidades están siendo rebasadas.

En conclusión, los requerimientos que exigen las aplicaciones de monitorización de la salud imponen diferentes retos al desarrollo de las WSNs, tanto en el diseño de los

dispositivos como en la calidad de la comunicación de los datos. En particular, nuestro trabajo se enfoca en lo relativo a los requerimientos de QoS en la transmisión de datos, llamados métricas QoS [Yigitel et al., 2011]: Retardo promedio de Transmisión de Paquete extremo a extremo (RTP), Tasa de Paquetes Perdidos (PER, Packet Error Rate) y tasa de paquetes entregados o Throughput; que corresponden a los tres rubros de los requerimientos de la red.

2.2 El efecto de la Congestión en la QoS que ofrecen las WSNs

El objetivo de cualquier red de comunicaciones es el trasporte de la información entre los dispositivos y/o aplicaciones dentro de la red; buscando que el transporte sea lo mas *eficientemente* posible. Esto es, que el cien por ciento de la información que generan las aplicaciones sea entregada al destino en tiempo y forma. Entonces, cualquier pérdida de información o retraso en su entrega afecta la eficiencia de la QoS que ofrecen las redes a las aplicaciones. A continuación se describen los retos de las WSNs para proporcionar los requerimientos de QoS; principalmente debido a los problemas de la **congestión** de la red, al **uso compartido del canal** de comunicaciones y a las características propias del **medio inalámbrico**.

2.2.1 La Congestión en las redes WSNs

Una red se dice congestionada cuando el tráfico de datos que recibe es superior al que puede manejar [Kurose and Ross, 2013]. La capacidad de una red esta supeditada a la capacidad del medio de comunicaciones y a los mecanismos utilizados en la transmisión de los datos. Por ejemplo, una WSN que utiliza el estándar IEEE 802.15.4 (Sec.2.5), tiene un medio inalámbrico con una capacidad de transmisión de 250 kbps; entonce, la congestión se presenta cuando el tráfico que recibe de todos los dispositivos de la red superan ésta capacidad. Cuando ésto sucede, la red empieza a perder los datos que no puede manejar o a incrementar el retardo de transmisión; afectando los requerimientos

de QoS (Sec.2.1.2) ofrecidos a las aplicaciones [Yunus et al., 2011].

Dos tipos de *congestión* ocurren en las WSNs [Wang et al., 2007]. El primer tipo es la *congestión a nivel del dispositivo* (o nodo de la red), que es común en la redes convencionales (Fig.2.6a). Ésta es causada por el desbordamiento del búfer (dispositivo de almacenamiento de datos) del nodo, cuando la cantidad de paquetes que llegan excede su capacidad de servicio o de envío. Cuando el búfer del nodo se satura ocasiona que los nuevos paquetes que arriban se *"tiren"*; esta pérdida se clasifica como *paquetes descartados*. Considerando la dirección del flujo de datos en las WSNs, tráfico "convergente", se espera que este problema se agudicé en los nodos intermedios cercanos al Sink (Fig.2.4); ya que ellos manejan mayor tráfico combinado de otros nodos [Rahman et al., 2008a].

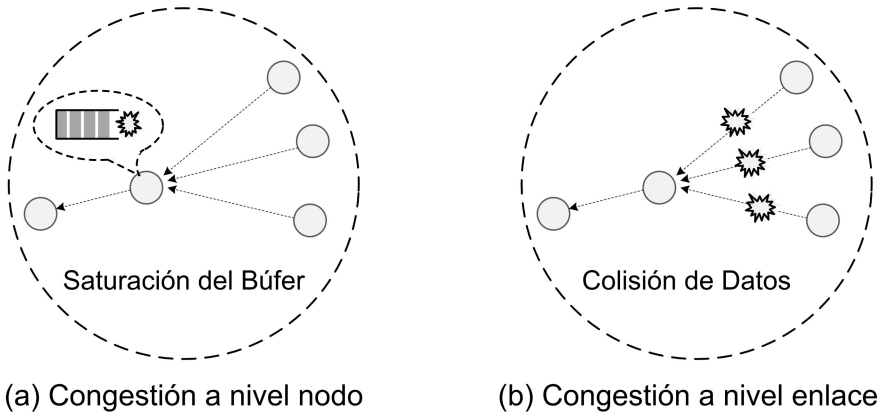

Figura 2.6: Congestión en las redes inalámbricas de sensores.

El segundo tipo de congestión es referido como *congestión a nivel de enlace* (Fig.2.6b). Éste se presenta cuando la capacidad límite del enlace o canal es rebasada por el trafico generado por las aplicaciones. En las WSNs este problema es crítico, ya que además de la capacidad del canal, éstas utilizan un *canal inalámbrico compartido* con todos los nodos de la red, mediante el uso de algún mecanismo de acceso al medio; lo cual trae consigo otros retos. Un canal inalámbrico es un medio de transmisión altamente inestable, ya que utiliza el aire libre como medio de transmisión. Por ello, el medio de comunicación esta sujeto a muchos factores incontrolables, tales como: condiciones climáticas, obstáculos urbanos, interferencias por múltiples trayectorias de la señal,

grandes objetos en movimiento, entre otros (Fig.2.7). Factores que afectan la potencia de la señal y en consecuencia la calidad de las comunicaciones, a diferencia de las redes que usan medios cableados [Tian et al., 2005]. Los problemas relacionados con el medio compartido es referente a su utilización; ya que sólo un nodo a la vez puede usarlo para transmitir. De lo contrario, si dos o más nodos transmiten simultáneamente las señales se interfieren y se destruyen; a ésto se le llama *colisión de datos* o perdidas de paquetes por colisión [Wang et al., 2007]. Para éste problema, las WSNs utilizan métodos de acceso por contienda (como el algoritmo CSMA-CA, ver sección 2.5.4). Sin embargo, estos tipos de métodos no son ciento por ciento efectivos y su eficiencia se reduce conforme el número de nodos en la red se incrementa, como se detalla en la secciones 2.2.2 y 2.5.4. En consecuencia, debido al medio inalámbrico, las WSNs pueden perder paquetes por problemas de capacidad, por el mecanismo de acceso al medio y por la naturaleza del canal inalámbrico.

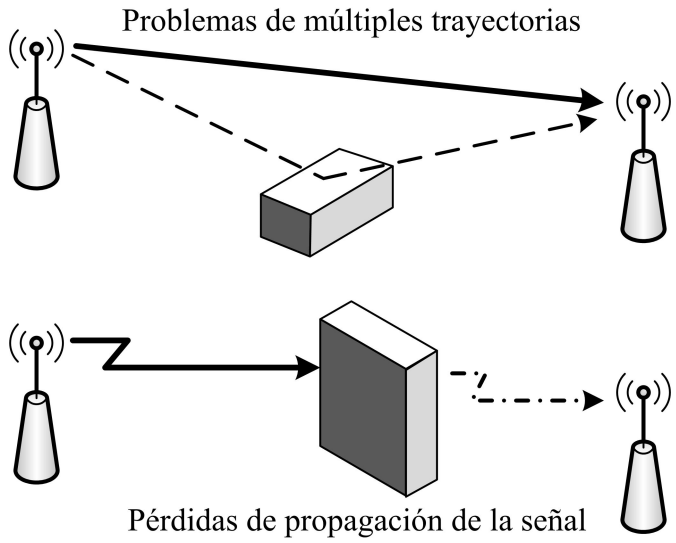

Figura 2.7: Interferencias y alteraciones de la señal.

Ambos tipos de congestión, ya sea en el nodo o en el enlace, provocan paquetes perdidos que afectan fuertemente la *confiabilidad de la red* (capacidad para transmitir paquetes con éxito), la *utilización eficiente del enlace* inalámbrico, el tiempo de transmisión de los datos y el consumo de energía del nodo (cada paquete que se pierde podría necesitar una retransmisión y por lo tanto un consumo adicional de energía). Bajo este contexto,

la *congestión* impacta negativamente el rendimiento global de las WSNs, y en consecuencia el nivel de QoS que pueden ofrecer a las aplicaciones; de ahí la importancia de incorporar mecanismos que controlen sus efectos o impidan que se presente.

2.2.2 Evaluación del impacto de la Congestión en la QoS

Para evaluar el impacto de la congestión en la QoS que ofrecen las redes WSNs, se realizó un experimento bajo un escenario de trabajo que involucra una WSN y una arquitectura de red en capas (Fig.2.8); la cual se enfoca en las capas de Aplicación, de Transporte y la subcapa MAC, del modelo de Interconexión de Sistemas Abiertos (OSI, Open Systems Interconnection) [ISO/IEC, 1994], ver secciones 2.4 y 2.5. La capa de Aplicación involucra los requerimientos de la aplicación médica para una señal ECG, los cuales fueron detallados en la sección 2.1.2 y en la Tabla 2.1. En lo referente a la capa de Transporte se utilizó el Protocolo de Control de Transmisión (TCP, Transmission Control Protocol) [Institute, 1981], como medio para establecer la comunicación extremo a extremo a nivel de aplicación, debido a sus prestaciones de control de flujo y confiabilidad en la entrega de paquetes. Para la subcapa MAC y capa Física se utilizó el estándar IEEE 802.15.4, en su versión de 2.4 GHz a una tasa de 250 kbps, que cubre las especificaciones de ambas entidades y es parte de nuestro interés de estudio. Además, el escenario incluye:

- Una WSN con 6 nodos transmisores y un nodo receptor, en configuración tipo estrella. Cada nodo transmite de forma independiente y constante durante todo el tiempo de simulación.
- Para generar los datos en cada nodo se utilizó una aplicación de tráfico de Tasa de Bits Constante (CBR, Constant Bit Rate). La cual se configuró a 18 kbps y 50 p/s, para simular una fuente de aplicación ECG. El valor de tráfico que cada nodo genera es mayor a 18kbps, puesto que TCP y el IEEE 802.15.4 le agregan información de control al paquete de datos.
- Se utilizó la herramienta de simulación de redes NS2 [Institute, 2008], que cuenta con módulos para el registro detallado de la operación de la red bajo el escenario propuesto.

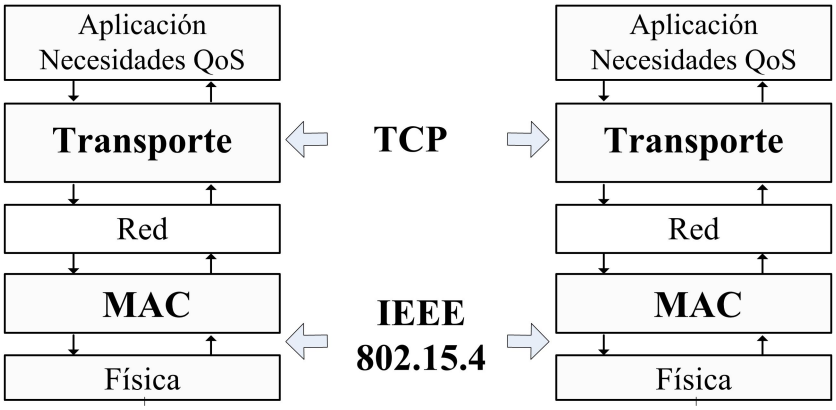

Figura 2.8: Arquitectura de la red del escenario de prueba.

Las métricas de evaluación QoS son: RTP, PER y TH de la red, como se definieron en la sección 2.1.2. El TH mide la cantidad de paquetes recibidos exitosamente en la capa de Transporte del nodo receptor y está dada en bits por segundo:

$$TH = \frac{Bits\,recibidos}{Ventana\,de\,medición} = (bps) \tag{2.1}$$

RTP mide el intervalo de tiempo desde que un paquete es enviado hasta el momento cuando es recibido en el nodo destino. RTP es medido extremo a extremo entre capas pares (Fig.2.8), por ejemplo, entre las capas TCP de cada nodo.

$$RTP = (Tiempo\,de\,Arribo - Tiempo\,de\,Salida) = (seg) \tag{2.2}$$

PER se mide como la proporción de paquetes perdidos respecto a la cantidad total de paquetes generados por la aplicación. Para ésta medida, se cuentan los paquetes perdidos por congestión a nivel de nodo (desbordamiento del Búfer) y a nivel de la red (colisión de datos):

$$PER = \left(\frac{Paquetes\,Perdidos}{Paquetes\,Enviados}\right) * 100 = (\%) \tag{2.3}$$

El caso de estudio es conocer el comportamiento de las métricas QoS cuando la red WNS entra en estado de congestión. Para ello, se iniciará con una red con baja carga de tráficos (1 nodo) y se incrementará gradualmente hasta que se congestione. En particular se pretende observar los efectos del mecanismo de acceso al medio de la

subcapa MAC, los problemas del canal inalámbrico y la operación del protocolo de transporte TCP, en presencia de congestión.

A continuación se muestran los resultados para cada métrica de estudio. Cada resultado obtenido es el promedio de 200 eventos (al menos) y repetido el experimento 10 veces. Primero, con la finalidad de clarificar el efecto de la congestión en la WSN, presentamos la figura 2.9. En ésta se presenta la cantidad de paquetes que arriban a un nodo, la cantidad de paquetes que el nodo transmite al canal inalámbrico y la carga de tráfico de toda la red. Observe que cuando hay baja carga en la red (menor a 5 nodos), el 100% de los paquetes que recibe el nodo son transmitidos. Sin embargo, conforme la carga de tráfico en la red aumenta, también lo hace el tiempo que el canal se mantiene ocupado; lo cual reduce la probabilidad de transmisión y por ende la cantidad de paquetes enviados. Cuando la red alcanza los 6 nodos, se reduce un 20% la cantidad de paquetes que el nodo puede transmitir. Este efecto es similar en el resto de los nodos, ya que utilizan los mismos mecanismos de acceso al canal; afectando el nivel global de QoS de la red.

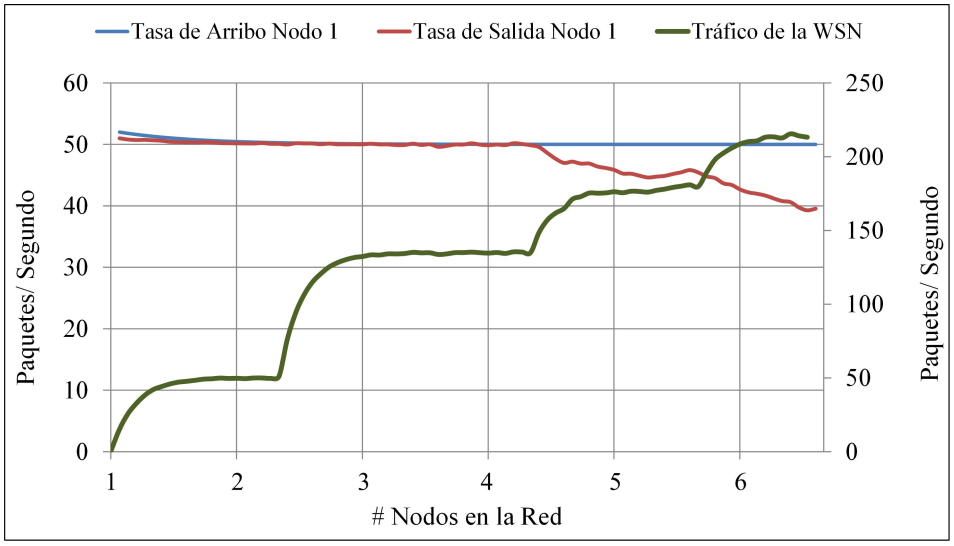

Figura 2.9: Efecto de la Congestión en el Nodo 1.

Al respecto de las métricas QoS, la figura 2.10 presenta el TH alcanzado por la red durante la simulación. Observe que en general conforme la cantidad de nodos en la red se incrementa (o cantidad de tráfico) también lo hace el valor de TH. Sin embargo, el valor que alcanza el TH, después de 2 nodos, no crece significativamente con la

incorporación de más nodos y se mantiene alrededor de los 64 kbps. A diferencia del caso cuando pasa de 1 a 2 nodos, de 36.04 a 63.36 kbps, un incremento de más del 50%. Esto es explicable, ya que cuando existen pocos nodos en la red el nivel de congestión es bajo, al igual que la probabilidad de colisiones y el desbordamiento de búfer. Por ende, una mayor cantidad de paquetes pueden ser enviados al nodo destino. Sin embargo, cuando la cantidad de nodos se incrementa, el nivel de contienda por el acceso al canal también lo hace. Provocando una mayor congestión, lo cual se traduce en un mayor número de paquetes perdidos; afectando los niveles de TH que la red puede alcanzar.

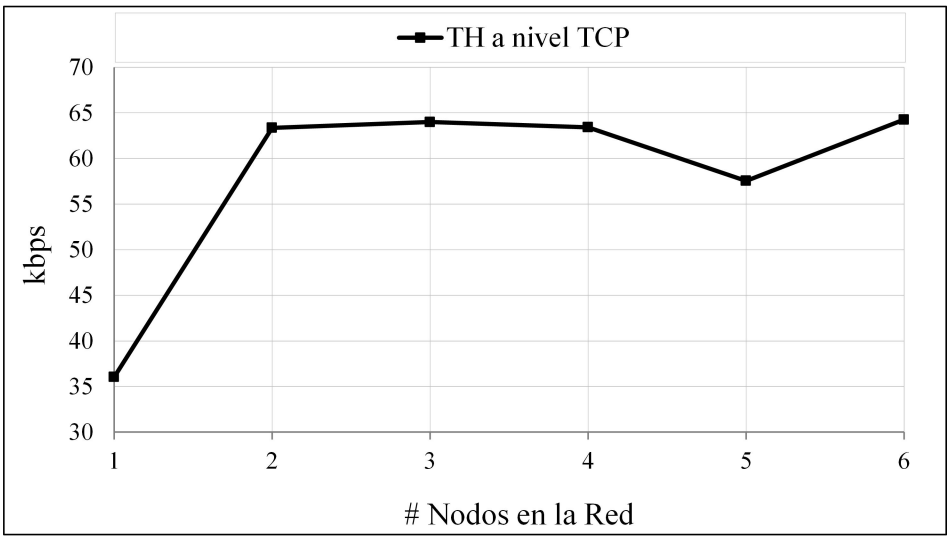

Figura 2.10: Efecto de la Congestión el Throughput.

La figura 2.11 muestra el comportamiento de la WSN para las métricas RTP (tanto a nivel de TCP como a nivel de la subcapa MAC) y PER. Al igual que en el caso del TH, los mejores resultados para RTP y PER se obtienen cuando la red tiene la menor cantidad de nodos. Por ejemplo, cuando la red tiene un sólo nodo, el valor de RTP a nivel TCP es de 5.8 ms y el valor de PER es de 0.25%. Sin embargo, cuando la red tiene carga máxima (6 nodos), el valor de RTP a nivel TCP es de 45.77 ms y un PER de 26.06%. De igual forma es consecuencia del incremento de la congestión en la red. Una conclusión importante, a partir de la figura 2.11, es que el valor del retardo TCP es mayor que el de la subcapa MAC, sobre todo en alta congestión. Primero, es claro que debe ser mayor por la forma en que se mide (extremo a extremo entre capas

TCP), ya que el retardo TCP incluye el tiempo de MAC más el tiempo incurrido por el propio protocolo TCP. Sin embargo, TCP aporta la mayor parte del valor final del retardo. Por ejemplo, para el caso de 3 nodos de la figura 2.11, el retardo de TCP fue de 22.48 ms y para MAC fue de 6.1 ms; que representa un incremento de 72.84%. Esto es debido, a los mecanismos de usados por TCP para el intercambio de datos entre las capas de Transportes (emisor-receptor).

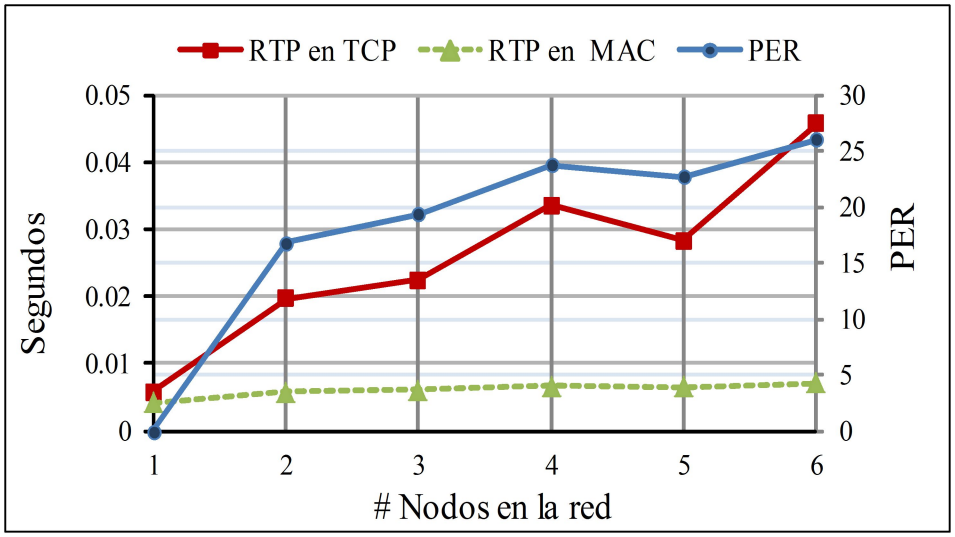

Figura 2.11: Efecto de la Congestión en el Retardo y Paquetes Perdidos.

En conclusión, los resultados muestran que el desempeño alcanzado en las métricas QoS por las WSNs dependen en gran medida del grado de congestión de la red, y este grado depende del *nivel de contienda* por el acceso al canal. Mayor tráfico en la red provoca mayor nivel de contienda y los peores niveles QoS en las 3 métricas. Esto debido a las características de operación del protocolo de acceso al medio de la subcapa MAC y del protocolo de comunicaciones TCP. Lo cual, hace que éstos dos entidades, capa de Transporte y subcapa MAC, sean fundamentales para lograr una comunicación eficiente de los datos generados por las aplicaciones. Información más detallada de éste evaluación se presenta en nuestro trabajo presentado en [Buenrostro et al., 2013].

2.3 Control de Congestión

La congestión en las redes WSNs es un reto obligado de resolver, para cumplir con el objetivo primario de la recolección remota de datos. Ya que ésta impacta negativamente los niveles de QoS ofrecidos a las aplicaciones, como se vio en la sección 2.2. Por lo tanto, es necesario incorporar mecanismos que controlen sus efectos o impidan que se presente; si queremos habilitar WSNs para aplicaciones médicas.
En la evaluación del impacto de la congestión (Sec.2.2.2) se determinó que los protocolos de comunicación de la capa de Transporte y la subcapa MAC juegan un rol importante en la transmisión de dato extremo a extremo, y por ende en el QoS que las WSNs pueden ofrecer. Por lo tanto, es importante la selección de estos protocolos con el fin de alcanzar los mejores niveles de desempeño de la red. A continuación se presenta los conceptos fundamentales de la capa de Transporte y la subcapa MAC.

2.3.1 Subcapa MAC

La subcapa MAC es la parte inferior de la capa de Enlace de Datos o capa dos del modelo de referencia OSI [Sec.2.4 y Fig.2.13). La subcapa MAC se utiliza para regular el uso del medio de comunicaciones. El control de acceso al medio se puede implementar tanto de forma centralizada como distribuida. El esquema centralizado es utilizado en redes con accesos controlados por un sólo nodo en toda la red. En el esquema distribuido, cada nodo de la red cuenta con los mecanismos para usar el canal de forma independiente. El esquema centralizado se distingue por el orden, para evitar las colisiones; mientras que el esquema descentralizado por la independencia y la reducción del tiempo de acceso. Esta subcapa es especialmente importante en las redes que utilizan un sólo canal común de comunicaciones de acceso múltiple; ya que un mal control puede dar lugar a transmisiones simultaneas que provoquen colisión de datos en la red. Entonces, el punto clave consiste en determinar quien tiene el derecho a utilizar el canal y por cuanto tiempo, cuando existe una competición por éste. Existen diversos algoritmos para compartir un canal de acceso múltiple, y de acuerdo su diseño puede ser más o menos efectivo en su operación para asignar el canal. Por ejemplo, en las redes bajo el estándar IEEE 802.15.4 la operación recae en el algoritmo de Acceso Múltiple por

Detección de Portadora con Prevención de Colisiones (CSMA-CA, Carrier Sense Multiple Access with Collision Avoidance). El estándar IEEE 802.15.4 y CSMA-CA son detallados en la sección 2.5, puesto que son la base de nuestra propuesta.
En lo que respecta a la función del control de congestión, la subcapa MAC tiene algunos puntos que demuestran su importancia. Aunque el esfuerzo colectivo de todos los protocolos de comunicación, de la arquitectura de capas 2.8, es esencial para la provisión de una transmisión eficiente de datos, la subcapa Medium Access Control posee una particular importancia entre ellos; ya que rige el control del acceso al medio de comunicaciones, y todos los protocolos de capas superiores necesitan de ello [Yigitel et al., 2011]. La capa de Transporte (o cualquier otra) no puede proveer una transmisión eficiente sin el supuesto de que un protocolo MAC resuelve los problemas de un canal compartido. Además, la subcapa MAC se ocupa de los retos adicionales de las WSNs, como las limitaciones de energía y del canal inalámbrico (sujeto a diferentes factores incontrolables que afectan la calidad, sección 2.2.1). Contribuyendo con métodos de verificación de errores de bit en el paquete, métodos de retransmisiones para recuperar paquetes perdidos debido a estos errores de bit y con métodos de control de potencia de transmisión, para reducir o aumentar la fuerza de la señal.
En suma, la subcapa MAC es indispensable por que, sin ella no hay acceso al medio para enviar o recibir datos; y por que juega un papel clave para la provisión del control de congestión y del QoS en las redes WSN.

2.3.2 Capa de Transporte

La capa de Transporte es una pieza central de una arquitectura de red por capas (capa 4 de la Fig.2.8). Su objetivo fundamental es proveer servicios de comunicación confiable entre dos procesos de aplicación que se ejecutan en diferentes nodos [Kurose and Ross, 2013]. Además, la capa de transporte es una verdadera capa de extremo a extremo, ya que permite extender los servicios de comunicación de las aplicaciones a través de los diferentes nodos de la red. A diferencia de la capa de RED y la subcapa MAC, que están encadenadas entre un nodo y su vecino inmediato, y no extremo a extremo [Halsall, 2006].
Hablando de forma general, la capa de Transporte puede ofrecer dos tipos de servicios:

una transferencia confiable de datos y el control de congestión. Transporte debe enfrentar uno de los problemas más fundamentales en las redes de comunicaciones: cómo 2 entidades pueden comunicarse confiablemente sobre un medio que puede perder y corromper datos. Para lo cual, propone una serie de técnicas de seguimiento, notificación, ordenamiento y recuperación de datos [Kurose and Ross, 2013]. Otro problema fundamental que debe enfrentar la capa de Transporte es la congestión, cuando la velocidad de transmisión de la capa de transporte es mayor a la que puede recibir la capa de transporte receptora. Para ello, los protocolos de esta capa deben incluir mecanismos para regular el flujo de información entre dos entidades de transporte; con el fin de evitar, o recuperarse de, la congestión de la red. Además, en el rubro de la QoS, los protocolos de transporte pueden ofrecer un trato especial a los datos de la aplicación, ya sea para programarlos y/o priorizarlos de acuerdo a su importancia [Rahman et al., 2008b].

Protocolo TCP y UDP como ejemplos de Protocolos de Transporte

El Protocolo de Control de Transmisión (TCP) [Institute, 1981] y el Protocolo de Datagrama de Usuario (UDP, User Datagram Protocol) [Institute, 1980], son dos protocolos de la capa de transporte ampliamente usados en las redes de Internet.

TCP es un protocolo orientado a conexión que ofrece servicios de comunicación extremo a extremo entre dos nodos de la red. TCP incluye servicios de confiabilidad en la entrega de datos y control de congestión, con el objetivo de proveer una transmisión confiable y ordenada de los datos. En TCP cuando dos procesos desean comunicarse, necesitan forzosamente establecer una conexión entre ellos antes de iniciar el envío de los datos; para ello utiliza un mecanismo de sincronía de tres-pasos: solicitud de conexión, respuesta a la solicitud y un mensaje de confirmación por parte del nodo solicitante. La entrega confiable se sustenta en que TCP necesita recibir un mensaje de acuse de recibo, por cada paquete de datos que se envía; llamado mensaje de Reconocimiento (Ack, Acknowledgement). Si el mensaje no se recibe en un determinado tiempo, el nodo emisor debe retransmitir dicho paquete.

Para efectos de controlar la congestión, TCP plantea como base regular el tráfico generado en ambos extremos de la conexión. Para la detección de la congestión en TCP, cada

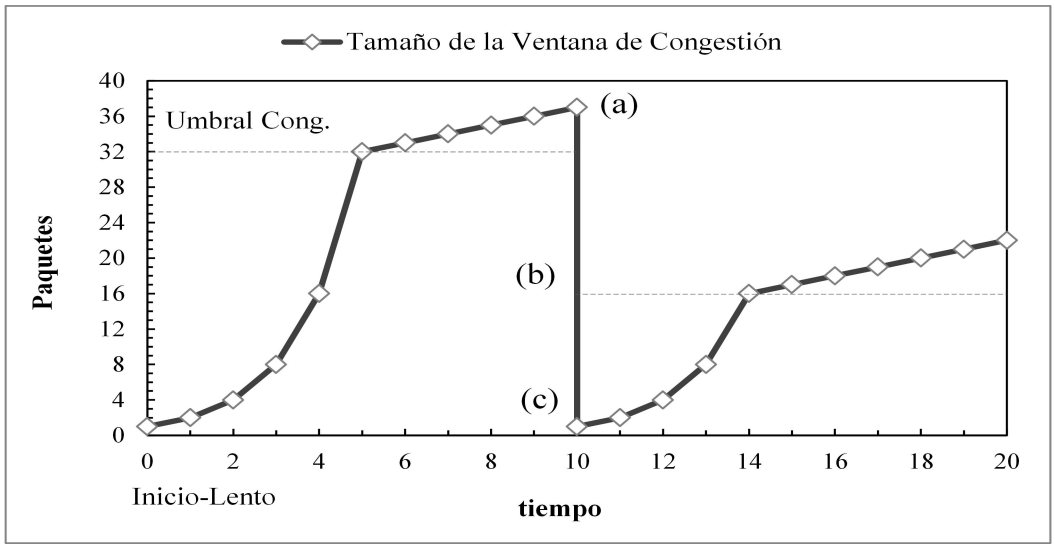

Figura 2.12: Algoritmos de Inicio-Lento y Prevención de Congestión.

nodo utiliza como "indicador" del estado de la red la pérdida de uno o más paquetes, y/o la recepción de Acks duplicados; e infiere a partir de ello, la congestión en la red. El control de la congestión la realiza en base dos algoritmos [Yang et al., 2013]: una estrategia de Inicio lento (Slow-Start) y uno para prevenir la congestión. Estos algoritmos buscan el equilibro de la conexión: tasa a la cual debería inyectarse los paquetes para utilizar al máximo la conexión, pero sin provocar la saturación de la misma. La figura 2.12) muestra la operación de estos dos algoritmos de control de congestión. Al inicio, Slow-Start permite incrementar *exponencialmente* la cantidad de datos que un nodo puede transmitir (iniciando con un segmento de datos), en tanto éste reciba la confirmación de los paquetes, hasta que se alcanza un umbral de tamaño predefinido (ventana de congestión). Cuando esto sucede, se activa el algoritmo de prevención de la congestión, lo cual provoca un incremento lineal de la cantidad de datos que se pueden enviar. Si ocurre una pérdida de paquete (punto "*a*" en la Fig.2.12), TCP lo interpreta como un síntoma de congestión y lleva a cabo un *estrechamiento exponencial de la ventana*, que consiste en reducir a la mitad el tamaño del umbral de la ventana de congestión y reinicia el proceso de envío de datos de Slow-Start a un segmento (puntos "*b*" y "c" en la Fig.2.12). Con esta estrategia TCP busca utilizar al máximo el canal de comunicaciones y anticiparse a la congestión de la red.

Por su parte UDP es un protocolo que utiliza un mínimo de mecanismos para proveer la comunicación de paquetes entre dos procesos [Institute, 1980]. Por lo tanto, UDP es considerado como un protocolo de operación simple y de bajos requerimientos. Además, se clasifica como un protocolo que ofrece bajos retardos en la comunicación de datos; ya que no requiere confirmar la entrega de paquetes ni establecer una sesión de datos. UDP es un protocolo "no regulado", por lo cual, una aplicación que utiliza UDP puede enviar paquetes a cualquier velocidad que le plazca, durante el tiempo que quiera.
Es importante mencionar que el concepto básico de operación de TCP y UDP ha dado forma al diseño de nuevos protocolos de capa de transporte para redes inalámbricas.

Como aquí se ha explicado la capa de Transporte y la subcapa MAC son dos entidades claves para el control de la congestión en las WSNs; pero aún más importante el hecho de que éstas puedan trabajar en conjunto para mejorar sus resultados y optimizar los recursos de la red. Dentro de los protocolos de comunicación de las WSNs. Esto nos lleva a la estrategia de diseño interacción de capas (Cross-Layer), la cual se explica en la sección 2.4.

2.4 Estrategía de Diseño Cross-Layer para las WSNs

Tradicionalmente se ha utilizado el modelo de Interconexión de Sistemas Abiertos (OSI, Open Systems Interconnection de siete capas [ISO/IEC, 1994]) en el desarrollo de la pila de protocolos, para la inter-conexión de redes (2.13) [Zimmermann, 1980]. Principalmente por las ventajas anunciadas por este modelo: (i) un sistema modular, donde cada módulo tiene claramente definido sus funciones y procedimientos para habilitar la independencia de capa. (ii) el diseño de capas reduce la complejidad de diseño, que permite una fácil implementación y mantenimiento. (iii) la estandarización, asegura la inter-operabilidad entre los diferentes sistemas integrados a la red, si fueron diseñados en base al modelo. Las primeras cuatro capas de éste modelo se conocen como la Pila de Protocolos de Comunicaciones; ya que estas capas se encargan de la comunicación por el medio físico y la interacción entre los mecanismos de transmisión extremo a extremo [Vuran and Akyildiz, 2010].

Figura 2.13: Modelo de Referencia de capas OSI

Aunque el estricto enfoque por capas sirve como una solución elegante para la interconexión de redes cableadas y estáticas (nodos fijos a la red), se argumenta que el diseño de protocolos de comunicaciones basado en éste enfoque de capas no es adecuada para la funcionalidad eficiente de las redes inalámbricas [Fu et al., 2014], [Edirisinghe and Zaslavsky, 2013], [Vuran and Akyildiz, 2010], [Wang and Abu-Rgheff, 2003], [Stine, 2006]. Hay varias razones para éste argumento. En primer lugar, el modelo de capas OSI se diseño para las redes cableadas y estáticas, que son muy diferentes a la naturaleza de las redes inalámbricas. Las redes inalámbricas son altamente dinámicas y tienen recursos de red limitados. La naturaleza dinámica se refiere a la movilidad de usuarios, a la conexión y re-conexión de nodos, lo cual hace una red cambiante y compleja. Los dispositivos de la red tienen recursos limitados de energía, de capacidad de procesamiento y de almacenamiento (Sec.2.1). Lo cual obliga a mejorar el desempeño de los protocolos de comunicaciones y su capacidad de adaptación al entorno y los requerimientos. El diseño tradicional de protocolos en capas, no es capaz de atender estas necesidades de adaptaciones, aplicaciones más exigentes y las condiciones de redes complejas. En segundo lugar, el concepto estricto de capas impide las mejoras en el rendimiento del sistema inalámbrico a través de interacciones conjuntas de varias capas, limitando la

posibilidad de compartir información valiosa que pueda mejorar la consciencia de las necesidades de operación de cada protocolo. En tercer lugar, los estándares estrictos pueden inhibir la innovación. Se argumenta, por lo autores aqui citados, que las normas deben mantener al mínimo las regulaciones, para que ésta permita a los investigadores futuras innovaciones.

2.4.1 Diseño de Interacción de capas o Diseño Cross-Layer

En la literatura revisada clasifican dos tipos de diseño para hacer frente a las limitaciones del diseño basado en capas para las redes inalámbricas: Los diseños incluyen *arquitecturas que no utilizan capas* y los diseños que usan la *Interacción entre capas.*
Arquitecturas de red que no utilizan el modelo de capas o proponen integrar varias capas en una sola, proponen una alternativa de diseño a la arquitectura tradicional de protocolo de comunicaciones en capas; mediante la introducción de nuevos conceptos de abstracción [Vuran and Akyildiz, 2010], [Braden et al., 2003]. Por ejemplo, Braden propone una arquitectura basada totalmente en una nueva abstracción funcional. Su arquitectura organiza la comunicación utilizando unidades funcionales llamadas "Roles", en lugar de dividir las funciones en capas. Por su parte Vuran propone algo similar, en su propuesta de un Protocolo de Comunicaciones Cross-Layer (XLP), XLP integra o "funde" las funciones principales de la pila de protocolos de comunicaciones (capa Física hasta la capa de Transporte) en una sola capa. El enfoque de XLP son las funciones de acceso al medio, ruteo de la información y control de congestión. Sin embargo, es discutible hasta qué punto estas propuestas podrían reemplazar el modelo de red de capas, que se ha desplegado ampliamente como una arquitectura de red de gran alcance. Además, propiamente hablando, XLP no es un protocolo Cross-Layer, ya que éste propone una sola capa integrada con todas las funciones; lo cual trae consigo los problemas de diseño, mantenimiento y división de tareas.
Puesto que una arquitectura sin capas es considerada como sustituto del modelo de capas, ya que presenta abstracciones totalmente nuevas, ésta no se ajusta al concepto de estratificación en absoluto. Estas soluciones requieren reemplazar y re-diseñar por completo la pila de protocolos existentes, lo cual es un trabajo complejo. Una propuesta alternativa es la Interacción Cross-Layer (o interacción entre capas), la cual puede

superar las limitaciones e ineficiencias de la pila de protocolos en capas para redes inalámbricas [Haas, 2001]. La interacción Cross-Layer permite el intercambio de información de las diferentes capas con el fin de optimizar los servicios de comunicaciones que ofrece la pila de protocolos (Fig.2.14), por ejemplo, niveles de seguridad, mejorar el QoS, ahorro de energía y movilidad. A través de esta interacción, se puede utilizar y/o modificar los estados de parámetros de dos o más capas para alcanzar los objetivos de optimización planteados [Mendes and Rodrigues, 2011], [Fu et al., 2014].
Desde el punto de vista de la implementación, adaptación y aceptación por parte de la comunidad; las soluciones que pueden coexistir con la actual pila de protocolos en capas (como la interacción Cross-Layer) son significativamente bien consideradas [Edirisinghe and Zaslavsky, 2013]. Por lo anterior, el diseño Cross-Layer es considerado como una solución prometedora, a diferencia de las propuestas de arquitecturas sin capas.

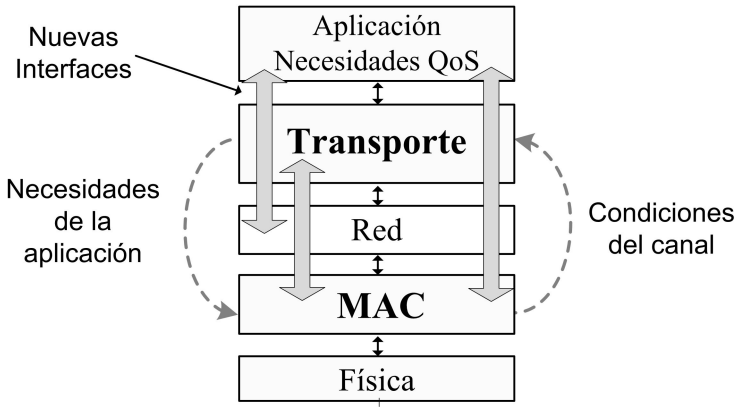

Figura 2.14: Ejemplo de interacción Cross-Layer.

2.4.2 Retos en el diseño Cross-Layer

Un protocolo bajo diseño Cross-Layer es definido como un "Protocolo que rompe premisa de independencias del modelo tradicional de capas". A partir de ésto, se entiende que los protocolos de diferentes capas no adyacentes pueden tener una comunicación directa y/o compartir parámetros entre ellos [Srivastava and Motani, 2005]. El autor extiende la definición aún más con el hecho de que la violación de la arqui-

tectura por capas, obliga a la creación de nuevas interfaces entre capas (Fig.2.14), la re-definición de los límites de capa, el diseño de protocolos de una capa basada en los detalles de otra capa, a la sintonización conjunto de parámetros a través de las capas y renunciar al diseño de los protocolos de forma independiente.
Aun que la interacción Cross-Layer se concibe como una alternativa viable para diseñar una arquitectura de protocolos para redes inalámbricas, como las WSNs; los beneficios se logran a un costo. A continuación se presentan algunos retos potenciales que debe considerar un diseño Cross-Layer:

- *Dificultad de implementación*, el diseño de soluciones Cross-Layer deben ser cuidadosas y mantener al mínimo los cambios de los protocolos existentes de cada capa. Ya que cualquier modificación puede ser compleja por la necesidad de hacerlo de forma coordinada con los protocolos de cada capa. Por lo tanto el diseño debe asegurar un fácil mantenimiento y actualización.

- *Interoperabilidad*, el diseño Cros-Layer debe preservar la modularidad de la arquitectura de capas; con el fin de asegurar una operación con otros protocolos del modelo de capas.

- *Consecuencias no deseadas*, la interacción Cross-Layer puede causar efectos no deseados en el rendimiento final de toda la red, debido a la afectación en la operación de otras partes del sistema debido a los cambios realizados para instrumentar la interacción Cross-Layer.

- Además, se discute en algunos trabajos que las interacciones Cross-Layer pueden causar *bucles de control*, entre las partes de los diferentes protocolos que interactúan entre sí [Kawadia and Kumar, 2005].

- El *número de capas* que debe integrarse en la interacción de la solución Cross-Layer debe ser el mínimo necesario, para lograr los objetivos de optimización. Además, que capas de la pila protocolos son las que se deben integrar. Así mismo, debe quedar claro que capa debe interactuar con cual otra, par evitar interacciones inútiles que hagan más compleja la solución.

Por lo tanto, para conseguir las ventajas de las interacciones Cross-Layer, la solución debe ser diseñada e implementada cuidadosamente.
A pesar de éstos retos, sostenemos que la interacción Cross-Layer es una solución viable para hacer frente a los problemas de la pila de protocolos en capas para redes inalámbricas. Además, las WSNs para aplicaciones médicas necesitan de la optimización de los recursos para poder satisfacer las demandas de QoS de éstas aplicaciones (ver secc. 2.1.2). Si bien, diseñar por capas reduce la complejidad de diseño y optimizar el protocolo de forma particular. Sin embargo, es mejor que todas las capas trabajen en sintonía con los objetivos del servicio para lograr una mejor optimización y más en las WSNs que son de recursos limitados.
Una importante conclusión a partir de ésta sección es evitar el uso de múltiples capas, el uso de capas sin sentido a los objetivos de optimización y seguir manteniendo el diseño general en capas. Además, como se evidenció en la sección 2.2.2 la capa de Transporte y la subcapa MAC estan relacionadas con el control de la congestión y los niveles de QoS que una WSN puede ofrecer. Por lo tanto, primero, proponemos el uso de la interacción Cross-layer que puede operar dentro de la pila de protocolos existentes; esto es, que permite el uso de protocolos por capas (como el IEEE 802.15.4 y TCP) dentro del diseño. Y segundo, proponemos el uso de la capa de Transporte y la subcapa MAC para la solución de los objetivos de optimización (control de congestión y los niveles de QoS).

2.5 Descripción Técnica de IEEE 802.15.4

IEEE 802.15.4 es uno los estándares más utilizada para desarrollar tecnologías para las WSNs. El estándar pertenece al grupo de las Redes Inalámbricas de Área Personal (WPAN, Wireless Personal Area Network) [Gutiérrez et al., 2011] y fue diseñado para operar dentro del Espacio de Operación Personal (POS, Personal Operating Space); extendiéndose hasta un máximo de 10 metros a la redonda. El estándar IEEE 802.15.4 es ideal para construir dispositivos que necesiten una transmisión de datos de baja velocidad (250 kilobits por segundo o kbps a 2.4 GHertz o GHz), muy bajo consumo de energía y bajo costo de implementación, como se muestra en la figura 2.15. Que

además requieran un diseño de baja complejidad y sean dispositivos de fácil portabilidad [LAN/MAN Standard, 2006]. Por estas características el estándar IEEE 802.15.4 se ha propuesto para aplicaciones en el entorno de las personas como la supervisión de la salud. Donde los dispositivos dentro de la WSN pueden recuperar signos vitales de las personas y transmitirlos hasta un punto remoto para su análisis (como se menciono en el uso de las WSN).

El estándar IEEE 802.15.4 no fue diseñado para competir con otras tecnologías inalámbricas, si no que complementa la gama de tecnologías inalámbricas disponibles [Gutiérrez et al., 2011]. La figura 2.15 ilustra el espacio de operación de algunos estándares IEEE-802 clasificados en: WLAN, (Wireless Local Area Network) y WMAN (Wireless Metropolitan Area Network). Note que el estándar IEEE 802.15.4 esta diseñado para no traslaparse con otras tecnologías inalámbricas de alta velocidad y cobertura como las WLAN y/o las WMAN.

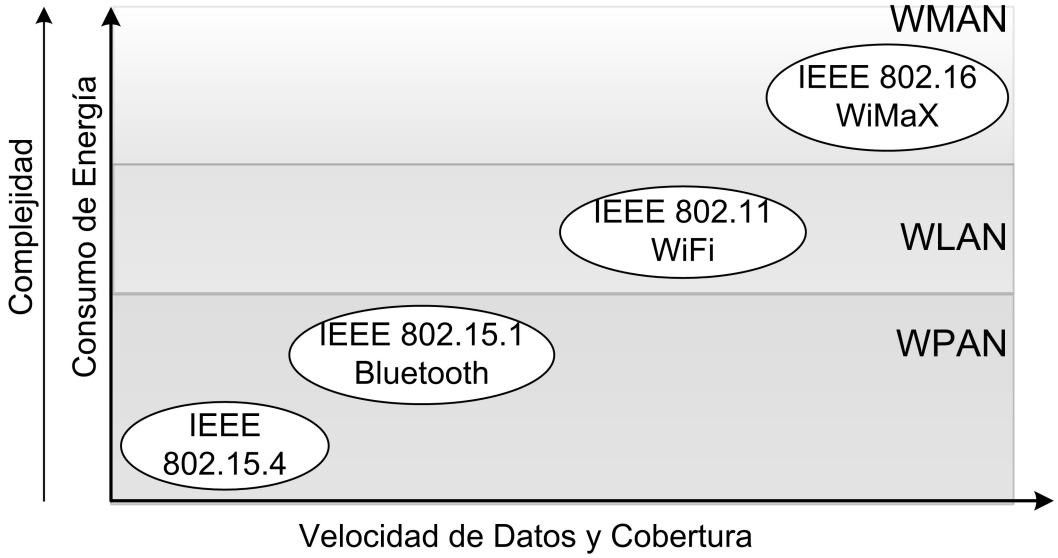

Figura 2.15: Tecnologías de Comunicación inalámbricas

2.5.1 Componentes y Topología de Red de IEEE 802.15.4

El estándar define tres diferentes componentes o dispositivos inalámbricos para conformar una WSN: Coordinador de la PAN, Coordinador y dispositivo Cliente. Cada

WSN contiene un sólo Coordinador de la PAN, que es el único dispositivo que puede establecer una nueva red y definir el modo de operación de la misma. La WSN puede tener varios Coordinadores y dispositivos Cliente interconectados entre sí. Los dispositivos Coordinadores proveen servicios de control a otros dispositivos de la red y actúan como un puente entre el Coordinador de la PAN y los dispositivos Cliente, cuando éstos últimos están fuera de su rango de cobertura. Los dispositivos Cliente están propuestos principalmente para las funciones de detección de variables físicas y para el envío y recepción de datos [Gutiérrez et al., 2011].

Una red IEEE 802.15.4 puede operar en una de las dos siguientes topologías base: Tipo Estrella y Par a Par. Ambas se muestran en la figura 2.16a y 2.16b. La topología Estrella se forma con los dispositivos Cliente conectados a un único dispositivo central, que actúa como Coordinador de la PAN. El dispositivo central es el único que puede conectarse con más de un dispositivo. Por su parte, los dispositivos Cliente sólo pueden conectarse al dispositivo central. La topología Estrella esta restringida a conexiones directas (o de un salto) entre los dispositivos Cliente y el Coordinador de la PAN. Esto limita su cobertura de red; sin embargo, su topología es muy simple y tiene un bajo retardo de transmisión de datos.

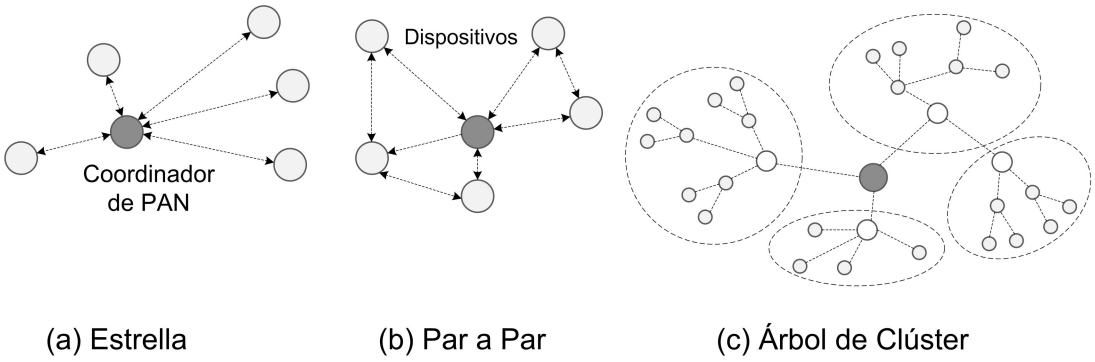

Figura 2.16: Topologías de red de las WSNs.

La topología Par a Par también tiene un Coordinador de la PAN; sin embargo, se diferencia de la topología Estrella en que cualquier dispositivo puede comunicarse con cualquier otro de forma directa, siempre y cuando estén en el mismo rango de cobertura. Esta característica permite crear redes más complejas y de mayor cobertura. Una de

éstas redes, que puede ser soportadas por el IEEE 802.15.4, es la red Árbol de Clúster. Una red Árbol de Clúster (Fig.2.16c), puede ser vista como un *árbol jerárquico de dispositivos en clúster*. El despliegue de los dispositivos en este tipo de redes proveen una completa conectividad, un enrutamiento simplificado y una cantidad menor de enlaces directos; comparado con la topología Par a Par [Gutiérrez et al., 2011]. Las hojas del árbol son ocupadas por los dispositivos Cliente. Sin embargo, todos los demás dispositivos del árbol deben ser Coordinadores; ya que necesitan retransmitir mensajes a otros dispositivos y conectarse a más de un dispositivo de la red. La red Árbol de Clúster debe contar con un sólo Coordinador de PAN. Y cada clúster del árbol debe tener un Coordinador para administras sus nodos en su entorno, llamado Cabeza del Clúster. Los dispositivos de cada clúster son independientes entre sí. Además, si los nodos Cliente quieren enviar datos fuera de su clúster, lo hacen a través del dispositivo Cabeza de Clúster.

Esta variante, de la topología Par a Par, muestra la flexibilidad del estándar para soportar configuraciones diferentes a las topologías base (Estrella y Par a Par); con la intención de mejorar la cobertura y la operación de la red. Sin embargo, requieren del apoyo de las capas superiores para su construcción y operación. En nuestro trabajo de investigación se considera esta flexibilidad para diseñar nuestra topología de red; la cual se basa en un árbol jerárquico de clústeres (con dispositivos en configuración de Estrella). Dicha propuesta se detalla en la sección 4.2.1.

2.5.2 Arquitectura de IEEE 802.15.4

La arquitectura del IEEE 802.15.4 es definida en términos de un número de "capas o bloques" con el fin de simplificar el estándar. Cada capa es responsable de definir una parte del estándar y de ofrecer servicios a las capas superiores. El diseño de estas capas es basado en el modelo de referencia OSI. El estándar IEEE 802.15.4 define las especificaciones de las capas inferiores del modelo OSI [LAN/MAN Standard, 2006]: la capa Física (PHY, Physical) y la subcapa de Control de Acceso al Medio (MAC, Medium Access Control). La figura 2.17 muestra estas capas y su relación con el modelo de referencia OSI.

El estándar IEEE 802.15.4 ofrece dos tipos de servicios a través de los Puntos de Acceso

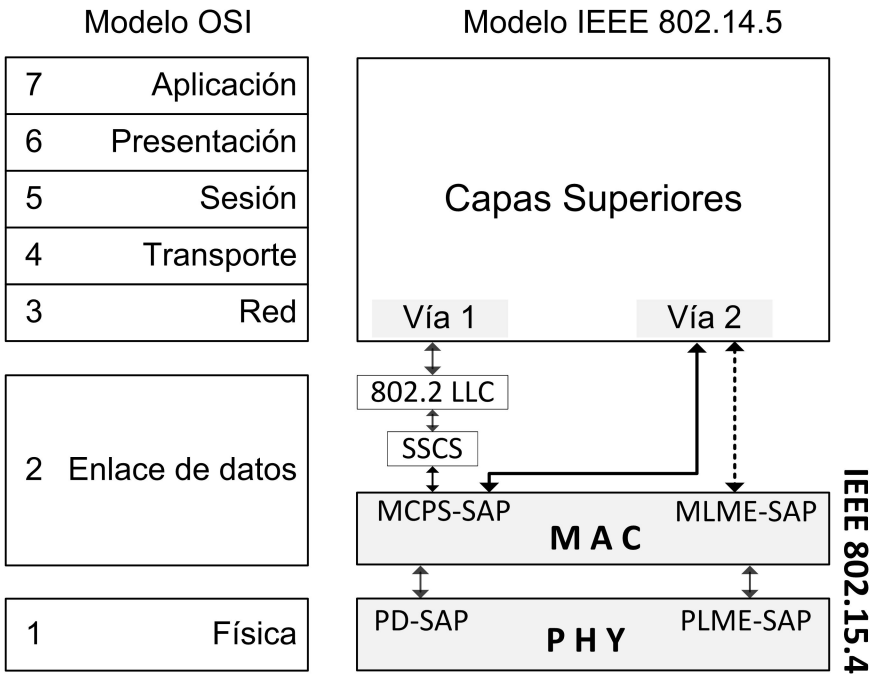

Figura 2.17: Modelo de Referencia OSI y Arquitectura IEEE 802.15.4

al Servicio (SAP, Service Access Point), como lo muestra su arquitectura de la figura 2.17:

- **Servicio de Datos**: Para el intercambio de unidades de datos entre las capas del IEEE 802.15.4 y las capas superiores. El SAP correspondiente a la capa PHY es el servicio de *Datos de Capa Física* (PD-SAP, PHY Data-SAP) y para MAC es la *Parte Común de la Subcapa MAC* (MCPS-SAP, MAC Common Part sublayer-SAP).

- **Servicios de Administración**: A través del cual se pueden administrar las funciones de cada capa y gestionar sus bases de datos de objetos o parámetros de operación. El SAP de la capa PHY es la *Entidad de Gestión de la Capa Física* (PLME-SAP, PHY Layer Management Entity-SAP) y para MAC es la *Entidad de Gestión de la Subcapa MAC* (MLME-SAP, MAC sublayer Management Entity-SAP).

Estos servicios pueden ser accedidos por las capas superiores por dos vías: *1)* usando una subcapa de Control de Enlace Lógico (LLC, Logical Link Control) tipo IEEE 802.2 [ANSI/IEEE, 1984]. En éste caso, el IEEE 802.15.4 define la Subcapa de Servicios Específicos de Convergencia (SSCS, Service-Specific convergence sublayer) como una interfaz entre ellas (vía 1 de la Fig.2.17). *2)* de forma directa, mediante el diseño propietario de las interfaces de enlace lógico entre las capas superiores y el IEEE 802.15.4; a través de los puntos de acceso MCPS-SAP y MLME-SAP (vía 2 de la Fig.2.17.
La selección de la vía de acceso en el desarrolla de la arquitectura de capas de una WSN, depende de las necesidades y restricciones de los protocolos de comunicaciones que se utilicen.

Capa Física IEEE 802.15.4

La base de la capa PHY es el transceptor de radio frecuencia usado para la transmisión y recepción de datos por el medio físico inalámbrico. Otras tareas son [LAN/MAN Standard, 2006]:

- **Activación y desactivación del transceptor de radio**: Cambia los estados del radio transceptor a los modos de transmisión, recepción o apagado, a solicitud de la subcapa MAC del dispositivo.
- **Detección de Energía** (ED, Energy Detection): Esta tarea realiza la estimación de la potencia presente en el canal de comunicaciones inalámbrico.
- **Indicación de la Calidad del Enlace** (LQI, Link Quality Indication): Esta tarea mide la fuerza de la señal presente en cada paquete de datos recibido, por la cual el dispositivo puede estimar tanto la calidad del paquete recibido como la calidad del enlace inalámbrico.
- **Evaluación de Canal Libre** (CCA, Clear Channel Assessment): La capa PHY realiza la tarea CCA usando ED, si el resultado alcanza un nivel de potencia predefinido se considera que el canal esta ocupado. El resultado de esta tarea es utilizado por el algoritmo de acceso al medio de la subcapa MAC.

Tabla 2.2: Bandas de Frecuencias y Tasa de datos del IEEE 802.15.4.

Banda de Frecuencia (MHz)	Canales	Tasa de Bit (kbps)	Tasa de Símbolos (ksímbolosps)	Región Geográfica
868 a 868.6	1	20	20	Europa
902 a 928	10	40	40	N. América
2400 a 2483	16	250	62.5	Mundial

- **Selección de la frecuencia del canal**: IEEE 802.15.4 esta diseñado para operar en uno de los diferentes canales inalámbricos disponibles, de acuerdo a la banda de frecuencia, a petición de la subcapa MAC.

La capa PHY IEEE 802.15.4 se diseño para operar en una de las tres bandas de frecuencia que se muestran en la Tabla 2.2. Cada banda de frecuencia tiene especificado un conjunto de canales y velocidad de datos disponibles. Además, algunas bandas están limitadas a ciertas regiones geográficas del mundo [LAN/MAN Standard, 2006]. Para el presente trabajo seleccionamos la frecuencia de 2.4 GHz para la operación de los dispositivos; debido a que esta frecuencia presenta la mayor tasa de datos y el mayor número de canales. Se hace uso también de la ED para la selección del canal a utilizar en el momento de la conformación de la red.

La subcapa MAC IEEE 802.15.4

La subcapa MAC se ocupa del control de acceso de los dispositivos de la WSN al medio de comunicaciones inalámbrico. Administrando las oportunidades de acceso al canal para todo los tipos de transferencias de datos. Para este fin, IEEE 802.15.4 utiliza el algoritmo de Acceso Múltiple por Detección de Portadora con Prevención de Colisiones (CSMA-CA, Carrier Sense Multiple Access with Collision Avoidance), que es detallado en la sección 2.5.4. La subcapa MAC, además, es responsable de las siguientes tareas[LAN/MAN Standard, 2006]:

- **Sincronización del Envío de Datos**. MAC utiliza dos modos de sincronización: *i)* cuando un dispositivo Coordinador central genera mensajes periódicamente en la red, llamados Balizas o Faros (por Beacons); para sincronizar el envío de datos

entre sus dispositivos. Un dispositivo de la red no puede enviar o recibir sus datos hasta que reciba un mensaje Baliza del Coordinador Central y ejecute el algoritmo CSMA-CA. Esta operación se conoce como *modo Baliza o Beacon* (Fig.2.18). *ii)* los dispositivos Cliente de la red pueden enviar sus datos al Coordinador central en cualquier momento, mediante la ejecución de CSMA-CA (Fig.2.19a); sin necesidad de esperar un mensaje Baliza, por ello es conocido como *modo Sin-Baliza*. Esta forma envío se conoce como transmisión *Directa*. Sin embargo, cuando un dispositivo Cliente necesita recibir datos del Coordinador debe solicitárselos, mediante el envío de un mensaje especial de "Solicitud de Datos". Sí es el caso, el Coordinador envía los datos usando igualmente el algoritmo CSMA-CA (Fig.2.19a). Esta forma envío se conoce como transmisión *Indirecta*.

- **Asociación y Disociación de dispositivos**. MAC puede integrar o eliminar un dispositivo de la red mediante los mecanismos de Asociación y Disociación. Estos mecanismos permiten la configuración y re-configuración automática de los dispositivos de la WSN.

- **Proporcionar un enlace fiable entre dos dispositivos**. MAC emplea dos mecanismos para mejorar la fiabilidad de la conexión: *i)* Reconocimiento y Retransmisión de paquetes, a través del cual un dispositivo puede generar un mensaje de reconocimiento por cada paquete que recibe; para notificarle al dispositivo emisor que su paquete fue realmente recibido. Sin embargo, si el mensaje de reconocimiento no es recibido, el emisor retransmite el paquete nuevamente. *ii)* Verificación de Datos, para revisar la integridad de los datos dentro del paquete recibido. El dispositivo utiliza una Verificación Cíclica de Redundancia (CRC, Cyclic Redundancy Check) de 16 bits, la cual es conocida tanto por el dispositivo emisor y como por el receptor [Gutiérrez et al., 2011].

En nuestro trabajo hemos seleccionando el modo Sin-Baliza como modelo de transferencia de datos; por la simplicidad de su mecanismo de envío de datos (hacia y desde el dispositivo Coordinador) y por que alcanza los retardos de transmisión más bajos con respecto del modo Beacon. Como lo muestran las figuras 2.18 y 2.19. Por último, el mecanismo de Reconocimiento y Retransmisión de paquetes es activado sólo si las

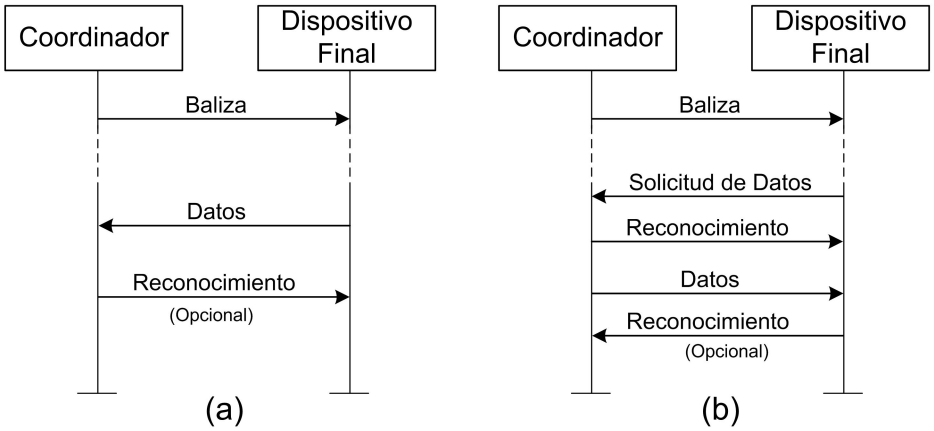

Figura 2.18: Envío de datos en modo Baliza: a) hacia el Coordinador y b) desde el Coordinador.

capas superiores lo requieren. Con la intención de disminuir los paquetes perdidos en la transmisión.

2.5.3 Estructura del Marco de Datos IEEE 802.15.4

La estructura del marco de datos (o FRAME) se ha diseñado para mantener al mínimo su complejidad, mientras que al mismo tiempo alcanza la suficiente robustez para su transmisión en un canal "ruidoso" como el medio inalámbrico [LAN/MAN Standard, 2006]. La figura 2.20 muestra la estructura del marco de datos, éste es originado en las capas superiores. La carga útil de datos se pasa a la subcapa MAC y se conoce como Unidad de Datos de Servicio MAC (MSDU, MAC Service Data Unit). El MSDU es precedido con un Encabezado MAC (MHR, MAC Header) y finalizado con un MAC Footer (MFR). El MHR contiene el control de marco, número de secuencia de datos y el campo de direccionamiento. El MFR está compuesto de una Secuencia de Verificación de Marco (FCS, Frame Check Secuence) de 16 bits. El MHR, MSDU y MFR juntos forman el marco de datos MAC, conocido como Unidad de Datos del Protocolo MAC (MPDU, MAC Protocol Data Unit).

El MPDU se pasa a la capa PHY como carga útil y se conoce como Unidad de Datos de Servicio PHY (PSDU, PHY Service Data Unit). La longitud de la PSDU debe ser menor o igual a 127 octetos [LAN/MAN Standard, 2006] y es precedida por un Encabezado de

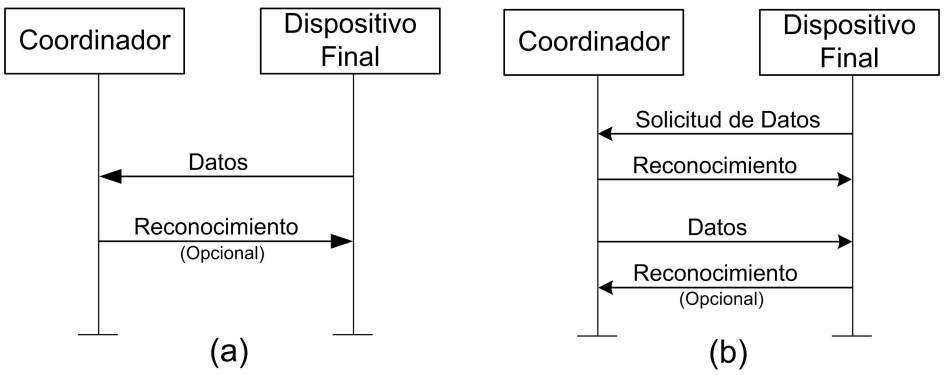

Figura 2.19: Envío de datos en modo Sin-Baliza: a) hacia el Coordinador y b) desde el Coordinador.

Sincronización (SHR, Synchronization Header); que contiene los campos de secuencia de preámbulo y el Delimitador de inicio de Marco (SFD, Start of Frame Delimiter), y por un Encabezado PHY (PHR, PHY Header) que contiene la longitud de la PSDU en octetos. La secuencia de preámbulo y los datos del SFD habilitan al receptor para que este pueda lograr la sincronización de símbolo. El SHR, PHR, y el PSDU juntos forman la Unidad de Datos del Protocolo PHY (PPDU, PHY Protocol Data Unit).

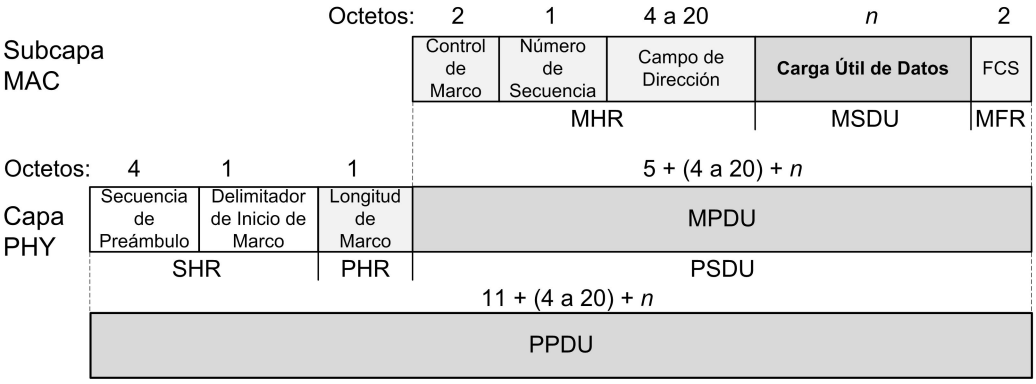

Figura 2.20: Estructura del Marco de Datos IEEE 802.15.4.

2.5.4 Algoritmo de Acceso al medio CSMA-CA

Las WSNs tienen un "canal común" como medio de comunicaciones para todos los dispositivos de la red. Lo cual significa que sólo puede ser usado por un dispositivo a la vez; de lo contrario puede generarse una "colisión de datos" por la transmisión simultanea de múltiples dispositivos. La figura 2.21 ejemplifica éste problema, cuando los dispositivos A y B transmiten hacia C al mismo tiempo. El resultado de una colisión es la perdida de los datos transmitidos, debido a que el receptor no puede recuperar la señal correctamente. Entonces, el punto clave consiste en determinar quien tiene el derecho a utilizar el canal, cuando existe una competición por éste.

IEEE 802.15.4 emplea el algoritmo CSMA-CA como mecanismo para "regular" el uso del medio común por todos los dispositivos de la red. La base de operación de CSMA-CA es "escuchar" la actividad del canal antes de transmitir para reducir la probabilidad de colisiones con otras transmisiones en curso [Buratti et al., 2009]. El IEEE 802.15.4 considera dos variantes de operación para el algoritmo CSMA-CA, dependiendo si la red esta configura en modo Baliza o Sin-Baliza. Pero en ambos casos, el algoritmo CSMA-CA es construido usando *unidades de tiempo* llamados **periodos de Retroceso o de Backoff**, donde un periodo de Backoff debe tener una duración igual a 20 símbolos (320 μseg de acuerdo a la Tabla 2.2) [LAN/MAN Standard, 2006].

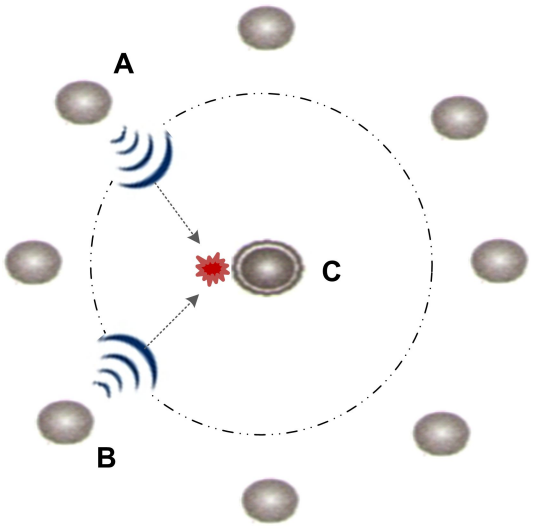

Figura 2.21: Problemas de Colisión de datos en redes de canal común.

A continuación se describe el CSMA-CA para el modo Sin-Baliza, por ser el modo seleccionado en nuestro trabajo. En este modo, cada vez que un dispositivo necesita transmitir un paquete de datos usa el algoritmo CSMA-CA de la figura 2.22 para competir por el acceso al canal. Para la operación de CSMA-CA, un dispositivo debe mantener dos variables en cada intento de transmisión: **Número de Backoffs (NB)** y **Exponente de Backoff (BE)**. NB registra el número de veces que el algoritmo CSMA-CA intenta la transmisión del paquete, para la solicitud en curso. El valor de BE esta relacionado con el número de períodos aleatorios de Backoff que un dispositivo debe esperar antes de evaluar la disponibilidad del canal. NB es usado para controlar el número máximo de veces que un dispositivo puede intentar la transmisión en curso; y el valor de BE es usado para obtener diferentes tiempos de espera, con lo cual se busca evitar los problemas de colisión en la red.

La figura 2.22 ilustra los pasos del algoritmo CSMA-CA. Primero la subcapa MAC debe inicializar los valores de NB=0 y BE=macMinBE (paso 1). Después, la subcapa MAC debe retardar el proceso de transmisión un tiempo aleatorio; esto lo consigue mediante el paso (2) al obtener períodos de Backoffs en el rango de [0 a 2^{BE}-1]. Una vez que expira este tiempo, entonces verificar la disponibilidad del canal mediante la tarea CCA de capa PHY (paso 3). Si el canal esta ocupado (paso 4), la subcapa MAC deberá incrementar en uno el valor de NB y BE; asegurándose que BE no sea mayor a macMaxBE. Si el valor de NB es menor o igual a macMaxCSMABackoff (paso 6), el algoritmo CSMA-CA debe regresar al paso al paso (2) para solicitar otro período de espera antes de intentarlo de nuevo. Este proceso se repite hasta que el valor de NB sea mayor que macMacCSMABackoff. Si esto ocurre, se declara una Falla de Acceso al Canal (CAF, del inglés Channel Access Failure); lo cual implica que el paquete en curso debe ser eliminado del dispositivo.

Si en la evaluación del estado del canal resulta un canal libre, la subcapa MAC debe iniciar con la transmisión del paquete inmediatamente (paso 5).

La intención general de CSMA-CA es incrementar el tiempo de espera para transmitir un paquete, mediante el ajuste del tamaño de la ventana Backoff (paso 4), conforme se detecta una canal ocupado (significa una alta demanda de uso del canal). Por el contrario, si el canal se encuentra libre (lo que significa una baja demanda del canal),

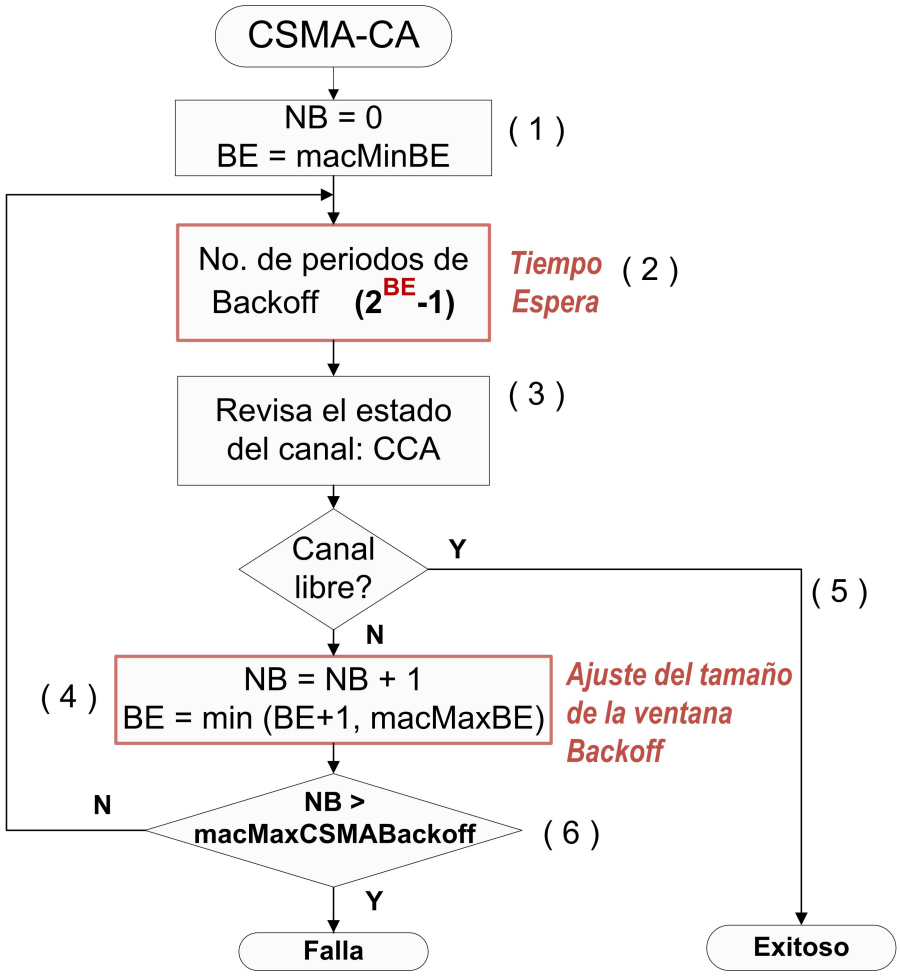

Figura 2.22: Algoritmo CSMA-CA para modo No-Baliza.

CSMA-CA usa el valor más bajo de BE (BE= macMinBE) para acelerar la transmisión del paquete en curso.

Los valores de macMinBE, macMaxBE y macMaxCSMABackoff son establecidos por el estándar IEEE 802.15.4 y se detallan en la Tabla 2.3. Cada variable puede tomar un rango de valores dentro de un margen permitido; los cuales pueden ser configurados por las capas superiores a la subcapa MAC.

La configuración de estas variables juega un papel importante en la operación del IEEE 802.15.4 y por ende del desempeño final de la WSN. Por ejemplo, un valor inicial pequeño de BE provoca que el dispositivo agilice la evaluación del canal (2^{BE}-1), lo cual

Tabla 2.3: Parámetros de CSMA-CA de la subcapa MAC

Parámetro	Valores permitidos [LAN/MAN Standard, 2006]
macMaxCSMABackoffs	Rango de 0 a 5 (por defecto: 4)
macMaxBE	Rango de 3 a 8 (por defecto: 5)
macMinBE	Rango de 0 a macMaxBE (por defecto: 3)

reduce el tiempo de transmisión del paquete. Sin embargo, aumenta la probabilidad de colisiones al incrementar el tráfico en la red. Además, si BE es pequeño, el rango de períodos de Backoff ([0 a 2^{BE}-1]) que un dispositivo puede tomar al inicio puede ser insuficiente; aumentando la probabilidad de que más de un dispositivo obtengan el mismo tiempo de espera (ya que todos los dispositivos inician con el mismo valor de BE) y puedan transmitir al mismo tiempo. Resultando en un problema de colisión de datos. Para el caso de macMaxCSMABackoff, si toma valores grandes reduce el número de fallas de acceso al canal; al permitir más intentos de transmisión del paquete en curso. Sin embargo, se incrementa el tiempo de entrega del paquete a su destino, al retenerlo por más tiempo en el dispositivo transmisor.

Por defecto el estándar fija los valores iniciales para cada variable, como se aprecia en la Tabla 2.3; sin embargo, hay estudios que argumentan que los valores no son los más adecuados en términos de desempeño de la red [Rohm et al., 2009], [Anastasi et al., 2011], [Rao and Marandin, 2006]. Por lo tanto, la configuración óptima de estas variables son parte de éste trabajo de investigación y se expondrán en el capitulo correspondiente.

2.6 Conclusiones del Capítulo

En este capítulo se hace una revisión del marco teórico requerido para la realización del presente trabajo. Inicialmente se describió las aplicaciones y requerimientos de las WSNs para la monitorización de pacientes de forma remota.

Se mostró como la congestión, ya sea en el nodo o en el enlace, es uno de los problemas que afectan fuertemente el rendimiento global de la red y los niveles de QoS ofrecidos. Ya que la congestión es causa de paquetes perdidos, consumo de energía y aumento del

retardo en la entrega del paquete.

Se expuso que la congestión en las WSNs depende fuertemente del tráfico generado por las aplicaciones, el medio inalámbrico y las características de operación del protocolo de acceso al medio. Por ello, la capa de Transporte y subcapa MAC, son fundamentales para lograr una comunicación eficiente; incluso, se justificó que estas entidades puedan trabajar en conjunto para mejorar sus resultados.

Se evidenció que las interacciones Cross-Layer son bien consideradas para la optimización de la operación de los protocolos de comunicación de las WSNs. Además, se puntualizaron los retos que significa el diseño Cross-Layer. Una importante conclusión fue evitar el uso de múltiples capas, capas sin sentido a los objetivos de optimización planteados; y seguir manteniendo un diseño general en capas.

El presente trabajo seleccionó el estándar IEEE802.15.4, por ser uno de los esfuerzos formales más completos y utilizados para el desarrollo de redes WSNs en aplicaciones médicas. Seleccionamos el modelo de operación sin baliza, debido a que presenta mayor eficiencia en cuanto a consumo de energía, menor retardo de transmisión y mejor desempeño con tráfico asimétrico. Se seleccionó la frecuencia de operación de 2.4 GHz debido a que presenta la velocidad más alta de transmisión del estándar de 250 kbps. Así mismo, se detalló la operación del algoritmo de acceso CSMA-CA de IEEE802.15.4; éste cuenta con los mecanismo para "regular" el uso del medio común por todos los dispositivos de la red.

CAPÍTULO 3

ESTADO DEL ARTE

En este capítulo se describe el estado del arte sobre protocolos de transporte que proponen una solución al problema del *Control de Congestión en las redes WSNs*. Estas soluciones, en general, buscan ofrecer un transporte confiable de datos entre los nodos de la red, considerando las características únicas de las WSNs. En este sentido, no se incluyen soluciones diseñadas para redes cableadas o redes inalámbricas convencionales; ya que por sus características, no son una opción viable para las WSNs, como se expuso en la sección 2.3.2 y lo argumenta Akyildiz ([Akyildiz and Vuran, 2010]).
Derivado del estudio de las diferentes propuestas, además, proponemos las *consideraciones y retos* más importantes que un protocolo de transporte debe cubrir para el control de la congestión en las redes WSNs.

3.1 Protocolos de Transporte para el Control de congestión en WSN

Lo principales mecanismos de un protocolo para el control de congestión se pueden agrupar en tres módulos, como son:

1. *Detección de congestión*, se refiere a la identificación de posibles eventos que crean congestión en la red. Los eventos pueden ser medir el nivel del búfer del nodo, nivel de carga en la red, retardo de envío, entre otros. Este mecanismo primero debe detectar si la congestión ocurre o no, y cuándo y dónde ocurre.

2. *Notificación de congestión*, cuando la congestión se presenta, el nodo que la detecta debe notificarla a los nodos de la red para que tomen consciencia de ello. Principalmente hay dos maneras de hacerlo, de forma *explícita*, mediante la construcción de un paquete de control especial; o de forma *implícita*, incrustando en el paquete de datos la información de control. Esta última manera se conoce como "llevar a cuestas" o piggyback.

3. *Resolución de congestión*, los protocolos debe evitar la congestión en la red; pero si se presenta, deben resolverla. La resolución puede eliminar la congestión o reducirla. Generalmente se resuelve mediante el ajuste de las tasas de transmisión de datos de los nodos sensores; sin embargo, hay otras formas que pueden utilizarse en las WSNs.

Diversos protocolos han sido propuestos para el control de congestión que difieren en cómo implementan los mecanismos para estos tres módulos. A continuación se presenta el estado del arte clasificado de acuerdo a cada uno de los tres módulos del control de congestión.

3.1.1 Detección de congestión

A continuación se presentan algunos protocolos diseñados para las WSNs para la detección de congestión; los cuales utilizan esta función en cada nodo de la red. El protocolo STCP [Iyer et al., 2005] utiliza el grado de utilización del búfer en cada nodo

involucrado en una transmisión de datos, como el evento para detectar congestión de forma local. Al igual que lo hace el protocolo HOCA para iniciar el algoritmo de resolución de congestión [Rezaee et al., 2014]. Por su parte CTCP [Giancoli et al., 2008] determina la presencia de congestión cuando se rebasa un umbral de búfer y una tasa de error de bit (tasa de pérdida de paquetes), mediante el mecanismo de reconocimiento de paquetes (Ack) en modo *salto por salto*. Debido a este mecanismo y al nivel de bufer, CTCP puede distinguir si la causa de congestión es por desbordamiento de búfer o por errores de transmisión. En CODA [Wan et al., 2003], además del nivel de ocupación del búfer, se utiliza una combinación de las condiciones de carga actual y pasada del canal inalámbrico, para inferir una detección más precisa de congestión en cada receptor. De forma similar TRCCIT [Shaikh et al., 2010] utiliza la tasa de datos de la red, donde compara la tasa de arribos de paquetes contra la tasa de envío de paquetes del nodo, para declarar congestión en el nodo. TRCCIT asume un control de congestión pro-activo, al monitorear la carga local de la red y adaptarse en consecuencia. RT^2 [Gungor et al., 2008] define un umbral de retardo promedio del nodo, con el cual, el nodo tiene una idea acerca del nivel de contención o saturación del medio en su entorno; y lo combina con un nivel de búfer para detectar de forma precisa la congestión. SenTCP [Rahman et al., 2008b] utiliza la relación del tiempo promedio de arribo de paquetes (TAP) contra el tiempo promedio de servicio de paquetes (TSP), en combinación con el nivel de uso del búfer para detectar el grado de congestión local en cada nodo. PCCP [Wang et al., 2007], al igual que SenTCP, utiliza el tiempo promedio de arribo de paquetes y el tiempo promedio de servicio de paquete en la subcapa MAC, para determinar el *grado de congestión* $C(i)$ actual en el nodo y/o del enlace. Este índice de congestión es definido como $C(i) = t^i_{Sp}/t^i_{Ap}$, entonces cuando $C(i) > 1$ asume que el nodo experimenta congestión; ya que la frecuencia con la que recibe paquetes es mayor a la frecuencia con la que puede transmitirlos. PCCP nombra a ésta forma de detección como *Detección de Congestión Inteligente*. DPCC [min LIN et al., 2011] es similar al protocolo PCCP; sin embargo, cambia la medida de tiempos a paquetes. Esto es, la congestión la determina al medir la cantidad de paquetes que arriban entre la cantidad de paquetes que transmite el nodo. Además, la medición de arribo de paquetes la realiza en la capa de transporte y no en la subcapa MAC, como lo hace PCCP.

LCART [Sharif et al., 2010] detecta la congestión de la red mediante el uso simultáneo de varios eventos como el TSP, TAP, el nivel de uso del búfer y un umbral de retardo extremo a extremo.

3.1.2 Notificación de congestión

CODA, CTCP, RT^2 y HOCA utilizan la notificación explícita para difundir el aviso de congestión; sin embargo lo hace de forma diferente. CODA crea un paquete especial para notificar de la congestión y lo envía hacia los nodos fuente en un modo de comunicación salto por salto; el nodo que recibe este tipo de mensajes reduce la tasa de envío y retransmite el aviso de congestión a los nodos vecinos, esta forma es conocida como *mensajes de retroceso o Backpressure*. Por su parte, CTCP utiliza la notificación explícita para indicarles que no puede recibir más paquetes de ningún nodo por el momento. De la misma forma, cuando se resuelve la congestión, los nodos son informados para re-establecer las tasas de transmisión. HOCA también necesita enviar la notificación de forma explícita mediante el envío de un paquete de control salto por salto desde el nodo congestionado hasta el nodo de origen. Pero a diferencias de los dos anteriores, necesita incorporar información del ajuste de la tasa de transmisión de datos a cada nodo fuente cuando se presenta la congestión.

La mayoría de las propuestas de notificación de congestión utiliza el concepto de piggyback para enviar de forma implícita el aviso de congestión a los nodos de la red (STCP, SenTCP, TRCCIT, LCART, PCCP, CRRT, DPCC). STCP incorpora esta forma de notificación y lo realiza de dos maneras: dentro del paquete de datos y dentro del paquete de Ack; fijando un bit en el campo de *Notificación de Congestión* (NC) del encabezado del paquete. En STCP cada nodo de la red tiene la posibilidad de detectar la congestión y generar la notificación hacia el Sink para informarle del estado de la red, a su vez el Sink genera el aviso de congestión utilizando el paquete Ack hacia los nodos fuente. SenTCP usa una notificación salto por salto incluyendo en el paquete el grado de congestión y el nivel de uso del búfer a los nodos de la red, para que los nodos puedan usarla en su mecanismo de resolución. LCART recibe la notificación de congestión de forma implícita, pero es recibida de dos manearas: *(i)*cuando un nodo recibe más paquetes que los que puede transmitir emite una señal de congestión en cada

paquete que envía al Sink. *(ii)* cuando el retardo extremo a extremo rebasa un umbral, en este caso el Sink detecta y notifica al nodo sensor. TRCCIT utiliza la notificación implícita salto por salto mediante la activación de un bit en cada paquete que sale del nodo; pero a diferencia de STCP y LCART, TRCCIT avisa de la congestión local de forma inmediata a los nodos vecinos y no tiene que ir al Sink para ello. PCCP utiliza el mismo tipo de notificación implícita que STCP y TRCCIT pero no sólo cambia el estado de un bit, si no que agrega información útil al mensaje de notificación: nivel de prioridad global, t^i_{Sp}, t^i_{Ap} y cantidad de nodos hijos. Los nodos que reciben el mensaje realizan acciones diferenciadas de control de la congestión, en base a la información recibida. Además, PCCP tiene dos eventos que pueden provocar el envío del mensaje de notificación, la primera ocurre cuando el número de paquetes reenviados por el nodo rebasa un umbral predefinido y la segunda cuando el nodo recibe el mensaje de notificación de congestión enviado por otros nodos. DPCC es similar que PCCP, necesita incluir información de la prioridad local y global del nodo dentro del paquete de datos, para proporcionar la información que necesitan sus nodos vecinos para su control de congestión.

3.1.3 Resolución de congestión

CRRT [Alam and Hong, 2009] implementa un mecanismo centralizado en el Sink para eliminar la congestión de la red en modo *extremo a extremo*; esta forma contribuye a una decisión imparcial, ya que el Sink tiene una visión global de la red y puede controlar la velocidad de todos los nodos en la red. El mecanismo base para resolver la congestión en CRRT es el esquema "Incremento Aditivo y Decremento Aditivo" (AIAD, Additive Increase Additive Decrease), el cual ajusta gradualmente la tasa de transmisión en lo nodos fuente evitando una reducción agresiva. En STCP cuando un nodo recibe la notificación de congestión, éste puede en-rutar los paquetes sucesivos del flujo de datos por una ruta diferente, siempre y cuando se tenga un algoritmo en la capa de red que permita este proceso; o disminuir la velocidad de transmisión de los nodos fuente. CODA utiliza dos mecanismos para resolver la congestión: un esquema de lazo abierto salto por salto y un esquema de lazo cerrado extremo a extremo. En el primer esquema, los nodos utilizan el método Backpressure para propagar el aviso e indicarles

a los nodos vecinos reducir su tasa de envío, esquema conocido como auto-regulación de nodo fuente. En el esquema de lazo cerrado, el que controla la tasa de transmisión de los nodos fuente es el Sink, que emite un mensaje de regulación hacia todos los nodos cuando existe congestión persistente, esquema llamado regulación múltifuente. CODA decide que esquema utilizar de acuerdo al grado de congestión de la red, si la tasa de eventos de los nodos fuente está por debajo de cierta fracción de la capacidad teórica del canal utiliza auto-regulación de nodo; de lo contrario se utiliza regulación desde el Sink. LCART es similar a CODA, ya que utiliza un esquema de lazo abierto y lazo cerrado para ajustar las tasas de transmisión de los nodos fuente. Sin embargo, la activación de cada caso es diferente; LCART activa el lazo cerrado cuando el retardo extremo a extremo rebasa un umbral definido y el lazo abierto cuando un nodo recibe más paquetes que los que puede transmitir. SenTCP realiza el control de la congestión en lazo abierto; con ajuste de la tasa de transmisión del nodo en base al grado de congestión y al nivel de uso del búfer, en relación con sus nodos vecinos. PCCP implementa un control de congestión distribuido salto por salto, de acuerdo al *grado de congestión* $C(i)$ *y a los índices de prioridad del nodo*; con lo cual garantiza que el nodo que tiene un mayor índice de prioridad obtenga más ancho de banda de la red. Cada nodo cuenta con un manejador de paquetes de dos "búfers", entre la capa de RED y la subcapa MAC, uno para el tráfico propio y otro para el tráfico en tránsito que recibe de otros nodos. Entonces, de acuerdo a la prioridad definida para cada tráfico el nodo ajusta y programa el envió de sus datos dentro del programador de paquetes. Bajo esta base, PCCP cuenta con una resolución de congestión flexible, diferenciada y distribuida para ofrecer *equidad ponderada* a los nodos de la red. DPCC es muy similar a PCCP, al enviar la prioridad local y global de cada nodo para el ajuste de la tasa de transmisión. Sin embargo, DPCC tiene el programador de paquetes entre la capa de Transporte y la subcapa MAC. HOCA utiliza múltiple rutas para el envío de paquetes, cuando el tiempo de entrega del paquete al extremo rebasa un umbral predefinido; ésto lo hace para prevenir la congestión de la red. Sin embargo, si la capacidad de las múltiples rutas ya no pueden cumplir con la meta, la congestión se presenta; entonces, ejecuta un algoritmo de resolución de congestión que reduce la tasa de transmisión en los nodos fuente. TRCCIT opera de forma similar a HOCA, pero TRCCIT resuelve la congestión

de forma pro-activa mediante la selección de varias rutas para redirigir el tráfico cuando detecta la congestión. La definición de congestión pro-activa se refiere a la capacidad de seleccionar *sobre la marcha* una ruta múltiple. TRCCIT evita la congestión en la red, esto lo fundamenta al resolver rápidamente la *congestión transitoria* en los nodos. Entonces, en la medida que un nodo se adapta o resuelve la congestión transitoria evita el incremento y una condición prolongada de congestión en una trayectoria dada. RT^2 y CTCP usan la reducción de la tasa de transmisión para controlar la congestión. Sin embargo, hay una diferencia fundamental; RT^2 reduce la tasa de transmisión de los nodos fuente mediante una indicación de ajuste a cada nodo. Por su parte, CTCP sólo informa a los nodos vecinos que no puede recibir más paquetes por el momento. Esto último obliga al resto de los nodos a suprimir el envió de mensajes y a reactivarlo cuando reciben el aviso de no-congestión.

3.2 Clasificación de los Protocolos de Control de Congestión

A partir de los trabajos consultados, expuestos en la sección 3.1, realizamos una *propuesta de clasificación* de las soluciones existentes para el control de congestión en las redes inalámbricas de sensores. Específicamente destacamos los mecanismos utilizados para los módulos de Detección, Notificación y Resolución de la congestión. Esta clasificación se presenta en la Tabla 3.2. De ésta, se puede resaltar la preferencia de medir el nivel de búfer y la diferencia entre los paquetes que arriban al nodo contra los que puede servir, para la detección de congestión. Para la notificación prefieren la forma implícita. El ajuste de la tasa de transmisión en los nodos y el uso de múltiples rutas como mecanismos para mitigar la congestión es lo más utilizado. De igual forma se aprecia que sólo cinco propuestas utilizan el diseño Cross-Layer; sin embargo, hacen uso limitado de esta capacidad de diseño (excepto por LCART). Ya que usan este diseño sólo para recibir datos de la subcapa MAC, pero no hay una interacción para modificar parámetros o cambiar algún comportamiento entre las capas.
Solamente STCP, RT^2, LCART, DPCC y HOCA consideran la diferenciación del tráfico por el tipo de aplicación, para otorgarle privilegios de comunicación; lo cual es un requi-

sito importante en las redes WSNs con aplicaciones heterogéneas. PCCP sólo considera prioridad a nivel nodo y el resto de acuerdo al grado de congestión experimentado por cada nodo. Sin embargo, soló HOCA da cuenta de los retos de QoS que demandan las aplicaciones de cuidado de la salud a través de una WSN. Con la limitante de concentrarse sólo en las métricas de consumo de energía y retardo del paquete; dejando fuera dos métricas importantes: la tasa de paquetes perdidos y el nivel de Throughput de la red (Sec.2.1.2). Además, ninguna propuesta considera en su diseño y evaluación al estándar IEEE 802.15.4, que es uno de los más utilizado para la construcción de las WSNs (Sec.2.5). Por ello, se hace necesario una investigación sobre el desarrollo de protocolos de transporte para el control de la congestión en las WSNs, que sean consientes de Qos de la aplicaciones.

3.3 Consideraciones Generales de Diseño para el Control de Congestión

En esta sección proponemos las *consideraciones de diseño* más importantes en el desarrollo de un protocolo de Transporte para el control de congestión en las WSNs. Esta propuesta considera las aplicaciones del seguimiento de la salud a través de una WSN, de la siguiente manera:

1. *Soporte de tráfico Heterogéneo*, este requisito es un punto clave para diferenciar los diferentes tipos de datos que generan las aplicaciones en la misma red. Además, que considere que existen flujos de datos de tipo "Continuo y por Eventos".

2. *Control de congestión dinámico*, que le permita adaptarse a las prioridades de la aplicación y al estado actual de la red; a través del cual pueda asignar diferentes mecanismos de resolución a los diferentes niveles de prioridad.

3. Ofrecer *equidad ponderada* a los nodos de la red en el uso del canal de comunicaciones, que asigne mayor acceso al canal de comunicaciones al tráfico de las aplicaciones con mayor prioridad.

Tabla 3.1: Clasificación de los protocolos analizados.

Propuesta	Detección	Notificación	Resolución	Diseño
STCP	Nivel de búfer	Implícita	Ajusta velocidad Redirige tráfico	Tradicional
SenTCP	Nivel de búfer Tiempo de Servicio de P. Tiempo de Arribo de P.	Implícita	Ajusta velocidad	Tradicional
TRCCIT	Carga de la red	Implícita	Ajusta velocidad Redirige tráfico	Tradicional
CODA	Carga de la red Nivel de búfer	Explícita	Ajusta velocidad Tira paquetes	Tradicional
CTCP	Tasa de error de Trans.	Explícita	Ajusta velocidad	Tradicional
RT^2	Tasa de error de Trans.	Explícita	Ajusta velocidad	Cross-Layer
LCART	Carga de la red Nivel de búfer Tiempo de Servicio P. Tiempo de Arribo P.	Implícita	Ajusta velocidad	Cross-Layer
PCCP	Tiempo de Servicio P. Tiempo de Arribo P.	Implícita	Ajusta velocidad	Cross-Layer
CRRT	Nivel de búfer Tasa de datos	Implícita	Ajusta velocidad	Tradicional
HOCA	Nivel de búfer Tasa de datos	Explícita	Ajusta velocidad Redirige tráfico	Cross-Layer
DPCC	Tasa de Servicio P. Tasa de Arribo P.	Implícita	Ajusta velocidad	Cross-Layer

4. Centrar el diseño en *evitar congestión*, más que pensar en mitigarla; lo cual necesita de mecanismos rápidos y efectivos de detección y notificación.

5. Con la intención de optimizar la notificación de congestión es necesario que el mecanismo de detección distinga casos de *congestión transitoria*; que pueden ser resueltos de forma local con el objetivo de evitar que la congestión evolucione a niveles más severos.

6. Diferenciar el *origen de la pérdida del paquete*, ya sea por desbordamiento del búfer o por problemas del enlace inalámbrico, es vital para las decisiones del módulo de resolución de congestión.

7. Es importante que el módulo de Detección se *retroalimente de los parámetros de confiabilidad*, tales como porcentaje de paquetes perdidos, retardo en la entrega de los paquetes, tasa de error de bit del canal, entre otros; para contar con mayor información que optimice el proceso de evaluación del módulo de detección. Lograr la interacción de la capa de Trasporte con el canal de comunicaciones es fundamental para este fin.

8. Para lograr el punto anterior el protocolo de transporte debe ser capaz de trabajar de forma integrada con otras capas, tales como la de Aplicación, la de RED y la subcapa MAC.

9. Habilidad para decidir cuándo operar en modo *salto por salto y/o extremo a extremo*, para agilizar el tiempo de respuesta de los procesos de detección, notificación y resolución.

10. *Método de notificación eficiente*, es recomendable que la notificación de congestión no incremente el tráfico en la red al momento de enviar los paquetes de control; ya que esta acción contribuye al aumento de la congestión en la red. El *método implícito* es una de las formas para lograr este objetivo, ya que utiliza el concepto de piggyback para usar los paquetes de datos para incluir la notificación. Además, se debe notificar a los nodos de la red que la congestión se ha resuelto, para restablecer la operación normal de la red.

11. Ligado al punto anterior, el protocolo de transporte debe minimizar el uso de mensajes de control sin comprometer el nivel requerido de rendimiento de datos.

12. El *consumo de energía* tiene que estar presente en todas las etapas y procesos del control de congestión, con el fin de extender la vida útil de la red. Por lo tanto, es necesario reducir la pérdida de paquetes y mantener al mínimo la carga de tráfico de paquetes de control.

13. *Escalabilidad*: Las redes de sensores pueden comprender un gran número de nodos, por lo tanto, el protocolo debe ser escalable. Sus mecanismos de operación no deben verse afectado por el crecimiento del número de nodos en la red.

Mantener el conjunto de estas consideraciones para desarrollar un protocolo de Transporte para el control de congestión, involucran una serie de retos para las diferentes capas del modelo OSI; especialmente la capa de transporte y la subcapa MAC. Además, requiere del paradigma de diseño Cross-Layer para optimizar su operación y no romper con el modelo de diseño en capas. En el estado del arte, encontramos propuestas que consideran aplicaciones bastante simples y no cumplen con la mayoría de las consideraciones expuestas (Sec.3.2). Otras que incluyen un buen número de estas consideraciones, pero necesitan de esquemas de enrutamiento de múltiples rutas o de un control centralizado en el Sink; que son esquemas no adecuados para las WSNs.

3.4 Conclusiones del Capítulo

En este capítulo se expusieron los principales trabajos relacionados al problema de congestión en las WSNs. Además, se presentó una clasificación de los principales protocolos de transporte para el control de congestión; clasificados por los tres principales mecanismos de control de congestión: Detección, Notificación y Resolución. Se propusieron las consideraciones principales que deben ser cubiertas por los protocolos para el control de congestión en las redes WSNs; contemplando las aplicaciones médicas de supervisión de la salud.

Por último, se determinó que, para cumplir con los objetivos de control de la congestión es necesario construir el protocolo bajo el nuevo paradigma de interacción Cross-Layer,

que la capa de Transporte y la subcapa MAC juegan un papel vital en la operación de las WSNs, y que es necesario que se tome en cuenta los requerimientos QoS de las aplicaciones, en los procesos del control de congestión.

CAPÍTULO 4

PROPUESTA DEL PROTOCOLO DE TRANSPORTE PCCQ.

Como se evidencio en el capitulo 2 y 3 *el problema de la congestión* en las WSNs tiene un impacto directo en el desempeño global de la red, y por tanto, en la QoS que éstas pueden ofrecer a las aplicaciones. Cuando la congestión se presenta, la red empieza a perder paquetes y a incrementar el tiempo de entrega de los mismos, afectando las métricas de operación QoS (Retardo, Paquetes Perdidos y Desempeño, Sec.2.2.2). Entonces, en la medida que se controle la congestión, se mejoran los niveles de QoS.
Para resolver este problema, proponemos un **P**rotocolo de Transporte para el **C**ontrol de **C**ongestión, consciente del **Q**oS de las aplicaciones, llamado PCCQ. *PCCQ* controla la congestión de forma distribuida, encada nodo de la red; y de forma dinámica, de acuerdo al grado de congestión experimentado por el nodo y a la QoS del tipo de tráfico de aplicación. *PCCQ* esta diseñado bajo el concepto Cross-Layer, ya que la interacción múlti-capa le permite optimizar el uso de los recursos de la red y los niveles de QoS ofrecidos; mientras mantiene el esquema general de protocolos por capas, que asegura

la interoperabilidad. Para su diseño se utilizó la capa de Transporte y la subcapa MAC, ya que son las entidades que más impacto tienen para controlar la congestión y la calidad de la comunicación de datos.
A continuación se describe la arquitectura de diseño, el modelo del sistema y los módulos que componen el protocolo *PCCQ*.

4.1 Arquitectura de Diseño del protocolo PCCQ

La arquitectura de diseño Cross-Layer, bajo la cual descansa *PCCQ*, se muestra en la figura 4.1c. La arquitectura guarda el modelo en capas OSI (Fig. 4.1a) y usa como base la arquitectura del estándar IEEE 802.15.4 (Fig. 4.1b). Esta última, tiene habilitado una vía de acceso directo para interactuar con las capas superiores, con dos puntos de conexión (Sec.2.5.2): para manejar lo datos de la aplicación (MCPS-SAP) y para los datos de control (MLME-SAP). El protocolo *PCCQ* reside principalmente en la capa de Transporte, y se interconecta con la subcapa MAC gracias a la *interación* Cross-Layer, y mediante el diseño especifico de las interfaces de conexión a los puntos MCPS-SAP y MLME-SAP (Fig.4.1c). Por lo tanto, en nuestra propuesta, la definición Cross-Layer se entiende como: la capacidad del protocolo para acceder a la información de parámetros de dos o más capas, los cuales puede utilizar y/o modificar con un mismo fin.

4.2 Modelo de Sistema de *PCCQ*

Los modelos de RED y de NODO que componen el Sistema de *PCCQ* se muestran en la figura 4.2 y figura 4.3, respectivamente. Estos modelos son tomados en cuenta en el desarrollo del protocolo de Transporte para el control de congestión *PCCQ*.

4.2.1 Modelo de RED

El modelo de RED propuesto para la WSN se muestra en la figura 4.2. Este modelo es formado por clústeres en una estructura jerárquica de 4 niveles. Cada clúster, a su vez la conforma una red en estrella entre el nodo padre y los nodos hijos a su alcance. Por ejemplo, un clúster lo forman los nodos E, F, G y H, o los nodo A, B, C y D.

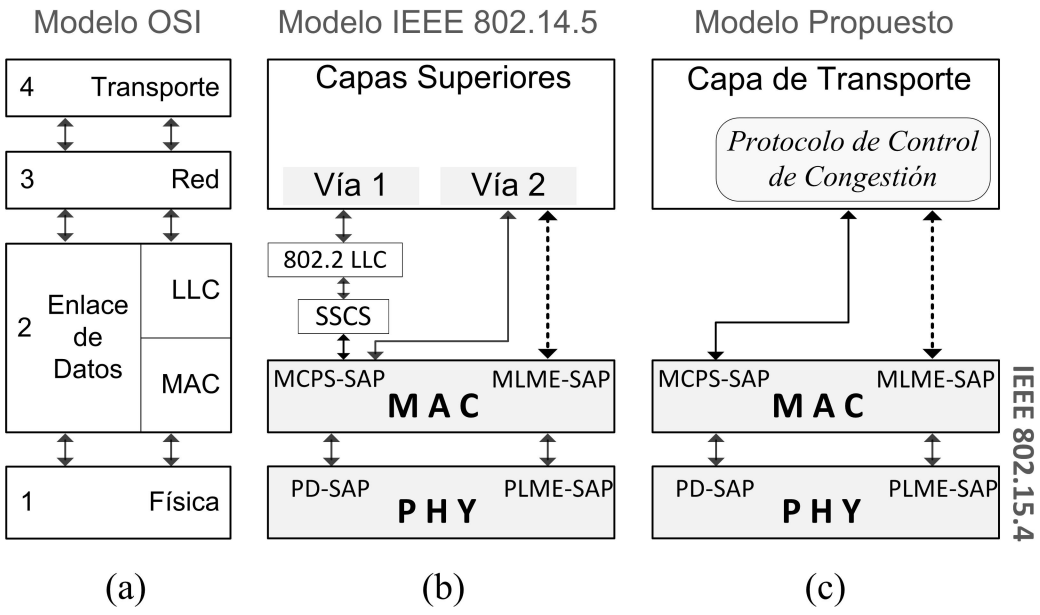

Figura 4.1: Arquitectura de Diseño del protocolo *PCCQ*.

El nivel 1 se compone de los nodos sensores o Nodos Fuente (NF) (peje. nodo F), cuya función principal es la recuperación de señales de interés. Este nivel es el único nivel (o nodo) que puede medir señales. Los nodos NF están considerados para que los porten las personas (Fig.2.3 del Cap.2), por ello, se requiere de movilidad. Sin embargo, consideramos que la movilidad es dentro del área de cobertura de un clúster; si la persona se aleja, puede reconectarse a otro cluster. Los procesos de transferencia entre clúster no están considerados en nuestro trabajo de investigación.

El segundo nivel lo forman los Nodos Coordinadores de clúster (NC) (peje. nodo E), cuya función principal es administrar la conectividad de los nodos dentro de su clúster y retransmitir los paquetes que reciba al siguiente nivel. Estos nodos están considerados para ser colocados en sitios fijos, cerca del entorno de las personas pero no en las personas; por lo cual, pueden considerarse con suficiente suministro de energía.

El tercer nivel lo forman los Nodos Retransmisores (NR), los cuales tiene la función principal de recibir los paquetes, de sus nodos NC, de otros nodos NR o incluso de algún nodo NF que este directamente conectado a él (peje. Nodo I al Nodo D); y retransmitirlos al siguiente salto. Este nivel puede tener varios saltos hasta llegar al

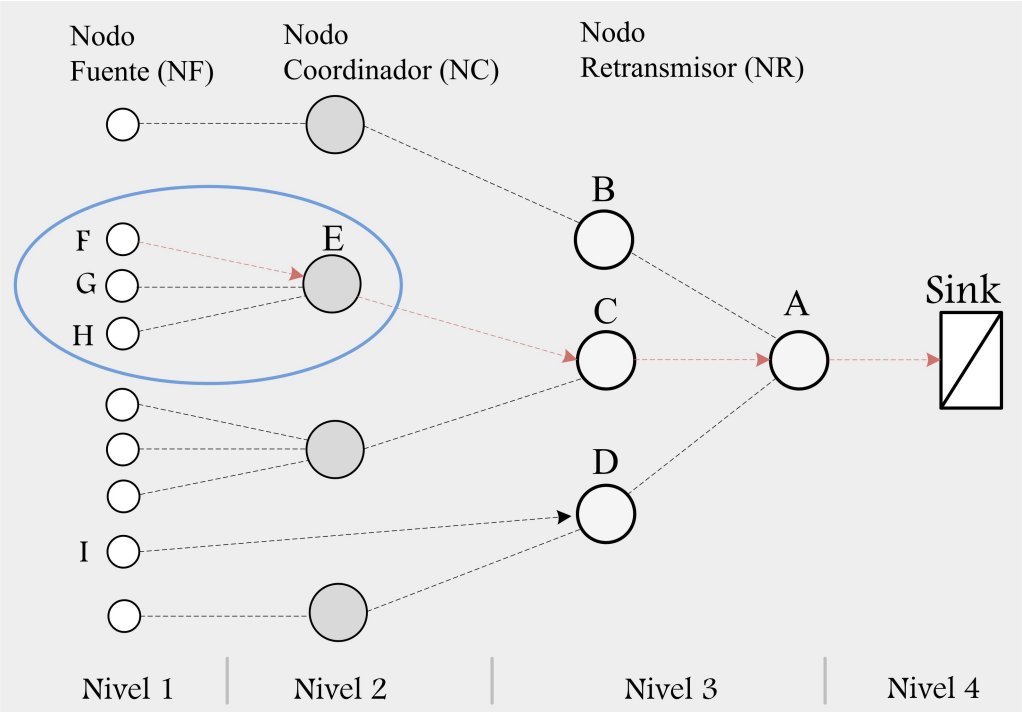

Figura 4.2: Modelo de RED propuesto para la WSNs.

último nivel donde se encuentra el nodo Sink. Los NR, al igual que los NC, son nodos fijos y conectados a fuente de energía. Por último, en el cuarto nivel se encuentra el nodo Sink, cuya función principal es recibir los datos para su posterior etapa de procesamiento. En nuestra propuesta se considera que hay un sólo nodo Sink, el cual es fijo, con mayores recursos de computación y energía que el resto de los nodos.

El uso de jerarquías en este modelo de RED simplifica el algoritmo de ruteo (para reenviar los mensajes a través de la red) y dividen las funciones de cada nivel. Por lo tanto, y dado que es una sola ruta hacia el Sink, el protocolo de ruteo es simple; proponemos un ruteo estático en cada nodo. Otro beneficio de este modelo es la topología en estrella de cada clúster, ya que esta topología ofrece los mejores rendimientos en cuanto al retardo y a la facilidad de sincronización de los nodos con el mínimo uso de paquetes de control.

Un punto clave del protocolo de Transporte es abstraerse lo más posible de las aplicaciones y concentrarnos sólo en el tipo de tráfico que éstas generan. El tráfico que se

considera para este modelo de RED lo generan los nodos NF. En este caso, se catalogan lo tráficos por el tipo de aplicación y las necesidades de comunicación que requieren (Tabla 2.1, Sec. 2.1.2), como: *(i)* Crítico y Urgente (P_1), para la señal de un ECG; *(ii)* Tráfico Crítico No-Urgente (P_2), como el recibido por los sensores de ritmo cardíaco y *(iii)* Normal (P_3), como los datos del sensor de temperatura. En general, estos tres tipos de tráfico se consideran que tienen un flujo de datos de tipo *convergente* hacia el Sink; en un esquema de ruta única de comunicación. Además, el tráfico es "continuo" y permanente, en intervalos de tiempo. Estas consideraciones pueden cubrir los requerimientos y el comportamiento de la mayoría de los signos vitales de una personas; dando posibilidad de considerar otros tipos de señales en la propuesta.

4.2.2 Modelo de NODO

El modelo general de NODO se presenta en la figura 4.3; sus funciones están desarrolladas principalmente en la capa de Transporte y en la subcapa MAC, de la pila de protocolos de comunicaciones. La subcapa MAC de cada nodo, utiliza el algoritmo CSMA-CA no-ranurado de IEEE 802.15.4 (Fig.2.22, Sec.2.5.4) para compartir el canal; que es la versión más sencilla de este estándar. Cada nodo tiene, además, una capa de Transporte que incluye el protocolo *PCCQ*, formado por el Control de Congestión y un Módulo de Priorización de paquetes; para gestionar la congestión y administrar el sistema de almacenamiento del nodo, respectivamente. La capa de Transporte esta conectada, vía Cross-Layer, a la subcapa MAC; específicamente para recibir los eventos que detectan la congestión de la red y para enviarle valores de configuración al algoritmo CSMA-CA (Detección y ajuste de la Fig. 4.3).

El origen del tráfico lo compone el **Tráfico Local**, generado por los sensores del *nodo i* a una tasa $\mathbf{r_{loc}}$; y el **Tráfico de Tránsito** que ingresa al *nodo i*, por la subcapa MAC a una tasa de $\mathbf{r_{tr}}$, de los *nodos i-1* o nodos hijos. Ambos tráficos se integran en el Módulo de Priorización para ser enviados a la capa MAC. Entonces, la tasa de entrada a este módulo es $\mathbf{r_{in}} = r_{loc} + r_{tr}$ y su tasa de salida $\mathbf{r_{prog}}$ esta dada por la operación interna del Módulo de Priorización. A su vez, r_{prog} se convierte en la tasa de entrada a la subcapa MAC ($\mathbf{r_{inMAC}}$).

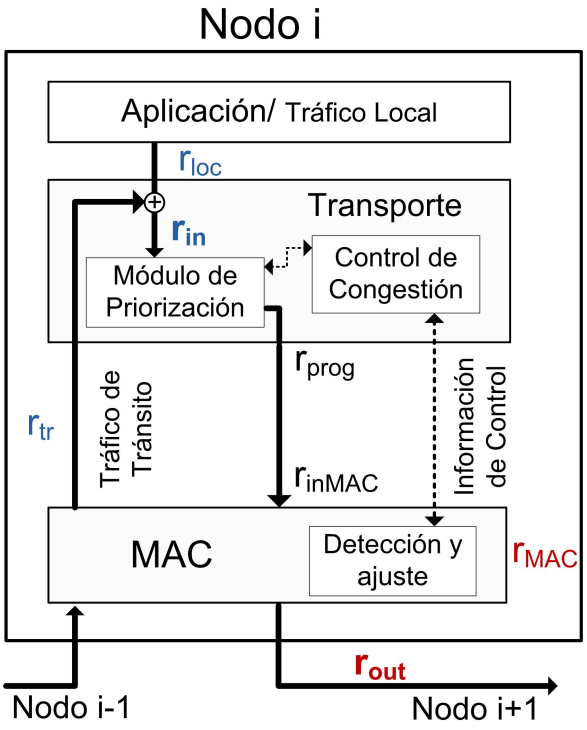

Figura 4.3: Modelo de RED propuesto para la WSNs.

La *tasa de servicio* ($\mathbf{r_{out}}$) del *nodo i* esta dada por la tasa de entrada (r_{in}) y la tasa de transmisión de la MAC ($\mathbf{r_{MAC}}$). r_{MAC} tiene que ver con la capacidad de la MAC para colocar paquetes en el canal inalámbrico; entonces, esta ligada a la operación del algoritmo CSMA-CA. Con estas consideraciones y a partir de la figura 4.3, r_{out} puede tomar los valores que se muestran en la ecuación 4.1.

$$r_{out} = \begin{cases} r_{in} & \text{si } r_{in} < r_{MAC} \\ r_{MAC} & \text{si } r_{in} > r_{MAC} \end{cases} \tag{4.1}$$

Por lo tanto:

$$r_{out} = min(r_{in}, r_{MAC}) \tag{4.2}$$

Esta propiedad (Ec.4.2) puede ser utilizada para reducir (o incrementar) indirectamente a r_{out} a través de la reducción (o incremento) de r_{in} y/o de r_{MAC}. Por ejemplo, cuando r_{in} es mayor que r_{MAC}, los paquetes pueden ser "encolados" en los búfer del Módulo de

Priorización. Dado que el nodo pierde su *equilibrio*, al recibir más paquetes de los que puede enviar (r_{out}); conduciendo inevitablemente a la congestión a *nivel de nodo*. Por otro lado, si hay colisiones en el enlace alrededor del *nodo i*, entonces el *nodo i* y sus nodos vecinos deberían reducir el acceso al canal, con el fin de evitar que se incremente la congestión a *nivel de enlace*. En ambos casos, cuando la congestión se presenta se pierden paquetes, tanto en el Módulo de Priorización de Paquetes (por desbordamiento de su búfer) como a nivel MAC (cuando el algoritmo CSMA-CA falla sus intentos de envío o CAF). Además, un búfer y un canal saturado incrementan el retardo de envío de paquetes al destino (Sec.2.2.1). Condiciones que afectan las métricas QoS ofrecidas a las aplicaciones.

Cuando hay un desequilibrio de nodo, podemos resolverlo incrementando (o reduciendo) las oportunidades de acceso al medio del algoritmo CSMA-CA, específicamente, ajustando los periodos de Backoff que el nodo debe esperar para transmitir (Sec.2.5.4). De igual forma, podemos resolver este problema al reducir la tasa r_{in}; ya sea, reduciendo el tráfico r_{loc} o r_{tr} (o ambos). El punto clave es mantener el equilibrio del nodo y realizar los ajustes de acuerdo a la prioridad del tráfico de aplicación. Estas propiedades del modelo de NODO son utilizados por el protocolo *PCCQ*, para el control de la congestión, como se vera en la sección 4.3.

4.3 Propuesta del protocolo de Transporte *PCCQ*

Para cumplir los objetivos y la problemática de investigación proponemos el protocolo de Transporte para el control de congestión *PCCQ*. Bajo nuestra perspectiva, un protocolo para el control de congestión en las WSNs debe incluir cuatro módulos: *(i)* Priorización de tráfico, *(ii)* Detección de Congestión, *(iii)* Notificación y *(iv)* Resolución de congestión. La figura 4.4 muestra el modelo funcional del protocolo *PCCQ* formado por estos cuatro módulos. Además, se muestran los elementos y acciones que cada módulo debe utilizar para lograr el control de la congestión.

El módulo de *Prirización* lo compone un Administración de Paquetes, ubicado en la capa de Transporte de cada nodo (ver Fig. 4.3). Este módulo diferencia el tráfico de origen sobre la base de la importancia de la aplicación (necesidades QoS); garantizando

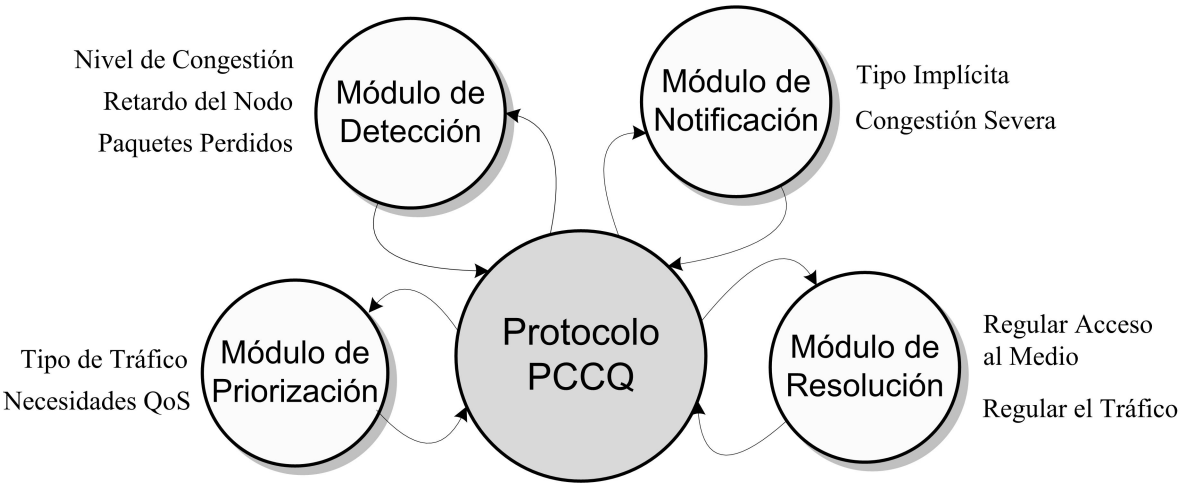

Figura 4.4: Modelo funcional del protocolo *PCCQ*.

que le ofrezca las mejores prestaciones de red al tráfico con mayor importancia (se detalla este módulo en la Sec.4.3.1).

El módulo de *Detección de Congestión* identifica la presencia de aquellos eventos que pueden conducir a la congestión de la red. Se proponen tres eventos: Grado de congestión (Tasa de arribo contra la tasa de salida de paquetes), retardo promedio de transmisión de paquete y tasa de paquetes perdidos por falla de acceso al canal (CAF); todos medidos en cada nodo (se detalla este módulo en la Sec.4.3.2).

El módulo de *Notificación de Congestión* debe difundir el aviso de congestión desde el nodo que la detecta hacia todos sus nodos vecinos (generalmente dentro de su clúster), ya que todos están involucrados en la congestión. Esta notificación es *implícita* y sólo se genera cuando el nodo alcanza un nivel "crítico" de congestión (se detalla este módulo en la Sec.4.3.3).

El módulo de *Resolución de Congestión* es activado en dos momentos: *(i)* Activación interna, por el módulo de detección propia del nodo y *(ii)* Activación Externa, al recibir un mensaje de notificación de otro nodo. La función del módulo de Resolución es prevenir que se presente la congestión, o en su caso, eliminarla o disminuirla (se detalla este módulo en la Sec.4.3.4).

Para la integración de los módulos dentro del protocolo se propone el diseño bajo interacción Cross-Layer, entre la capa de Transporte y la subcapa de acceso al medio

MAC; como se explicó en la arquitectura de nuestra propuesta (Fig.4.1). Con lo cual, la información y funciones de estas capas pueden trabajar en sintonía con el objetivo de controlar la congestión y optimizar los recursos de la red. En la capa de Transporte, se ubica la base de nuestro control de congestión (Fig.4.1); esta capa proporciona a la subcapa MAC la información de las *necesidades de operación* del protocolo y las *prioridades del tráfico* indicadas por la aplicación. Por su parte, la subcapa MAC proporciona información de las condiciones actuales del canal y del comportamiento del tráfico a la capa de Transporte, para estimar el grado de congestión al rededor del nodo.
Creemos que esta sinergia deberá mejorar los resultados de operación de los módulos de Detección, Notificación y Resolución de *PCCQ*. En consecuencia, la WSN debe alcanzar un mayor rendimiento en la métricas y objetivos QoS.

4.3.1 Módulo de Priorización

El proceso del protocolo inicia con el módulo de *Priorización*, que recibe el tráfico de las dos fuentes de llegada al nodo (tráfico local y tráfico de tránsito) para clasificarlos según la importancia de su origen. Este módulo parte del supuesto que:

> La capa de aplicación etiqueta cada tráfico (*Identificador de Precedencia*) según su importancia de acuerdo a la aplicación específica. Con esta etiqueta, le permite al nodo identificar los paquetes dentro de la red para ofrecer sus servicios de acuerdo a la aplicación.

> Al tráfico de alta prioridad lo etiqueta con el valor de P_1 (para el sensor ECG), al de prioridad media con el valor de P_2 (para el sensor de ritmo cardíaco) y al de baja prioridad con el valor de P_3 (para el sensor de Temperatura corporal), ver sección 4.2.1 y 2.1.2. Esta clasificación permite cubrir los diferentes tipos de tráfico que generan las aplicaciones médicas, debido a que tienen flujos de tráfico mixto con diferentes requisitos de QoS.

El módulo de *Priorización* lo compone un sistema de Administración de Paquetes, que está compuesto por tres elementos: *(i)* Clasificador de Paquetes, *(ii)* un Bloque de Almacenamiento de tres búfers (la cantidad se puede incrementar de acuerdo a las clases

de tráfico que se definan) y *(iii)* un Programador de Paquetes; como se muestra en la figura 4.5.

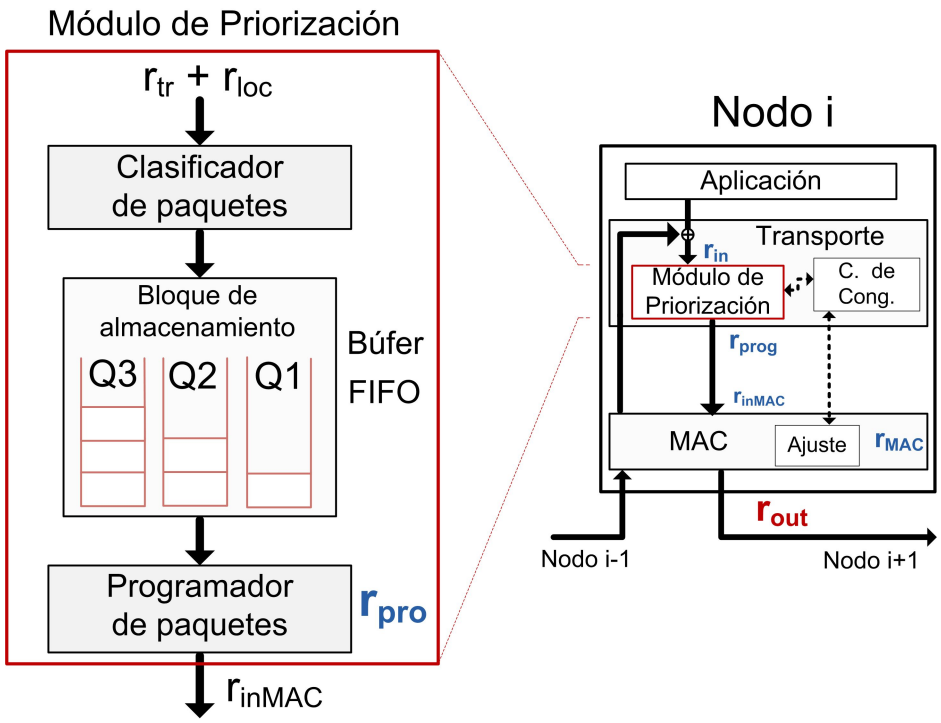

Figura 4.5: Esquema del módulo de Priorización de *PCCQ*.

El Clasificador revisa el Identificador de Precedencia, dentro del encabezado del paquete de datos, para dirigirlo al búfer correspondiente dentro del Bloque de Almacenamiento. Dentro del bloque hay tres búfers (Q_1, Q_2 y Q_3) del mismo tamaño y de diferente prioridad, para almacenar cada clase de paquete de acuerdo a su prioridad (P_1, P_2 y P_3). Por lo tanto, el tamaño total del Bloque de Almacenamiento esta dado por $T_{Lim} = Q_1+Q_2+Q_3$ (en cantidad de paquetes); entonces, cada búfer tiene un tamaño de $\frac{1}{3}(T_{Lim})$. Cada búfer acomoda los paquetes que recibe con el esquema básico "Primero en Llegar-Primero en Salir" (FIFO, First in-first out); en este esquema, cuando un bufer alcanza su limite de espacio, necesita "tirar" los paquetes nuevos que reciba hasta que se libere espacio. Para que un paquete pueda salir de un búfer, necesita de un mecanismo de servicio que lo atienda; en nuestra propuesta corresponde al Programador de Paquetes (Fig.4.5).

El Programador toma paquetes de los diferentes búfer para dirigirlos hacia la subcapa MAC, a una tasa de salida r_{prog}; de acuerdo a su mecanismo interno de servicio y las necesidades de QoS de las Aplicaciones. Para este mecanismo de servicio, se desarrolló un algoritmo de Ordenamiento Cíclico Ponderado para WSN (WWRR, WSNs Weighted Round Robin). WWRR es una versión creada a partir del algoritmo tradicional WRR, ya que éste nos permite manejar varios niveles de prioridad de forma simple y alcanza una gran velocidad de respuesta [Nasser et al., 2013]. WWRR cumple dos de las principales propiedades de un programador de paquetes: baja latencia y simplicidad; para una red de altas prestaciones con soporte para Qos [Francini et al., 2001]. La figura 4.6 muestra la operación del algoritmo WWRR, integrado al Programador de Paquetes de *PCCQ*. El algoritmo WWRR sirve los paquetes de cada cola en orden rotativo y según su prioridad o peso asignado; esto es, en cada turno, el Programador toma un número de paquetes de cada cola que es proporcional a su peso. Por ejemplo, la figura 4.6 muestra un caso de configuración para WWRR con una prioridad o peso de: $Q_1 = 3$, $Q_2 = 2$ y $Q_3 = 1$; entonces, en cada turno de servicio WWRR enviará a la subcapa MAC tres paquetes del búfer Q_1, dos del Q_2 y uno del Q_3.

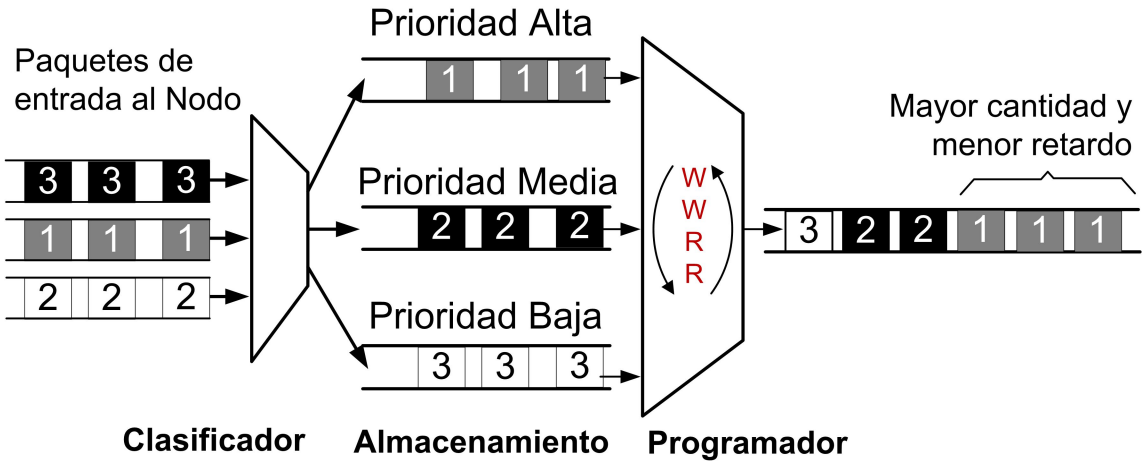

Figura 4.6: Operación del algoritmo WWRR del Programador de Paquetes.

WWRR mejora sustancialmente la versatilidad del Programado de Paquetes, ya que permite compartir los "recursos comunes" de forma controlada, decidiendo qué paquetes deberían tomar los recursos del nodo y usar el canal de comunicaciones. Esto facilita

cumplir los objetivos de QoS específicos de la aplicación y la equidad ponderada, asignando más ancho de banda para las aplicaciones más críticas.
El módulo de priorización, a través del Programador de Paquetes, apoya el problema de la congestión en las redes WSNs; ya que a través del ajuste de r_{prog} podemos controlar la cantidad de paquetes que debe atender la subcapa MAC. El ajuste puede ser hecho de forma dinámica durante la operación del protocolo *PCCQ*, gracias a la conexión que existe entre este módulo y el módulo de Control de Congestión (Fig.4.3). Esta posibilidad de ajuste es útil para contribuir a mantener el "equilibrio del nodo"; ya que podemos disminuir la cantidad de paquetes que salen del nodo (r_{out}) cuando existe congestión a nivel del enlace, de acuerdo a las ecuaciones 4.1 y 4.2 (Sec.4.2.2). Sin embargo, cuando esto se hace, los paquetes se empiezan acumular en el Bloque de Almacenamiento, formando una "Cola" en cada búfer; que provoca un incrementó del retardo de transmisión o de la cantidad de paquetes perdidos del nodo. Por ello, es importante hacer un ajuste dinámico e inteligente de la tasa r_{prog}, de acuerdo al nivel de congestión y a la prioridad de cada paquete, respectivamente. Para este ajuste proponemos 4 políticas de operación:

1. El trafico de alta prioridad (P_1) puede ocupar más del espacio asignado a Q1 hasta: $\frac{2}{3}(T_{Lim})$; incrementando las posibilidades de envío y reduciendo la pérdida de datos P_1.

2. El espacio extra que toma el tráfico P_1, es tomado exclusivamente del búfer Q_3, espacio reservado para el tráfico de más baja prioridad.

3. El espacio de Q_1 no puede ser usado por paquetes diferentes a P_1.

4. Cualquier configuración de peso, que se le asigne a cada búfer, debe cumplir con la siguiente relación $Q_1 > Q_2 > Q_3$.

Pruebas de Evaluación del Módulo de Priorización

En esta sección se presentan los resultados de evaluación del módulo de Priorización de paquetes, realizadas de forma independiente al resto de los módulos que forman el protocolo *PCCQ*. Esta evaluación tiene el objetivo de evidenciar la operación de este

módulo y las ventajas de QoS que ofrece a las aplicaciones de datos.

La prueba consistió en medir el *retardo promedio de salida de paquete*, del módulo de Priorización, para cada tipo de prioridad; esto es, el intervalo de tiempo desde que el paquete llega al Clasificador y hasta que éste sale del Programador (Fig.4.6). La medición fue realizada para dos casos: *(ii)* cuando los nodos de la red tienen una ponderación del algoritmo WWRR de $Q_1 = Q_2 = Q_3 = 1$; y *(ii)* cuando los nodos tiene una ponderación de $Q_1 = 4$, $Q_2 = 2$ y $Q_3 = 1$.

Para el escenario de pruebas se utilizó una WSN de 20 nodos sensores, los cuales generan los tres tipos de prioridad de tráfico. Generando una cantidad de tráfico tal que produjera congestión en la WSN de forma gradual. Para lo anterior, se consideró la capacidad teórica de transmisión de datos del estándar IEEE 802.15 de 250 kbps y paquetes de un tamaño de 80 bytes; entonces resulta:

$$Carga = 250kbps = 31250\ \frac{Bytes}{seg} * (\frac{1\,paquete}{80\,Bytes}) = 390.62\ p/s \tag{4.3}$$

El tráfico de la red inició con un sólo nodo, enviando paquetes por un periodo de tiempo, después se fue incrementando en uno la cantidad de nodos, hasta alcanzar los 20 nodos y un total de tráfico en la red de 400 p/s.

Para el desarrollo de las pruebas se utilizó el simulador de redes Network Simulator 2 (NS2) [Institute, 2008]. NS2 es una herramienta de software orientada a eventos que cuenta con módulos para implementar diferentes tipos de redes de comunicación; como las WSNs bajo el IEEE 802.15.4. Además, cuenta con algoritmos para los protocolos de cada capa del módelo OSI (Aplicación, Transporte, RED, etc.), con sistemas de almacenamiento y manejo de colas, con módulos para el registro detallado de la operación de la red y sus eventos, entre otros. NS2 es un software de diseño abierto, que permite la modificación de sus programas y la adición de nuevo software. En este sentido, incorporamos nuestro módulo de Priorización de paquetes a la capa de transporte de cada nodo en la red; con el fin de evaluar su operación con el resto de los protocolos de comunicación. Estos protocolos son: para la capa de física y subcapa MAC el estándar IEEE 802.15.4, el protocolo UDP para la capa de Transporte y un algoritmo de rutas estáticas para la capa de RED.

Las figuras 4.7 y 4.8 muestran los resultados obtenidos para estas pruebas. La figura 4.7 corresponde al caso cuando el Programador de paquetes no aplica ponderación al

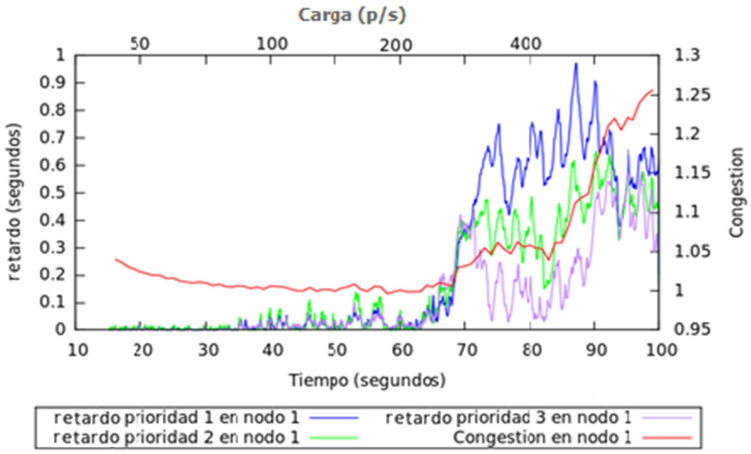

Figura 4.7: Retardo en el Nodo 1, con WWRR 1 1 1.

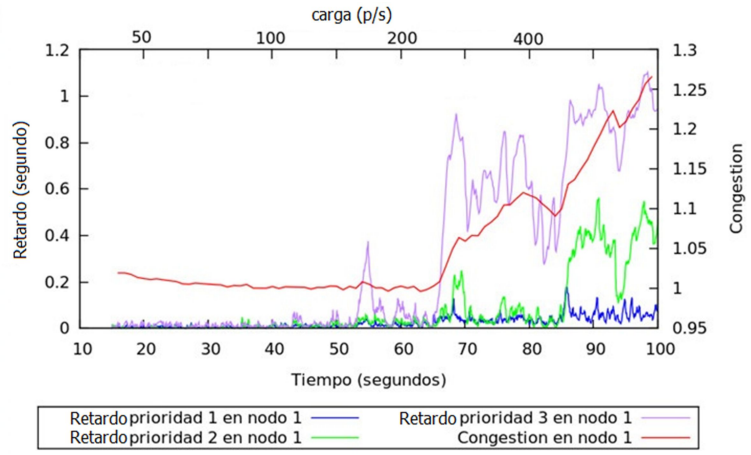

Figura 4.8: Retardo en el Nodo 1, con WWRR 4 2 1.

tráfico recibido. Observe que, cuando hay poca carga en la red (menor a 100 p/s) el retardo promedio es bajo para todos los tipos de tráfico (menor a 0.1 seg). Debido a que la demanda por el acceso al medio es baja, y los nodos puede transmitir sus paquetes casi inmediatamente conformen arriban al nodo. Esto significa que el nodo mantiene su "equilibro" en la mayor parte de tiempo ($r_{in} < r_{MAC}$, Ec. 4.1), debido a una congestión baja de la red (menor a 1.001, en la siguiente sección se explica). Por lo tanto, el grado de congestión refleja el nivel de equilibrio del nodo. Sin embargo, conforme la carga en la red empieza a incrementarse, el retardo se incrementa para los

tres tipos de tráfico. Por ejemplo, cuando la red alcanza la carga máxima (400 p/s) el retardo es alto y diferente para cada trafico, presentando el retardo más bajo el tipo 3 (P3) que es el tráfico con menores requerimientos de QoS; caso contrario para el tráfico más urgente (P1) que alcanza un retardo cercano a 1 seg tráfico de la red.
La figura 4.7, corresponde al caso cuando el Programador de paquetes aplica niveles de prioridad a los tipos de tráfico, para ser atendidos dentro de cada búfer. Cuando la carga es baja, es muy similar al resultado anterior, el nodo puede atender a todos los tipos de tráfico. Conforme la red se empieza a congestionar, el Programador atiende el trafico de cada búfer de forma diferencial. Observe que en el intervalo de simulación de 65 a 85 seg (eje X inferior). los tráfico tipo 1 y tipo 2 tienen los mejores tiempos; cuando la red alcanza los 400 p/s el único tráfico que se mantiene con los valores más bajos es el tipo 1, el cual tiene la ponderación más alta ($Q_1 = 4$).
En conclusión, el Programador de paquetes apoya al nodo a ofrecer QoS a cada tipo de tráfico que atiende; como se vio en las gráficas, el tráfico más crítico se mantuvo durante toda la simulación con los más bajos retardos de servicio. Se observó que conforme la red recibe más tráfico los niveles de congestión empeoran. Entonces se necesita de otros mecanismos que resuelvan la congestión.

4.3.2 Módulo de Detección de Congestión

El módulo de *Detección de Congestión* tiene la responsabilidad de identificar la congestión en etapas tempranas, que permita activar los mecanismos de resolución del nodo; con el objetivo de evitar que la congestión local evolucione a niveles más severos y se expanda a toda la red. Además, debido a la naturaleza del tráfico en las WSNs, se pueden presentar ráfagas de datos que rebasen la capacidad de la red por periodos cortos; provocando una *congestión transitoria*. Entonces, el protocolo debe ser capaz de adaptarse rápidamente a estos cambios y distinguir este tipo de congestión. La propuesta de este módulo es realizar la congestión de forma *distribuida* en cada nodo de la red, para agilizar la detección y hacerlo de forma independiente de otros nodos. Además, proponemos medir el *grado de congestión* más que sólo detectarla, para distinguir tres niveles de congestión: Bajo, Medio y Alto. Otro punto importante de nuestra propuesta es, que el módulo de Detección pueda distinguir si la congestión fue a nivel

del nodo (búfer saturado) o a nivel de enlace (canal inalámbrico saturado). Todo lo anterior, con el objetivo de ofrecer la mayor información posible para definir los mecanismos adecuados a utilizar para el control de la congestión.
A continuación se listan los eventos utilizados en nuestro módulo de Detección de Congestión, y la figura 4.9 muestra los puntos de medición, de éstos, dentro del nodo:

1. Vigilar el "Equilibrio del nodo", para obtener el Grado de Congestión (C) (Ec.4.6).
2. Monitorear el Retardo promedio de Transmisión del paquete en el Nodo (Rtn) (Fig.4.9), desde que el paquete llega a la capa de transporte (T_a) de un nodo, y hasta que la subcapa MAC, del mismo nodo, puede trasmitirlo (T_s):

$$Rtn = T_s - T_a \tag{4.4}$$

3. Medir la proporción media de paquetes perdidos por CAF (paquetes perdidos por falla de acceso al canal), en la subcapa MAC del nodo transmisor.

$$CAF = \frac{Paquetes\,perdidos\,por\,CAF}{Paquetes\,recibidos\,en\,MAC} \tag{4.5}$$

Para medir el *equilibrio del nodo* o congestión C, partimos de las ecuaciones 4.1 y 4.2, y del modelo de NODO propuesto. La idea central es conocer la tasa de paquetes que recibe el nodo (r_in) y la tasa de paquete que éste puede transmitir al canal inalámbrico (r_{MAC}); la congestión se mide como:

$$C(i) = \frac{r_{in}}{r_{out}} \tag{4.6}$$

Cuando $C(i) > 1$ el *nodo* i se declara en congestión, y el valor que resulta de esta división mide el grado de congestión o equilibrio del *nodo* i; ya que refleja la proporción de paquetes que puede transmitir del total que recibe.
r_{in} se mide a la entrada del Programador de Paquetes, en la capa de Transporte, y considera todos los tipos de paquetes sin importar su prioridad; con lo cual, podemos conocer la demanda de comunicación del nodo. r_{out} se mide en la subcapa MAC y refleja el grado de disponibilidad del enlace inalámbrico; ya que un canal saturado provoca que se transmitan menos paquetes, por el contrario, un canal libre permite un mayor número de transmisiones de paquetes. Por lo tanto, r_{out} refleja las condiciones

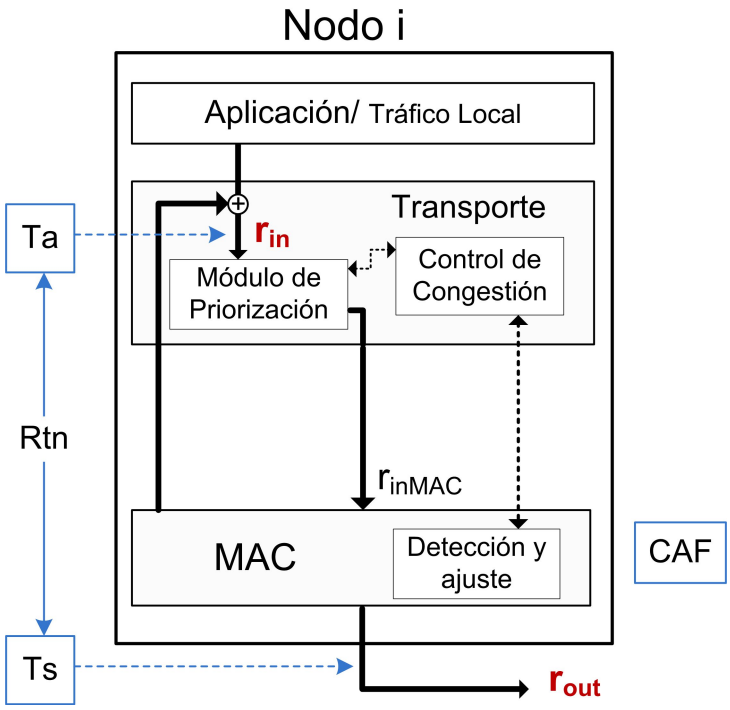

Figura 4.9: Modelo de Medición para la Detección de Congestión.

de tráfico en el entorno del nodo para una cierta cantidad de datos de entrada. La figura 4.9 muestra los puntos de medición de r_{in} y r_{out}.

Como se planteo al inicio de esta sección, la intención es proponer tres Estados de Congestión ($C(i) > 1$) dentro del nodo, que permitan detectar la congestión en sus inicios y su evolución hasta un nivel crítico. Para aplicar diferentes grados de control de congestión, según el estado en que se encuentre. Los tres estados de congestión se definen en base a los parámetros de carga de la red y de la QoS: Retardo Promedio de Transmisión de Paquete del nodo (Rtn) y a la proporción Promedio de Paquetes Perdidos por CAF. Aquí es, donde los eventos Rtn y CAF toman participación para establecer los límites de los tres Estados de Congestión. Ya que si bien, $C(i)$ mide el grado de congestión de la red (Ec.4.6), no tiene consciencia de las necesidades de QoS de las aplicaciones. Provocando que, el módulo de Resolución controle la congestión sin tomar en cuenta la QoS de las aplicaciones. Por ello, involucramos los parámetros Rtn y CAF, en el módulo de Detección, para que éstos sean la *base de referencia* de los

límites de cada Estado.

La propuesta es mantener el valor de *Rtn* y *CAF* por debajo de cierto umbral de QoS, para el tráfico con mayor prioridad (P_1), durante cualquier Estado de congestión. Este requerimiento, es pasado al módulo de Resolución para que haga lo necesario por mantenerlo. Para los tráficos tipo P_2 y P_3 dejamos que el protocolo *PCCQ* les ofrezca el *mejor esfuerzo*; esperando que los ajustes del módulo de Resolución, para cada Estado de congestión, les ayude a alcanzar un buen nivel de QoS. Entonces, $C(i)$ marca la congestión en base a la carga de la red y, *Rtn* y *CAF* informan cuando se alcanza los umbrales QoS predeterminados.

A continuación se presenta el Algoritmo 1 para el módulo de Detección, cuyo objetivo central es, identificar los tres Estados de Congestión en base a la carga de la red y vigilar el valor de las métricas QoS. El Algoritmo 1, que reside en cada nodo y es parte del Protocolo de Transporte *PCCQ*, para su operación recibe tres argumentos o medidas de los parámetros *C, Rtn* y *CAF*; entonces, evalúa a cuál de los tres Estados corresponde el grado de congestión, y cuál es el valor de los umbrales *Rtn* y *CAF*. Una vez que decide el estado, invoca el módulo de Resolución de Congestión, integrando como argumento el Estado correspondiente.

Este algoritmo se invoca cada vez que se realizan las mediciones de los parámetros mencionados, y no depende de otros nodos para ejecutar sus procesos. Además, permite transitar entre los tres Estados en cualquier momento, de acuerdo a los valores medidos. En este proceso, se propone una estrategia para fijar los valores de los umbrales, la cual consiste en:

> Cuando el nodo debe cambiar a un Estado de congestión más alto, debe ponerse los umbrales límites, impuestos por el QoS de las aplicaciones. Sin embargo, cuando el nodo debe pasar a un Estado menor, los valores de los umbrales deben ser menores a los límites requeridos, para manejar un margen de tolerancia, que asegure que realmente la congestión se redujo; evitando cambios innecesarios entre Estados de congestión (efecto *ping-pong*). Por ejemplo, los valores de los umbrales de las líneas 10, 14 y 16 del algoritmo, deben tener umbrales diferentes y más bajos que los umbrales de las líneas 2, 4 y 8.

Algoritmo 1: Algoritmo del Módulo de Detección.

Data: $C(i)$, $Rtn(i)$, $CAF(i)$

Result: Grado de Congestión:ESTADO ?

```
1  if (Estado Actual == 1) then
2      if ((Rtn ≥ U_Rtn) AND (CAF < U_CAF) AND (C < U_C)) then
3          Estado Actual =2; Módulo de Resolución ( 2 );
4      else if ((CAF ≥ U_CAF) OR (C ≥ U_C)) then
5          Estado Actual = 3; Módulo de Resolución ( 3 );
6      end
7  else if (Estado Actual == 2) then
8      if ((Rtn ≥ U_Rtn) OR (CAF ≥ U_CAF) OR (C ≥ U_C)) then
9          Estado Actual = 3; Módulo de Resolución ( 3 );
10     else if ((Rtn < U_Rtn) AND (CAF < U_CAF) AND (C < U_C)) then
11         Estado Actual = 1;Módulo de Resolución ( 1 );
12     end
13 else if (Estado Actual == 3) then
14     if ((Rtn < U_Rtn) AND (U_CAF1 ≤ CAF < U_CAF2) AND (U_C1 ≤ C < U_C2))
       then
15         Estado Actual = 2; Módulo de Resolución ( 2 );
16     else if ((Rtn < U_Rtn) AND (CAF < U_CAF1) AND (C < U_C1)) then
17         Estado Actual = 1; Módulo de Resolución ( 1 );
18     end
19 end
```

En la siguiente sección, se evalúa la operación de este algoritmo para conocer su desempeño; el cual, esta más ligado a los valores de los parámetros que a la ejecución misma del código. Además, se hace una propuesta de valores para los umbrales *C(i)*, *Rtn* y *CAF*, considerando lo anteriormente discutido.

Pruebas de Evaluación del Módulo de Detección

Con la intención de evaluar la operación del módulo de Detección y definir los límites de los tres Estados, se realizaron una serie de pruebas. Para éstas, se consideraron las

mismas configuraciones de red utilizada en la evaluación del módulo de Priorización (Sec.4.3.1). Sin embargo, aquí se midieron diferentes parámetros: el Retardo Promedio de Transmisión de Paquete del nodo (Rtn), la Proporción Promedio de Paquetes Perdidos por falla de acceso al canal (CAF) y el grado de Congestión del nodo (C); en los puntos que se marcan en la figura 4.9. El estándar IEEE 802.15.4 define los mecanismos para medir la cantidad de paquetes perdidos por CAF y lo registra en cada evento (Sec.2.5.4). Por medio del diseño Cross-Layer, el módulo de Detección de Congestión puede acceder a esta información y a la cantidad de paquetes recibidos en la MAC (r_{inMAC}); para aplicar la ecuación 4.5. Además, a cada evento en la simulación le asignamos un tiempo de aparición, con el cual se implementó la medición del retardo del paquete Rtn, mediante la ecuación 4.4. Estas mediciones se realizaron cada vez que expira un timer, el cual corresponde a la ventana de medición de cada parámetro. Para todas las mediciones, se obtuvo un promedio móvil ponderado exponencial (EWMA, Exponential Weighted Moving Average). El enfoque EWMA evita pronósticos equivocados, debido a cambios repentinos o abruptos de los valores medidos del experimento; a diferencia de cuando se usa la media normal [Wu et al., 2003]. Dado que en las redes inalámbricas puede haber varios factores que provoquen cambios súbitos, un filtro que suavice los datos nos permitirá hacer mejores decisiones sobre el control de congestión. La ecuación 4.7 fue utilizada para ejecutar las mediciones con EWMA.

$$S_{new} = (1 - \alpha)S_{old} + \alpha \left(\frac{N_T}{T} \right) \tag{4.7}$$

Donde α es una constante que toma valores entre $0 < \alpha < 1$, y representa el grado de ponderación que se le asigna a los valores nuevos (S_{new}) e históricos (S_{old}). A valores mayores de α, EWMA le asigna un menor peso a las observaciones antiguas; ya que $(1 - \alpha)$ tiende a cero (fijado a 0.1, por referencia de varios trabajos, en particular de [min LIN et al., 2011]). N_T denota el parámetro de medición (r_{in}, r_{out}, T_a, T_s, T_{inMAC} o CAF) en cada caso, que se mide cada intervalo de tiempo T (programado a 1 segundo para estas pruebas).

Para evaluar el módulo de Detección se realizaron dos casos de prueba:

(i) Evaluación libre: para medir los parámetros de C, Rtn y CAF contra la carga de la red. Con la intención de observar el comportamiento de sus valores, cuando la red

transita de una carga baja a una carga alta (en base a la capacidad máxima de IEEE 802.15.4 de 250 kbps, ver Sec.4.3.1). Este caso de prueba, nos permitió definir los umbrales de cada Estado de Congestión para C, y conocer el comportamiento de los valores Rtn y CAF durante los diferentes grados de congestión.

(ii) Prueba del algoritmo de Detección de Congestión: para validar si los umbrales definidos para C, en cada Estado (del caso anterior), corresponden con la carga de la red. Además, para evaluar si los valores propuestos para los umbrales QoS (Rtn y CAF), ejecutan una transición eficiente entre los Estados de Congestión, evitando cambios innecesarios (efecto *ping-pong*); especialmente cuando pasa a un Estado menor. Para ello, colocamos una variable de estados (*Estado Actual*) dentro del módulo de Detección; que indica en que Estado se encuentra el nodo durante la simulación, como lo muestra el algoritmo 1. Los valores propuestos para los umbrales Rtn y CAF pueden ser modificados, en base a las necesidades QoS del escenario de aplicación.

Es importante mencionar que, para estas pruebas, se configuró la red para que el incremento de tráfico fuera más lento y a intervalos más amplios, con respecto al caso del módulo de Priorización; con la intención de visualizar con más detalle los valores de los parámetros conforme el tráfico se incrementa.

La figura 4.10 y 4.11 muestran los resultados del caso 1 para los valores de C, Rtn y CAF del nodo 1. Observe de la figura 4.10, como el valor de retardo Rtn y congestión se incrementan conforme el trafico de la red lo hace. Por ejemplo, a los 200 seg de simulación (una carga de 150 p/s) el valor de congestión fue menor a 1.025 y el valor de Rtn menor a 7 ms. Sin embargo, cuando la red alcanza la carga máxima, los valores de congestión y Rtn crecen rápidamente debido a la saturación del canal que provoca un incremento del tiempo de espera para transmitir un paquete. Por su parte, la figura 4.11 también presenta el mismo comportamiento para la proporción de paquetes perdidos por CAF. Por ejemplo, hasta los 200 seg de simulación la pérdida de paquetes es baja, menor al 1.0%. Cuando la red alcanza una carga media, el valor de CAF es menor al 10%; después, alcanza el 20% cuando esta a máxima carga. En lo que respecta al valor de congestión, éste presenta el mismo comportamiento y va de 1.0 hasta 1.25, con 1.15 en la parte media de la simulación. De igual forma, cuando el canal tiene una alta demanda, el algoritmo CSMA-CA tiene mayor probabilidad de encontrar el

canal ocupado y fallar en su intento de transmisión, provocan una mayor cantidad de paquetes perdidos por CAF (Sec.2.5.4).

Estos resultados nos permiten concluir que, la forma de medir la congestión del nodo refleja fielmente la carga de tráfico en la red y el grado de congestión; sin necesidad de recibir información del resto de los nodos. Ya que los valores de *Rtn* y *CAF* se afectan conforme el valor de *C* se incrementa por la carga en la red.

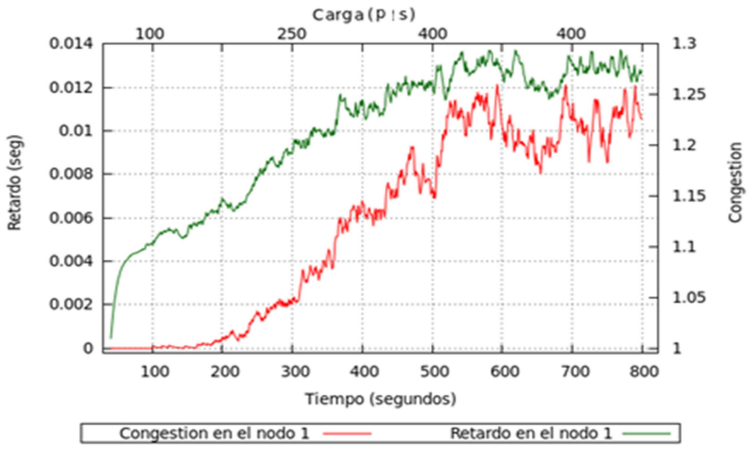

Figura 4.10: Retardo promedio de transmisión de paquete y Congestión del Nodo 1.

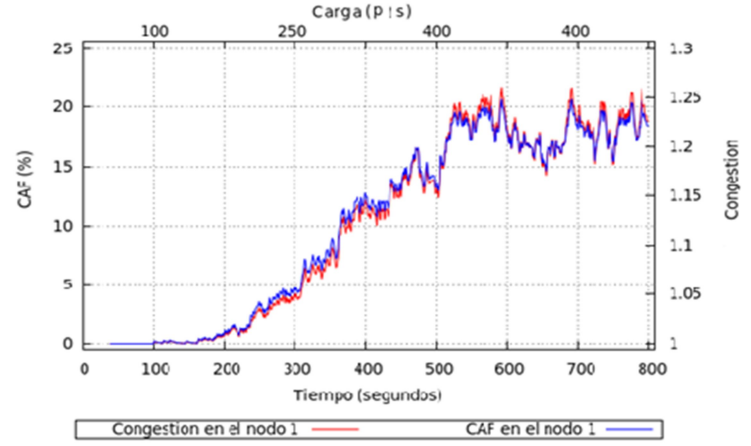

Figura 4.11: Proporción de Paquetes perdidos por CAF y Congestión del Nodo 1.

El segundo caso de prueba consistió en ejecutar el algoritmo 1, para diferentes valores del parámetros C en cada simulación; buscando sintonizar los umbrales de los tres Estados de congestión que *mejor se ajusten* a la carga de la red. Además, en esta misma prueba se evaluaron los valores de los umbrales Rtn y CAF, para ejecutar la transición de un Estado alto a un Estado bajo y viceversa. En las pruebas se fijo $Rtn = 22\ ms$ y $CAF = 10\%$, pudiendo ser cualquier otro requerimiento (marcado por las necesidades de QoS de la aplicación).

Recordemos que los Estados de congestión se dividieron en *carga baja (I)*, *carga media (II)* y *carga alta (III)*, en relación a la carga máxima que soporta la red de 390.62 p/s (Ec.4.3). Además, el algoritmo 1 nos indica en que Estado de congestión se encuentra el nodo, de acuerdo a los umbrales establecidos y a la carga presente en la red. Por lo tanto, de los resultados obtenidos en cada simulación, retroalimentamos el proceso de *sintonización* de los diferentes umbrales del algoritmo de Detección; hasta alcanzar el objetivo planteado en este caso de prueba. Después de extensivas simulaciones se presentan, en la figura 4.12, los resultados finales para cada umbral. Observe, como los umbrales de C, por si mismo, marcan los tres Estados de congestión; y como los umbrales Rtn y CAF para "subir" a otro Estado corresponde a los límites propuestos de QoS (Rtn=*22ms* y CAF=*10%*). La intención de los umbrales es permitir el cambio a un Estado que pueda mejorar el control de la congestión y mantener los umbrales QoS marcados. En este mismo sentido, para que un nodo pueda "bajar" a un Estado menor, los valores de los umbrales deben ser menores a los límites; para evitar un cambio inadecuado de estado. Por ejemplo, para pasar del Estado III al II el valor de CAF debe estar entre 5 y 10% y Rtn menor de 10 ms; y para pasar al Estado I, el valor de CAF debe ser menor al 5%. La figura 4.13, muestra los resultados que producen, estos umbrales, en la operación del algoritmo 1. El eje Y de lado izquierdo (Y1) de la figura 4.13 muestra el valor del Estado de congestión del nodo 1. El eje Y derecho (Y2) muestra la cantidad de paquetes que finalmente pudieron enviarse a la red, llamado tráfico enviado o entregado al canal de comunicaciones. El tráfico enviado es una proporción de la carga de la red, ya que la carga de la red considera el 100% de los paquetes que generan las aplicaciones de cada nodo (en esta prueba, va de 50 a 400 p/s), independientemente que se pierdan o se transmitan. Este cambio en la gráfica nos

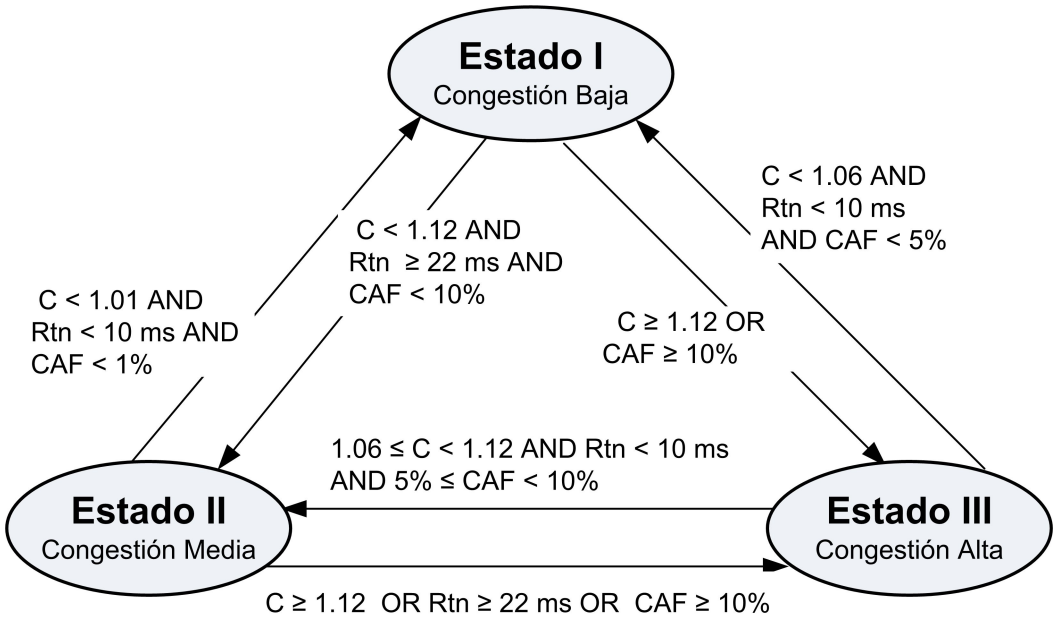

Figura 4.12: Umbrales para el Algoritmo del módulo de Detección.

permite observar como afecta la congestión a la eficiencia de la red; ya que muestra el tráfico real que se pudo transmitir del total de paquetes generados por las aplicaciones de los nodos, durante las diferentes etapas de congestión. Observe de la figura 4.13, cuando la red tiene una carga baja de tráfico (menor a 150 p/s) el nodo se mantienen en el Estado 1, como se esperaba. En este punto, el nodo logró transmitir el 100% de la carga ofrecida por todos los nodos en ese instante (150 p/s); esto puede verificarse observando la carga de la red en las figura 4.11 y figura 4.13. Después, se mantiene en el Estado 2 hasta 245 p/s. Al final, el Estado se mantiene en 3 cuando la carga de la red esta en su máximo valor. Se observa que a los 425 seg de simulación hay una transición entre el Estado 3 y 2; debido a que la carga de la red se redujo un instante por debajo de los 245 p/s. Pero en general, el efecto del problema de *ping-pong* por una mala elección de los umbrales no se presenta.

Al respecto del tráfico enviado por los nodos, conforme la congestión se hace más crítica, la proporción de paquetes que pueden ser colocados en el canal se reduce; ya que de los 400 p/s que se generan, sólo se alcanza un 62.5% (250 p/s). Evidenciando, los problemas de la congestión en el desempeño de la red y la necesidad de su control.

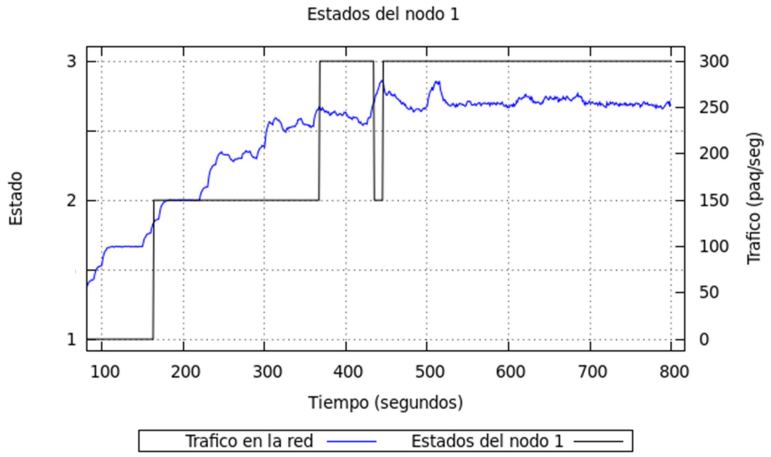

Figura 4.13: Estados del Nodo 1 y tráfico enviado a la red.

En conclusión se puede evidenciar en las pruebas de los dos casos, que el módulo de Detección de Congestión tiene un buen nivel de precisión; ya que los valores de C, Rtn y CAF estiman adecuadamente la carga presente en la red y evitan transiciones innecesarias de Estado, respectivamente. Estimación que se hace sin necesidad de interactuar con otros nodos y mucho menos con el nodo Sink; para agilizar la detección y realizarla de forma distribuida e independiente. Además, los límites establecidos para cada Estado de Congestión dentro del algoritmo 1, siguen correctamente los cambios de carga en la red; dividiéndola en congestión baja, media y alta. Lo cual nos permite diseñar nuestros módulos de Notificación y Resolución en base a estos tres Estados; evitando activarlos de forma anticipada y/o innecesaria, con algún cambio "transitorio" de congestión en la red.

4.3.3 Módulo de Notificación de congestión

Este módulo tiene su importancia en la medida que un nodo necesite del apoyo de los nodos en su entorno para controlar la congestión presente. En este sentido, un mensaje de notificación de congestión debe ser emitido por el nodo que la detecta, hacia los nodos de la red. Los nodos que reciban la notificación, deben actuar conforme el módulo de Resolución de congestión lo indique.

El módulo de *Notificación* propuesto, tiene dos principios de operación básicos: *(i)* Informar cuando el grado de congestión del nodo sea *crítico (Estado III)*. *(ii)* El modo de notificación es *implícito*, y se utilizará un cambio de bit para indicar presencia o ausencia de congestión crítica.

Una de las premisas del protocolo *PCCQ* es trabajar el control de la congestión independiente de otros nodos. Sin embargo, proponemos una excepción a esta premisa, cuando la congestión en la red no puede ser resuelta y alcanza el Estado III. Entonces, el nodo emite una notificación *Implícita* a sus nodos vecinos dentro de la red, para que éstos participen de forma activa en la resolución de la congestión. Este notificación deber ser eliminada, cuando el nodo sale del Estado III, con la intención de recuperar los niveles de operación del resto de los nodos de la red (como se verá en la Sec.4.3.4). Por lo tanto, el módulo de Notificación debe informar la presencia o ausencia del Estado III de congestión a los nodos de su entorno.

La notificación *Implícita*, segundo principio de operación de este módulo, es considerada una de las opciones más eficientes para la notificación de mensajes dentro de una WSN, en presencia de congestión; como se explicó en la sección 3.1 y 3.3. Ésta utiliza el concepto de "Piggyback" para incrustar la información necesaria, dentro de la carga útil del paquete de datos de la subcapa MAC (MSDU, Sec.2.5.3), para indicar la presencia o ausencia de congestión crítica. Esto es, aprovechar los paquetes de datos de la aplicación para transportar información de control por la red, como se muestra en la figura 4.14. Evitando la necesidad de crear un paquete de control exclusivo para este fin; ya que estos paquetes contribuyen al incremento de la congestión y al consumo de energía del nodo. Recordemos que, cuando existe congestión en la red, la idea es evitar que se genere más carga y sobre todo de paquetes de control. De ahí, la preferencia por este tipo de notificación. Además, se propone utilizar sólo un bit, para indicar presencia ("1") o ausencia de congestión crítica ("0"), llamado Notificación de Congestión (NC). Para afectar lo menos posible el espacio destinado a los datos de aplicación dentro del paquete IEEE 802.15.4; ya que el tamaño de MSDU esta limitado a 118 Bytes (Sec.2.5.3): $Carga\,útil = MSDU - Inf.Control$ (Fig.4.14).

Para hacer llegar la información a los nodos de la red, se aprovecha el canal común inalámbrico y la función de Recepción de datos de la subcapa MAC. En el primer caso,

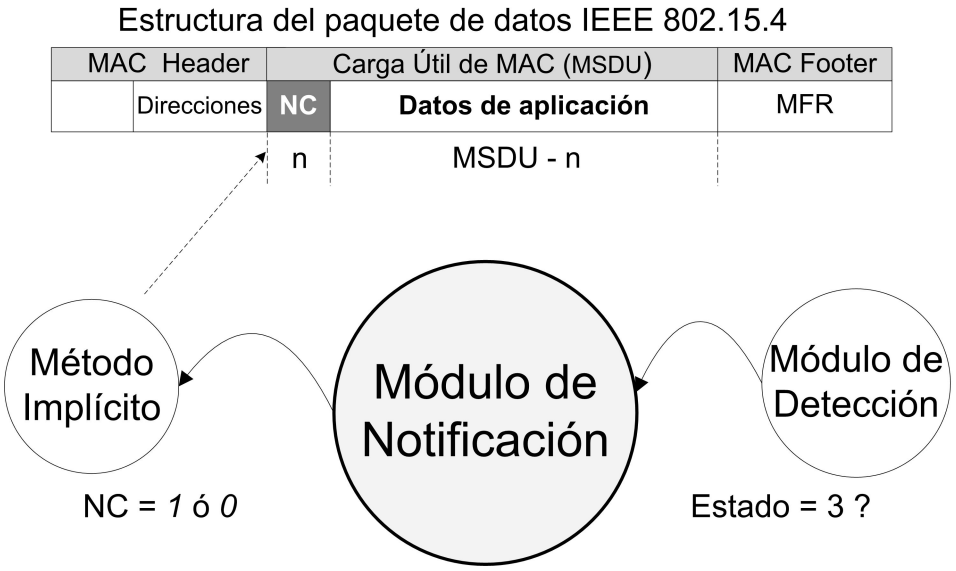

Figura 4.14: Módulo de Notificación y método Piggyback.

se aprovecha el medio de red compartido, para "escuchar" las transmisiones de los nodos vecinos dentro de su cobertura, independientemente del destino de la transmisión (Peje. todos los nodos en la cobertura del nodo A escuchan el envío de datos de A-B, Fig.4.15). Dicho de otra manera, los nodos en la red están obligados a recibir todos los paquetes que "escuchan", debido a la naturaleza de la radio comunicación. La MAC de cada nodo, en consecuencia, tiene que recibir estos paquete y aplicar una serie de *filtros*; para rechazarlos o aceptarlos (pasándolos a las capas superiores). Entre estos filtros, esta la verificación de las direcciones de origen y destino incluidas en el paquete (Fig. 4.14); para saber si el paquete recibido por el nodo es para él ([LAN/MAN Standard, 2006]). En este proceso es donde revisamos el campo de control *NC*, que viene en cada paquete, para saber su valor y almacenarlo en una variable; antes que el paquete sea desechado o enviado a otras capas del nodo. El protocolo *PCCQ*, que reside en capa de Transporte, accede al valor almacenado de NC vía la interacción Cross-Layer, empleada en nuestro diseño. Esta información es pasada al módulo de Resolución para que actué en consecuencia.

En conclusión, se logra un notificación efectiva y rápida sin necesidad de enviar paquetes especiales a cada nodo de la red. Además, el módulo de Notificación sólo tienen que

fijar el valor de NC, de acuerdo al Estado de congestión actual del nodo, e incluirlo en cada paquete de datos que se transmite; el resto del proceso es parte de la naturaleza del canal inalámbrico y de las operaciones incluidas en la subcapa MAC.

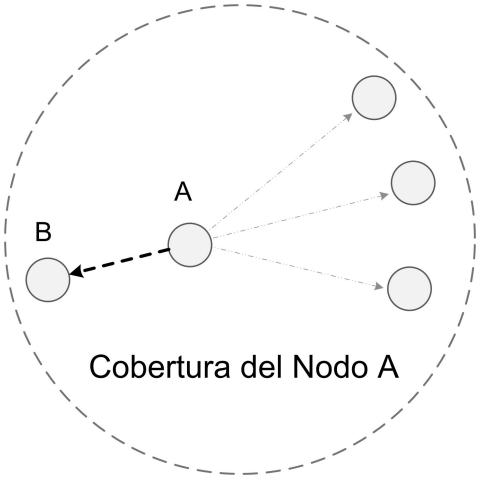

Figura 4.15: Medio común de comunicaciones en las WSNs.

4.3.4 Módulo de Resolución de congestión

La resolución de congestión debe ser dinámica y distribuida en cada nodo de la red; que tome en cuenta los requerimientos de la aplicación, el estado actual de congestión del nodo y el tráfico presente en la red. Se propone un módulo pro-activo que utiliza dos estrategias de operación, para el control de la congestión, (Fig.4.16):

1. Control de Flujo de datos. Auto-regulación del nodo y reducción diferencial de la tasa de transmisión.

2. Control del acceso al medio inalámbrico. Administración dinámica del algoritmo CSMA-CA.

Estas dos estrategias funcionan de acuerdo al Estado de congestión del nodo; y se activan de dos maneras: *(i)* de Forma Local, a través del módulo de Detección de cada nodo. *(ii)* de Forma Externa, cuando un nodo recibe un paquete y el valor de notificación de congestión Crítica es *"1"*. La figura 4.16 muestra el esquema del módulo

y sus estrategias de operación. El detalle de su operación se explican en las secciones siguientes.

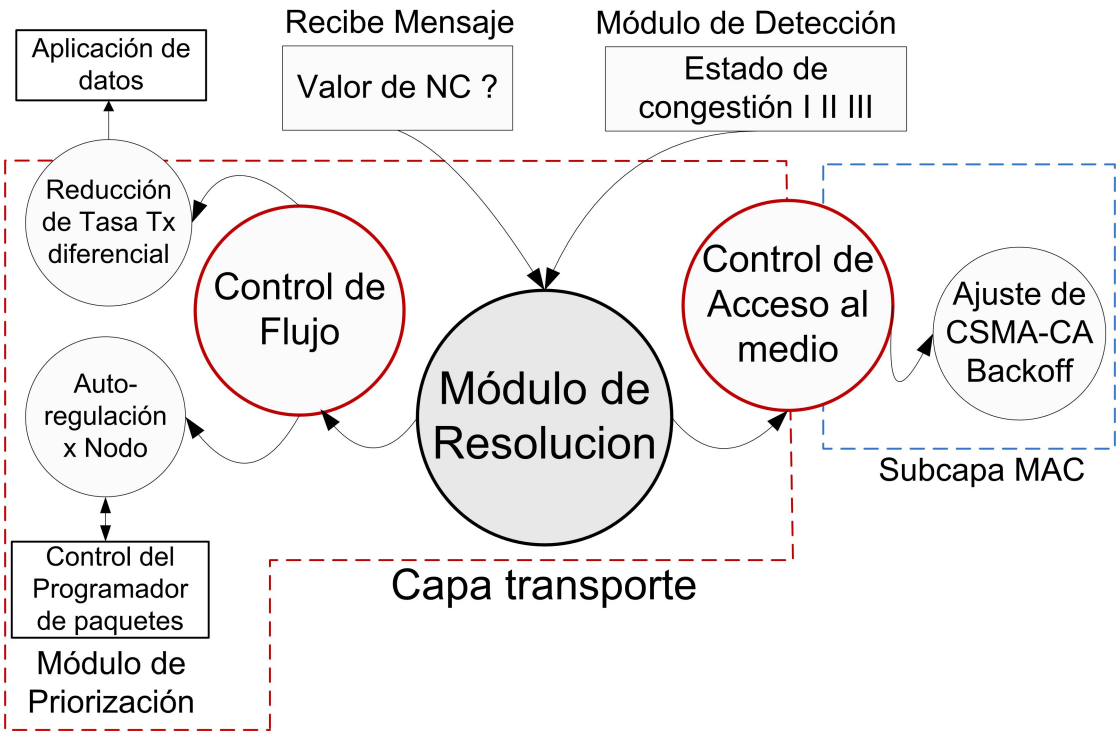

Figura 4.16: Módulo de Resolución de *PCCQ*.

Control de Flujo de Datos

La estrategia del Control de Flujo de datos tiene su base en el ajuste de la tasa de transmisión del nodo. Esto es, reducir o aumentar la cantidad de paquetes que un nodo puede transmitir hacia la red. El control de Flujo, considera la prioridad del tráfico para ajustar la tasa de datos, en relación a la QoS de cada tráfico y el grado de Congestión presente en el nodo. Este control tiene 2 modos de operación: *(i)* Auto-regulación del Nodo y *(ii)* Regulación de la tasa de datos de las aplicaciones.

La Auto-regulación del Nodo aprovecha la operación del módulo de Priorización, específicamente del Programador de Paquetes, para regular indirectamente la tasa de datos que un nodo puede generar (r_{pro}, Fig.4.5). Como se mostró en la sección 4.3.1, la cantidad y tipos de paquetes que el nodo puede transmitir, pueden controlarse por los

valores de ponderación del algoritmo WWRR del Programador de Paquetes (Q_1, Q_2 y Q_3, Fig.4.6); para usar el canal de comunicaciones de forma controlada. Entonces, la propuesta es modificar los valores de ponderación de acuerdo al grado de congestión y al tipo de tráfico presente en el nodo. De las pruebas realizadas en la evaluación del módulo de Priorización (Sec.4.3.1), proponemos un esquema de ponderación para cada uno de los Estados de congestión; como se muestra en la Tabla 4.1. Esta configuración permite reducir y/o seleccionar el tráfico de salida del Programador de Paquetes, en cada turno de servicio, conforme la congestión del nodo aumenta; además, cumple con la política de operación del módulo de Priorización: donde para cualquier valor de ponderación debe resultar que $Q_1 > Q_2 > Q_3$.

Tabla 4.1: Ponderación del Programador de Paquetes para cada Estado.

Congestión	$WWRR[Q_1]$	$WWRR[Q_2]$	$WWRR[Q_3]$
ESTADO I	$Q_1 = Peso$	$Q_2 = Peso*50\%$	$Q_3 = Peso*25\%$
ESTADO II	$Q_1 = Peso$	$Q_2 = Peso*25\%$	$Q_3 = Peso*25\%$
ESTADO III	$Q_1 = Peso$	$Q_2 = Peso*25\%$	$Q_3 = Peso*0\%$

De la Tabla 4.1 se pueden resaltar dos puntos importantes. El primero es, la ponderación para el trafico tipo P_1, el cual permanece sin cambios en cualquier Estado; buscando no afectar la tasa de envío de paquetes prioritarios, aún en presencia de congestión. El segundo es, la ponderación del tráfico tipo P_3, que es la que más se reduce; especialmente para el Estado III, donde se detiene la salida de estos paquetes. Lo anterior contribuye a reducir rápidamente la congestión crítica de la red; sin embargo, el nodo empieza a crear una cola de paquetes dentro del búfer que menos atención tiene (peje. Q_3). Entonces, cuando el búfer se satura, el nodo empieza a *descartar* o perder paquetes del búfer; lo cual a primera vista es un problema. Sin embargo, es preferible perder paquetes en este punto, que hacerlo dentro del canal inalámbrico. Por que un paquete que se transmite durante congestión crítica, tiene altas probabilidades de perderse por *colisión* de datos. Esta pérdida es más *costosa*, ya que consume energía inútilmente del nodo en su transmisión y contribuye a la congestión de la red, afectando la operación de otros nodos en su entorno. Por lo tanto, y con la intención de eliminar la congestión del nodo, es preferible perder paquetes del búfer que transmitirlos en

presencia de congestión crítica; y más, si son los paquetes menos importantes (tipo P_3). Aún con lo anteriormente descrito, proponemos una segunda linea de acción que es, el Ajuste de la Tasa de Transmisión del tráfico que generan las aplicaciones del nodo; cuando la congestión de éste, esta en el Estado III. Para lo cual, *PCCQ* notifica a su capa de aplicación que necesita reducir temporalmente su tasa de datos, para evitar saturar los búfers del módulo de Priorización. Esta reducción deber ser de acuerdo al QoS de las aplicaciones, por ello, proponemos reducir diferencialmente los trafico de acuerdo al esquema de la Tabla 4.2. Donde se observa que: sólo hay ajustes cuando el nodo detecta congestión crítica; el tráfico P_1 no sufre reducción de su tasa original (TO_{P1}); y el tráfico P_3 es el que más reducción presenta. Es importante mencionar que, estos valores propuestos pueden modificarse en cualquier momento por la capa de aplicación; ya que la responsabilidad de *PCCQ* es sólo detectar la presencia o ausencia de congestión crítica y notificar el resultado a la capa de Aplicación. Entonces, la capa de Aplicación puede hacer el ajuste como mejor le convenga al QoS de cada servicio. Sin embargo, dejamos estos valores con fines de evaluación del protocolo *PCCQ* y para ver los efecto al reducir la tasa de datos en presencia congestión.

Tabla 4.2: Reducción de la Tasas de Datos de las Aplicaciones del Nodo.

Congestión	$Tasa\ P_1[T_{P1}]$	$Tasa\ P_2[T_{P2}]$	$Tasa\ P_3[T_{P3}]$
ESTADO I	$----$	$----$	$----$
ESTADO II	$----$	$----$	$----$
ESTADO III	$T_{P1} = TO_{P1} * 100\%$	$T_{P2} = TO_{P2} * 75\%$	$T_{P3} = TO_{P3} * 50\%$

La activación de este ajuste de tasa es por dos vías, cuando el módulo de Detección informa que se encuentra en Estado III; o cuando el nodo recibe un paquete con la bandera de NC puesta a "1". La reducción de tasa se mantienen, mientras reciba paquetes marcados con $NC = 1$, independientemente del estado de congestión propio del nodo. Su desactivación tiene una regla que es, sólo se restablaceran los valores originales de tasa (TO) de cada tráfico, cuando el valor de NC sea igual a cero y el módulo de Detección observe un valor diferente del Estado III.

Las pruebas correspondientes a la estrategia de control de Flujo de Datos, no se presentan en esta sección por la amplitud de las mismas y por la necesidad del resto de

los módulos de *PCCQ*. Por lo cual, se dejan para el siguiente capitulo, donde se evalúa el protocolo completo con los 4 módulos de operación.

Control del Acceso al Medio

La segunda estrategia para el control de la congestión se da en la subcapa MAC del nodo, al administrar el algoritmo de acceso al medio CSMA-CA. Específicamente la administración de sus parámetros de operación, para incrementar o reducir las oportunidades de transmisión de datos de los nodos, de acuerdo al grado de congestión presente en la red. Para explicar esta estrategia, es necesario recordar la operación del algoritmo CSMA-CA, detallada en la sección 2.5.4. CSMA-CA *espera* un número aleatorio de períodos de tiempo (de 320 μseg cada uno), llamados períodos de Backoff, en el rango dado por la ecuación 4.8; para revisar si el canal de comunicaciones esta libre.

$$Periodos\,de\,Backoff = [0,\, 2^{BE} - 1] \tag{4.8}$$

El valor de *BE*, esta relacionado con el número de períodos de Backoff que el nodo puede seleccionar; y toma valores entre *macMinBE* y *macMaxBE* (por omisión en 3 y 5 respectivamente). De la siguiente manera, si el canal se encuentra ocupado, el nodo incrementa en uno el valor de *BE* (hasta *macMaxBE*), después repite los procesos de espera y revisión del canal de comunicaciones. Estos procesos los ejecuta hasta que cumple la cantidad de intentos permitidos, dado por el valor del parámetro *NB* o número de intentos de transmisión (macMaxCSMABackoff, por omisión en 4). Si no logra enviar el paquete, CSMA-CA lo "desecha" y reinicia los valores de *BE* y *NB* para el siguiente paquete (algoritmo de la Fig.2.22).
CSMA-CA, con estos procesos, intenta reducir la probabilidad de *colisiones* con otras transmisiones en curso. Además, considera que si el canal esta ocupado es un síntoma de congestión de la red; entonces incrementa exponencialmente el rango de periodos de Backoff del nodo (Ec.4.8). Con ello, reduce temporalmente la tasa de salida de paquetes del nodo, para no contribuir a la congestión de la red. Sin embargo, el algoritmo CSMA-CA tiene algunos inconvenientes de operación con estos procesos:

- Dos nodos pueden seleccionar el mismo número de periodos de Backoff, y por lo tanto, transmitir al mismo tiempo; provocando una colisión entre ellos. Esto

se debe, principalmente, a la configuración del valor inicial de *macMinBE* (por omisión en 3), que provoca un rango de períodos de Backoff muy corto. Esto se agrava, además, por que todos los nodos de la red usan el mismo valor inicial de *macMinBE* en cada paquete que necesitan transmitir. Entonces, es deseable que el rango inicial de períodos de Backoff sea lo mayor posible, y más cuando la red esta en congestión.

- Otro inconveniente de CSMA-CA es, la suposición de congestión de la red cuando el nodo detecta el canal ocupado. Ya que vasta que un sólo nodo este transmitiendo en toda la red, para que cualquier otro nodo detecte el canal ocupado. En consecuencia, todos los nodos incrementan exponencialmente su tiempo de espera (Ec.4.8), suponiendo erróneamente un canal congestionado. CSMA-CA, entonces, debería contar con más elementos de evaluación de congestión, para no incrementar el tiempo de espera tan drásticamente cuando no es necesario.

- Con la intención de evitar que un paquete se mantenga en espera de trasmisión por demasiado tiempo, CSMA-CA establece la variable *NB*; para desechar el paquete si alcanza el número máximo de intentos definidos. Este proceso, sin embargo, puede alcanzar su valor límite (por defecto 4) rápidamente, y más en una red inalámbrica; provocando una gran cantidad de paquetes perdidos por falla de acceso (CAF). Por lo tanto, es deseable que el nodo sea más tolerante y se mantenga por más tiempo intentando la transmisión de un paquete; si el retardo de transmisión requerido por la aplicación lo permite.

Por todo lo anterior, es claro que la operación y el tiempo de espera de CSMA-CA depende del valor de *BE* y *NB*; y estos valores dependen de cuantas veces se encuentra el canal ocupado, cuando el nodo intenta transmitir un paquete. Entonces, es necesario investigar el efecto del valor de *BE* y *NB* en el desempeño de CSMA-CA (del estándar IEEE 802.15.4). Además, es necesario considerar en esta evaluación los inconvenientes del algoritmo CSMA-CA, aquí expuestos; con la intención de mejorar nuestra propuesta de control de congestión. A continuación proponemos dos casos de estudio:

1. El impacto del valor de *BE*: para ello configuramos a *macMinBE* y *macMaxBE* con el mismo valor, con lo cual eliminamos la dependencia de la cantidad de fallos

de acceso al canal en su valores y reducimos las variables de diseño a un solo valor. Ya que el valor de *BE* no se incrementará, cada vez que encuentra el canal ocupado.

2. El impacto del valor de *NB*: en este caso, seleccionamos un valor fijo para *BE* y diferentes valores para *NB* para evaluar el desempeño de CSMA-CA.

Para los dos casos de estudio utilizamos la misma configuración de la red WSN, empleada en la evaluación del módulo de Priorización. Las métricas utilizadas en los casos de estudio son: *(i)* Retardo promedio de Transmisión del paquete en el Nodo (*Rtn*) (Ec.4.4); *(ii)* el porcentaje promedio de paquetes perdidos por falla de acceso al canal CAF (Ec.4.5).

Evaluación del impacto del valor BE (macMinBE=macMaxBE)

A continuación se muestra el impacto del incremento del valor de *BE* (permitidos por el estándar y que se muestran en la Tabla 2.3) sobre las métricas propuestas, conforme la carga de tráfico en la red se incrementa. En las pruebas, el parámetro *NB* se mantuvo en su valor por omisión (macMaxCSMABackoff = 4). La figura 4.17 muestra que, a carga bajas de tráfico, el incremento de *BE* puede reducir significativamente los paquetes perdidos para una carga de tráfico dada. Por ejemplo, a los 400 seg de simulación (o una carga de 315 p/s), el valor de $BE = 7$ presenta el mejor rendimiento con 10% de paquetes perdidos; mientras que, cuando $BE = 4$ su valor es más del doble. Sin embargo, conforme la carga en la red se incrementa, la diferencia se vuelve menos significativa. Por ejemplo, después de los 500 seg (o una carga de 400 p/s) el resultado de los diferentes valores de *BE* tiende alcanzar el mismo valor de CAF. Por tanto, en cargas altas de tráfico, aumentar el valor de *BE* ya no es relevante. Esto se explica, por que conforme la cantidad de nodos que necesitan transmitir en un determinado tiempo se incrementa, la competencia por el acceso al medio es mayor; superando el mecanismo de espera de CSMA-CA. Entonces, una alta demanda de acceso significa que el canal se encuentra la mayor parte del tiempo ocupado, provocando que se incremente el valor de paquetes perdidos por CAF en cada nodo. Por el contrario, en cargas bajas, aumentar el valor de *BE* extiende el tiempo de acceso al canal de cada nodo; que a

su vez, provoca que las transmisiones puedan ser repartidas en ese tiempo. Esto es, existen más espacios sin transmisiones durante la operación de la red, dando mayor probabilidad de transmisión a los nodos; y por tanto, que el valor de CAF sea más bajo.

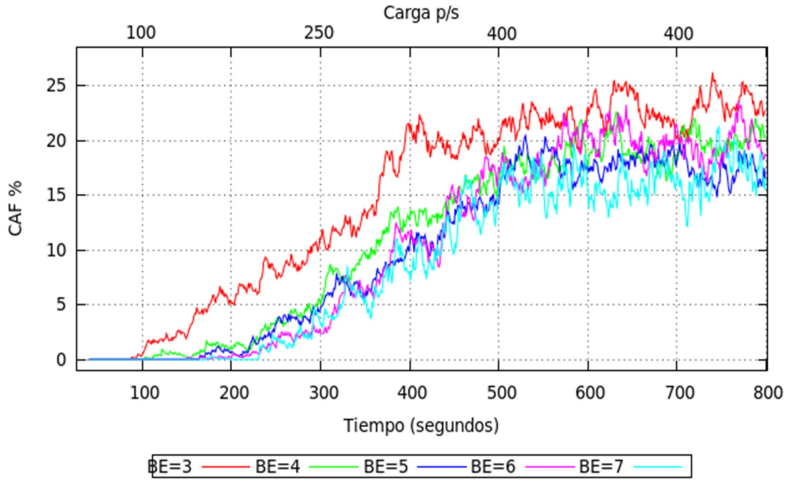

Figura 4.17: Paquetes perdidos por CAF Vs diferentes BEs.

La figura 4.18 muestra el retardo promedio de paquete para diferentes valores de *BE*. Observe que el valor de retardo es consistentemente mayor para los valores más altos de *BE*, a cualquier valor de carga de la red; con mayor impacto en cargas altas. Aún que esta figura muestra que los valores de retardo para *BE* de 3 a 5 son iguales, éstos no lo son. Para ello, se muestra la figura 4.19 con mayor detalle en la escala del eje Y; en ésta se aprecia que sigue dándose el mismo comportamiento, los peores resultados son para los valores más altos de *BE*. Lo que es determinante de estas dos figuras es, que existe una amplia diferencia entre el resultado del retardo de los casos con *BE* de 3 a 5, con respecto de lo alcanzando para los casos de *BE* con 6 y 7. Entonces, considerando el resultado de las figuras 4.17 y 4.18, existe una disyuntiva entre los resultados del retardo de transmisión y la tasa de paquetes perdidos. Por que los mejores resultados son para los valores de *BE* igual a 6 y 7; pero ésto, producen los peores resultados de retardo. Por ello, es necesario considerar los requerimientos QoS de las aplicaciones y el grado de congestión de la red, para seleccionar estos valores. Por ejemplo, puede utilizarse el valor de *BE* en 6, en cargas bajas (menor a 150 p/s), ya que alcanza buenos valores

de retardo (menor a 50 ms) mientras mantiene niveles muy bajos de paquetes perdidos (menor al 1%). En conclusión, encontramos que los valores de *BE* en 5 y 6 ofrecen el mejor equilibrio entre la cantidad de paquetes perdidos y el retardo de transmisión de paquete.

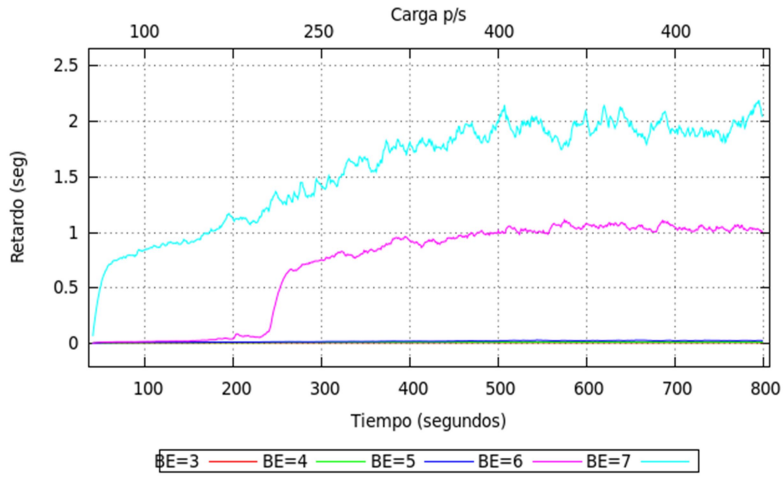

Figura 4.18: Retardo promedio de paquete para diferentes BEs.

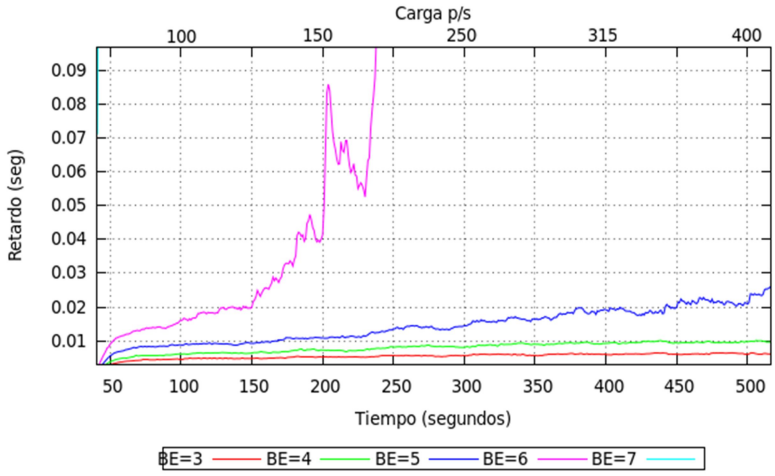

Figura 4.19: Detalle del Retardo promedio para diferentes BEs.

Evaluación del impacto del valor NB

Los resultados de la evaluación del impacto del valor de *NB*, en la operación del algoritmo CSMA-CA, se muestran en las figuras 4.20 y 4.21. En las pruebas se utilizó un valor fijo para $BE = 5$ y los valores de 2, 4, 5 y 7 para el límite de intentos de transmisión de un paquete o macMaxCSMABackoff. La etiqueta *maxBackoff* en las gráficas corresponde a la variable macMaxCSMABackoff, solo para efectos de visualización.
La figura 4.20 muestra el resultado del porcentaje de paquetes perdidos por CAF, para diferentes valores de macMaxCSMABackoff. Observe que, para el mismo valor de *BE= 5*, el resultado es mejor cuando el valor de macMaxCSMABackoff es mayor, para cualquier valor de carga de la red. Esto se explica por que permitir más intentos de acceso al canal para la transmisión de un paquete se reduce el número de fallas o paquetes perdidos. Se muestra en la figura, además, que en cargas bajas (menor a 150 p/s) el resultado de paquetes perdidos por CAF (cercano al 0%) alcanza los mismos valores para la configuración de macMaxCSMABackoff en 4, 5 y 7. Una importante conclusión a partir de esta gráfica es que, cuando se reduce la cantidad de paquetes perdidos por CAF, el nodo puede colocar más paquetes en la red; lo cual puede incrementar la probabilidad de colisión de paquetes en la red. Por ello, es necesario considerar el nivel de congestión presente en la red, antes de configurar los parámetros del algoritmo CSMA-CA.

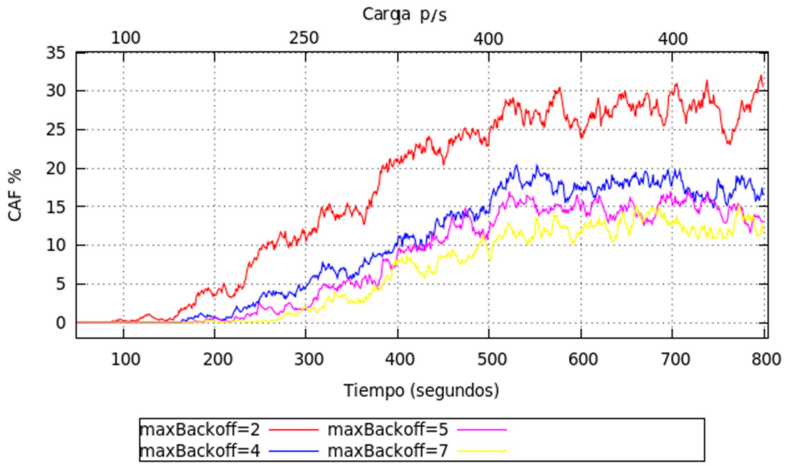

Figura 4.20: Paquetes perdidos por CAF Vs diferentes NBs.

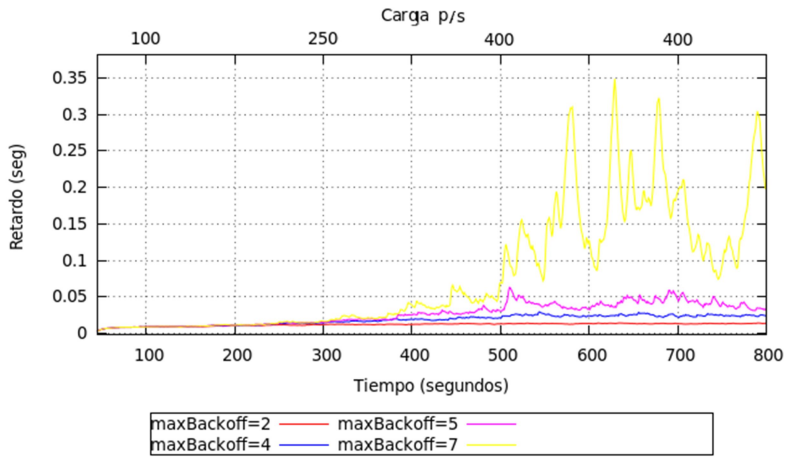

Figura 4.21: Retardo promedio de paquete para diferentes maxCSMABackoff.

La figura 4.21 muestra el valor de retardo promedio por paquete, que un nodo alcanza durante la simulación. Debido a que mayor número de intentos para transmitir un paquete se traduce en un mayor tiempo de espera. Entonces, el peor rendimiento se obtiene cuando el valor de macMaxCSMABackoff es alto (peje. en 7) y la red esta saturada; por ejemplo, después de 400 p/s el retardo alcanza valores hasta los 300 ms, para un macMaxCSMABackoff = 7. Esta situación se explica por que el nodo tiene menos oportunidades de acceder al canal y de enviar el paquete con una red congestionada. Sin embargo, cuando la red tiene una carga baja de tráfico (menor a 250 p/s), el retardo se mantiene por debajo de 15 ms para el mismo valor de macMaxCSMABackoff (Fig.4.22); lo cual es una gran diferencia contra los 300 ms del caso anterior. De la misma figura, se observa que los mejores resultados, durante toda la simulación, fueron para los valores de 2, 4 y 5; que además son similares, lo que nos permite una mayor posibilidad de elección.

En las pruebas presentadas para el caso del valor de *macMaxCSMABackoff*, se observa que los valores que ofrecen el mejor equilibrio entre paquetes perdidos y retardo de transmisión del nodo, dependen del nivel de congestión de la red. Ya que por ejemplo, a cargas bajas (menor a 250 p/s), el valor de $macMaxCSMABackoff = 7$ es el mejor; ya que alcanza un porcentaje de CAF menor al 1% y un retardo menor a 15 ms. Sin embargo, a cargas altas (mayor a 300 p/s) el mejor equilibrio lo alcanza el valor de *mac-*

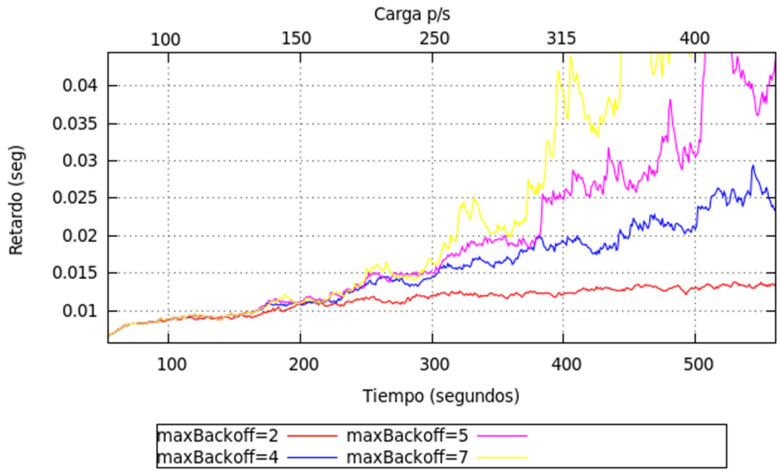

Figura 4.22: Retardo promedio de paquete para diferentes maxCSMABackoff.

MaxCSMABackoff = 4, que coincide con lo propuesto por el estándar IEEE 802.15.4.; alcanzando un porcentaje de CAF menor a 20% y un retardo menor a 30 ms.

Por lo tanto, proponemos utilizar un valor de 7 para *macMaxCSMABackoff*, mayor al permitido por el estándar (Sec.2.5.4), cuando la red tenga una carga de tráfico baja a media; con la intención de reducir la cantidad de paquetes perdidos. Además, cuando la red tenga una carga alta de tráfico, utilizar un valor de 4 o 5, dependiendo de las necesidades de retardo y paquetes perdidos establecidos. Aquí, vuelve a resaltar la importancia de identificar el grado de carga o congestión de la red y las necesidades QoS de las aplicaciones; trabajo que realiza el módulo de Detección.

En conclusión a las diferentes pruebas del desempeño del algoritmo CSMA-CA aquí presentadas, se observó la alta dependencia de los valores de *BE* y de *NB*, en conjunto con el grado de congestión presente en la red, en el desempeño las redes WSNs. Además, se observo que las configuraciones por defecto y la forma de administrar su valores, propuestas por el estándar IEEE 802.15.4, no son las más adecuadas; para el control de la congestión en escenarios de aplicaciones con diferentes tipos de tráfico.

4.3.5 Propuesta e Integración de Parámetros de los módulos de PCCQ

Los valores de los parámetros de los diferentes módulos expuestos en esta sección, se resumen en el esquema de la figura 4.23; dicho esquema contiene las configuración de los parámetros de cada módulo, que forman el protocolo *PCCQ*. Este esquema, propone un control de congestión basado en tres Estados de operación; los cuales son conscientes del grado de congestión de la red y las necesidades QoS de las aplicaciones. Como se muestra en la figura 4.23, se definieron umbrales para el módulo de detección, basados en la carga de la red, los valores de la tasa de paquetes perdidos por CAF y el retardo de transmisión del nodo (*Rtn*). Además, esta figura muestra los parámetros de configuración del módulo de Resolución, donde por un lado, se administra los valores de ponderación del módulo de Priorización y la notificación de congestión crítica. Y por otro lado, propone una forma diferente de administrar los valores de *BE* y de *NB* del algoritmo CSMA-CA, definido por el estándar IEEE 802.15.4. En este punto, es importante aclarar que no modificamos el estándar, solo administramos los valores de *BE* y de *NB* de una forma diferente, pero que es permitido por el estándar. Este cambio, en general, permite al mecanismo CSMA-CA contar con mayor información, además de sólo verificar el canal ocupado; para configurar de mejor manera estos valores.
En el estado I, se consideraron los valores de *BE* (macMinBE y macMaxBE) y *NB* (maxBackoff) más tolerante a la pérdida de paquetes. Aún que, estos valores son los que alcanzan el retardo más alto. Sin embargo, considerando que en este estado hay una carga baja de tráfico, no hay realmente un impacto en el retardo de transmisión de paquetes. Además, colocamos el umbral de *Rtn*, para cambiar a un mejor Estado, cuando se alcance el límite establecido. Por ello, el Estado II tiene valores que hacen más rápido el envió de paquetes (valor de *BE=5*), mientras mantenemos el valor tolerante de pérdida de paquetes (*NB=7*). Sin embargo, si la congestión sigue aumentando, es necesario transitar al Estado III, donde se tiene un esquema muy similar al estándar IEEE 802.15.4 (Tab.2.3), pero con un valor más tolerante a la pérdida de paquetes (*NB=5*). Esto, debido a que en las pruebas realizadas mostró el más bajo retardo de transmisión pero el peor resultado para la pérdida de paquetes. En el caso del módulo de priorización, observe que en los tres Estados, siempre se respeta que el tráfico más

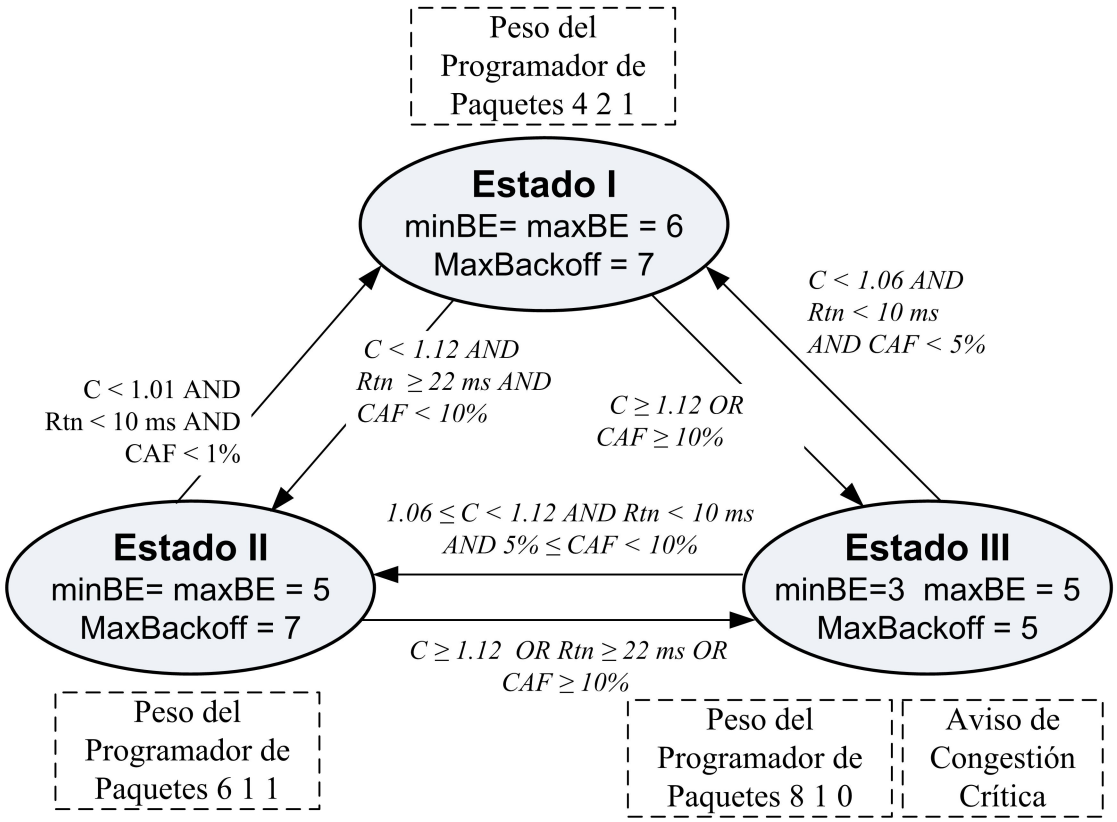

Figura 4.23: Esquema del Protocolo *PCCQ* y valores propuestos.

prioritario (P_1) tiene el mayor peso; y siempre se cumple que el peso de $Q_1 > Q_2 > Q_3$. Pero además, conforme la congestión en la red es más alta, le asignamos más peso a Q_1 y se reducen proporcionalmente Q_2 y Q_3; con lo cual garantizamos que los paquetes que sean más prioritarios tengan las mejores servicios de la red, y mediante el "sacrificio" de los paquetes menos prioritarios contribuir a eliminar o reducir la congestión de la red.

Recordando que *PCCQ* reside, principalmente en la capa de Transporte; a provechamos el diseño Cross-Layer para acceder a las subcapa MAC, para hacer la configuración de los parámetros de *BE* y *NB*, monitorear el valor de la variable de NC (Notificación de congestión Crítica) y obtener los valores necesarios para calcular *C*, *Rtn* y *CAF*. Sin necesidad de modificar el estándar IEEE 802.15.4. Además, dado que el módulo de Priorización esta en la capa de Transporte, éste esta conectado directamente al

módulo de Resolución; para administrar los valores de ponderación del Programador de Paquetes cuando cambia de Estado.
En conclusión, con este esquema podemos alcanzar un control de congestión dinámico, consiente del grado de congestión, tanto del nodo y como de la red, y de las necesidades de QoS de las aplicaciones. Que además, puede ser adaptado a otros escenarios de red fácilmente; ya que permite modificar el valor de sus parámetros, en cualquier momento, desde la capa de Transporte. El resultado integral de la operación de este esquema y en general de todos los módulos del protocolo *PCCQ* serán dados en el capítulo siguiente.

4.4 Conclusiones del capítulo

En este capítulo se presentó la propuesta del protocolo PCCQ, la cual consiste de un módulo de Priorización, de Detección, de Notificación y de Resolución; para resolver el problema del control de la congestión en redes WSNs. PCCQ se incluye en cada nodo de la red, con la intención operar de forma distribuida y directa donde se presente la congestión. La operación de PCCQ se basa en tomar en cuenta el grado de congestión del nodo y de la red; así como las necesidades de QoS de las aplicaciones. Para su diseño y construcción se utilizó el concepto de Cross-Layer, entre la capa de Transporte y la subcapa MAC; como se mostró en la arquitectura de diseño propuesta. Además, se presentaron los modelos de red y de nodo, propuestos para la operación de PCCQ.
Se realizaron diferentes pruebas para evaluar los mecanismo del protocolo PCCQ y conocer su operación en relación al control de la congestión. De estos resultados, se propusieron los valores de los umbrales de operación para cada parámetro de PCCQ.
Finalmente se concluye el capítulo, con un esquema de integración de los parámetros para cada estado de congestión; los cuales son la base de operación del protocolo PCCQ.

CAPÍTULO 5

EVALUACIÓN DEL PROTOCOLO PCCQ

Con la intención de evaluar la operación y el desempeño del protocolo *PCCQ*, en este capítulo presentamos una serie de pruebas, centradas en evaluar la capacidad del protocolo para el control de la congestión y el soporte de QoS en una red WSN. Estas pruebas consideran la evaluación integral de los cuatro módulos que forman el protocolo *PCCQ*, a diferencia del capítulo anterior que se evaluaron de forma independiente.
En general, se proponen dos casos de estudio: *(i)* la evaluación del desempeño de las métricas propuestas, con y sin el protocolo *PCCQ*; *(ii)* la comparación del protocolo *PCCQ* contra otros protocolos de Transporte, en el mismo escenario de operación.

5.1 Configuración de la Evaluación del protocolo PCCQ

A continuación se describe el escenario de red y la configuración de las pruebas, para la evaluación del protocolo *PCCQ*. El escenario consiste en la configuración de la red WSN, las características del tráfico y las métricas de evaluación. Además, las pruebas se agrupan en dos casos de estudio; con la intención de evaluar el protocolo *PCCQ* por si mismo y compararlo con otros protocolos de transporte.

5.1.1 Configuración de la WSN y características del tráfico

Para la configuración del escenario de red, el objetivo es construir un ambiente real de operación; que considere las características y necesidades principales de las aplicaciones que soportará la WSN. Principalmente, la cobertura de la red y el tipo de tráfico. En nuestro caso, la configuración en general, está enfocada a los sistemas de monitoreo continuo de pacientes de forma remota; sin embargo, los resultados no se limitan a este tipo de escenarios, ya que los requerimientos de este tipo de aplicaciones cubren una gran gama de otras aplicaciones.

La WSN esta formada por una topología de red que consiste de 21 nodo inalámbricos en la misma área de cobertura, como se muestra en la figura 5.1. Los nodos se desplegaron en una área de 35.0 x 35.0 metros (mts), con una separación promedio entre nodos de 8.5 mts. La intención es representar un espacio interior de una sala de hospital o una casa destinada al cuidado de personas. En la WSN existe un sólo nodo que recibe el trafico de toda la red, llamado nodo Sink; colocado al centro del área de cobertura. El resto de los nodos son considerados nodos sensores, los cuales pueden generar hasta tres tipos de tráfico de aplicaciones médicas. Estas aplicaciones corresponden a las señales del ECG (tráfico P_1), Ritmo Cardíaco (tráfico P_2) y Temperatura Corporal (tráfico P_3). Siendo P_1 y P_3 el tráfico de mayor y menor prioridad, respectivamente (Sec.4.2). Esta clasificación es acorde a las necesidades de comunicación de cada una de las señales (Tab.2.1) y a la importancia de la información que transporta. En general, nuestro escenario de prueba hereda todas las consideraciones del modelo de RED y de NODO, del modelo de sistema propuesto para el protocolo *PCCQ* (Secciones 4.2.1 y 4.2.2). La

Tabla 5.1 resume el resto de la configuración de la red y de los nodos utilizada en las pruebas.

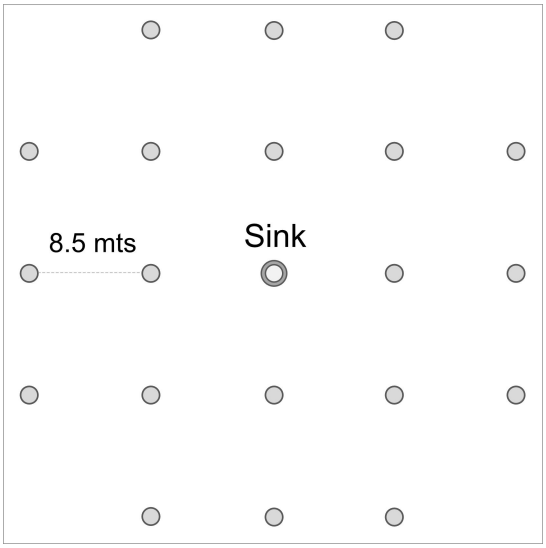

Figura 5.1: Topología de red de la WSN.

Tabla 5.1: Parámetros de configuración de la red y de los nodos de la WSN.

Parámetro	Valores
Agente de Aplicación	Tasa de Bit Constante (CBR)
Protocolos de Transporte	PCCQ, UDP, TCP-Tahoe, TCP-NewReno, TCP-Vegas
Protocolo MAC-PHY	IEEE 802.15.4 (en modo No-Baliza)
Agente de Ruteo	Estático
Tamaño de Búfer del nodo	21 paquetes (7 para c/u: Q_1, Q_2 y Q_3)
Tamaño de Paquete	80 Bytes
Tasa del canal inalámbrico	250 kbps (1 canal de Frecuencia a 2.47 MHz)

Para simular el tráfico de cada nodo sensor se utilizó un generador de tráfico en la capa de Aplicación del nodo; llamado CBR (de Constant Bit Rate). El CBR puede generar una tasa de paquetes constante, la cual puede cambiarse durante la simulación para cada nodo; de igual forma el tamaño del paquete de datos. El tamaño de paquete se definió

de 80 bytes, de acuerdo a los resultados presentados en el trabajo [Buenrostro et al., 2013]; donde se concluye que este tamaño ofrece el más bajo retardo de transmisión y una tasa de datos aceptable, cuando se utiliza el estándar IEEE 802.15.4. La aplicación CBR fue adaptada para agregar a cada paquete la etiqueta de prioridad y para recibir información de control de la capa de Transporte (como se vio en la Sec.4.3.4). Además, para las pruebas, se configuró el CBR para *crear* la misma cantidad de paquetes por cada tipo de tráfico, con una cierta tasa de salida. Esto es, un nodo inicia creando cuatro paquetes tipo P_3, seguido de cuatro tipo P_2 y finalmente cuatro tipo P_1, después repite el proceso hasta completar la tasa de paquetes por segundo (p/s) impuesta al nodo. Al respecto de la tasa de paquetes del CBR, cada nodo puede tener diferentes tasas de datos; por ejemplo, hay nodos que generan 15 p/s, y otros hasta 50 p/s. Los nodos con mayor tasa de datos simulan un paciente con los tres tipos de sensores o tipos de tráficos.

Es importante mencionar que la tasa de generación de paquetes del nodo, a nivel de capa de Aplicación, es llamada carga ofrecida a la red; que es distinta a la tasa de paquetes que un nodo puede transmitir al canal de comunicaciones, ya que esta última dependen del mecanismo de envío de la capa de Transporte y de la subcapa MAC (como explicó en la Sec.4.2.2).

La simulación se diseño para observar el desempeño del protocolo *PCCQ* en diferentes etapas de congestión, cargas de tráfico bajas, medias y altas. Por ello, la red inicia con un sólo nodo generando 50 p/s durante un tiempo de 50 segundos. Después, se incrementa gradualmente la cantidad de *nodos activos* (que generan tráfico) hasta 20; los cuales alcanzan los 400 p/s (Fig.5.2). Con lo anterior, se rebasa la capacidad de transmisión teórica de la red de 390.62 p/s, de acuerdo a lo establecido en la ecuación 4.3; colocando la red en máxima congestión. Después, aplicamos la operación inversa, para reducir gradualmente la carga de la red hasta llegar a 100 p/s (generados por 4 nodos). Este proceso crea una forma de "campana", como se muestra en la figura 5.2; lo cual nos permite contar con las tres etapas de congestión en la red, para evaluar a *PCCQ*.

La simulaciones fueron realizadas utilizando el programa "Network Simulator 2" [Institute, 2008], el cual recibe un archivo de entrada con todas las configuraciones de red,

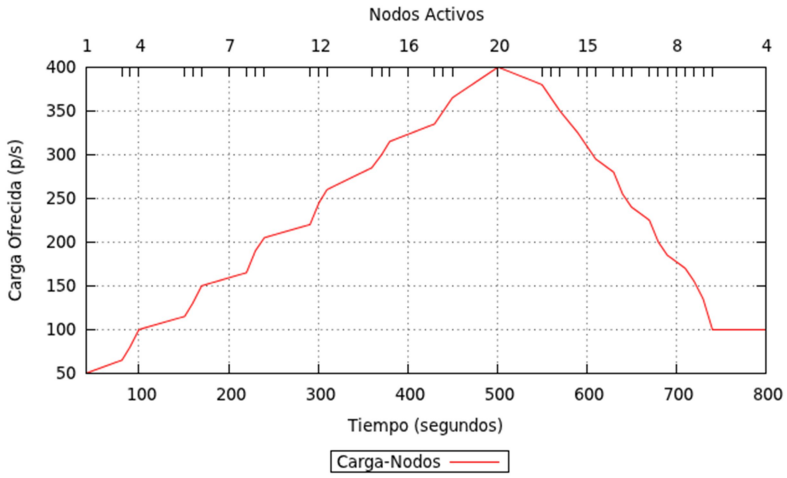

Figura 5.2: Carga de tráfico de red, durante las pruebas.

antes mencionadas, y con la operación de la red aquí descrita (Anexo A). Además, tiene integrado en sus librerías el código CBR, que genera los diferentes tráficos propuestos; los diferentes protocolos utilizados en la experimentación (como el IEEE 802.15.4, UDP, TCP, algoritmo de Ruteo, entre otros) y el código fuente de nuestro protocolo *PCCQ*. NS2 es herramienta orientada a eventos, que cuenta con módulos para el registro detallado de la operaciones creadas en el diseño de la red; en adición a las que implementamos en el protocolo *PCCQ*. Esto nos permite tener un seguimiento puntual del flujo que sigue un paquete dentro de los nodos y del canal de comunicaciones.

5.1.2 Métricas de Evaluación

Para medir el desempeño del protocolo *PCCQ*, dentro de la WSN, realizamos una serie de pruebas centradas en las métricas QoS, propuestas en la sección 2.1.2 y 2.2.2; y que aquí se retoman:

1. *Retardo promedio de Transmisión de Paquete (RTP)*: *RTP* es definida como el intervalo de tiempo, desde el instante que un paquete sale de la capa de Transporte de un nodo sensor y hasta que éste llega a la capa de Transporte del nodo receptor (Sink). *RTP*, entonces, considera el retardo de envío extremo a extremo que incluye: los posibles retardos de "encolamiento' de los búfers de almacenamiento,

el retardo de acceso al medio del mecanismo CSMA-CA, el retardo de transmisión por el canal hacia el destino y el tiempo que le toma al nodo Sink subir el paquete a la capa de Transporte. *RTP* es medido en el nodo Sink.

$$RTP = (Tiempo\,de\,Arribo - Tiempo\,de\,Salida) = (seg) \tag{5.1}$$

2. *Porcentaje promedio de Paquetes Perdidos (PPP)*: *PPP* mide el porcentaje promedio de paquetes perdidos por la red en cada ventana de medición. *PPP* considera, en su medición, los tres tipos de paquetes perdidos: por desbordamiento del búfer, por falla de acceso al medio (CAF) y por colisión de datos. Se obtiene dividiendo la cantidad de paquetes perdidos entre el total de paquetes generados en capa de aplicación de cada nodo en la red.

$$PPP = \left(\frac{Paquetes\,perdidos}{Paquetes\,generados}\right) * 100 = (\%) \tag{5.2}$$

3. *Desempeño de la red o Throughput (TH)*: *TH* mide la cantidad de paquetes de datos que la red puede entregar al nodo Sink. En detalle, se mide la cantidad promedio de paquetes de datos que recibe la capa de Transporte del nodo Sink, en la ventana de medición.

$$TH = \frac{Paquetes\,recibidos}{Ventana\,de\,medición} = (p/s) \tag{5.3}$$

RTP, *TH* y *PPP* son medidos para cada tipo de trafico de forma independiente, gracias a la etiqueta que cada paquete lleva; con la intención de saber el nivel de QoS que alcanza cada tráfico dentro de la red.

Estas métricas pueden reflejar claramente el nivel de operación del protocolo *PCCQ*; ya que un protocolo de Transporte debe ofrecer una transmisión de datos eficiente, con el menor número de paquetes perdidos, cumpliendo los tiempos límites de entrega y enviando el mayor número de paquetes por segundo al nodo destino. Además, los niveles que la red puede alcanzar, en términos de estas métricas, impactan el nivel de servicio que reciben las aplicaciones dentro de la WSN. Por ello, con estas métricas, podemos evaluar tanto el protocolo *PCCQ* como el nivel de QoS que la WSN puede ofrecer.

5.1.3 Casos de Estudio

Con la intención de conocer el desempeño de *PCCQ* sobre las métricas planteadas, se proponen dos casos de estudio para las diferentes pruebas realizadas:

1. *Evaluar el efecto de la congestión en las métricas QoS*, comparando los resultados alcanzados *con y sin* el control de congestión de la red. Esto es, realizar las pruebas con el protocolo *PCCQ* y sin él.

2. *Comparación del desempeño de PCCQ*, conocer el desempeño de *PCCQ* contra otros protocolos de transporte, bajo el mismo escenario de red.

Para ambos casos de estudio, se utilizaron las consideraciones anteriormente expuestas para la configuración de la WSN, las características del tráfico, la forma en que se presenta la carga de tráfico en la red y las métricas de evaluación. Además, para *PCCQ*, se utilizó el esquema de la figura 4.23 para ambos casos de estudio; éste muestra los valores de los umbrales de los diferentes parámetros del protocolo *PCCQ* (*C*, *Rtn*, *CAF*, ponderación para los búfer, *macMinBE*, *macMaxBE* y *macMaxCSMABackoff*). Como se obtuvieron estos valores y su importancia dentro de *PCCQ* fueron explicados en la Sec.4.3. Vale la pena recordar que estos umbrales sirven para *detectar* y *controlar* los diferentes grados de congestión presentes en la red. Además, los parámetros *Rtn* y *CAF*consideran las necesidades de QoS del tráfico de mayor prioridad (como el ECG) dentro del protocolo *PCCQ*; por lo cual, estos parámetros pueden ser configurados con diferentes valores de retardo y de paquetes perdidos.
Los resultados que se presentan, para cada caso de estudio, son obtenidos del promedio de 100 simulaciones diferentes; para lo cual, se configuró el simulador NS2 con un factor de aleatoriedad. Esto provoca que el simulador obtenga comportamientos no-determinísticos de la operación de la red en cada simulación; especialmente, en el funcionamiento del algoritmo CSMA-CA (para seleccionar los periodos de backoff) y el comportamiento del canal inalámbrico.

Configuración del primer caso de estudio

Para el primer caso de estudio, el objetivo fue evaluar el escenario de red con y sin el protocolo *PCCQ*. Para ello, para el caso "sin", usamos las mismas configuraciones de

la red, del tráfico y de la carga ofrecida por cada nodo; sin embargo, modificamos lo siguiente, en cada nodo:

- El protocolo UDP, en lugar de *PCCQ*, para la capa de Transporte; el cual no ofrece ningún control de congestión [Institute, 1980].
- Un sistema tradicional de almacenamiento FIFO; el cual no pondera ni clasifica el tráfico que recibe, como lo hace *PCCQ*.
- El estándar IEEE 802.15.4 con las configuraciones por *omisión* de los parámetros del algoritmo CSMA-CA, como la Tabla 2.3.

Esta configuración crea una red WSN sin control de congestión, ni priorización de tráfico; lo que permite visualizar el efecto del protocolo *PCCQ* en el control de la congestión de la WSN propuesta.

Configuración del segundo caso de estudio

Para el segundo caso de estudio, la intención fue comparar los niveles de desempeño del protocolo *PCCQ*. Para lo cual, una forma es compararlo con otros protocolos de transporte. Este objetivo no es una tarea fácil, ya que si bien, existe en la actualidad diferentes protocolos de transporte propuestos para redes WSN. Estos protocolos son probados en diferentes ambientes experimentales y con diferentes objetivos de estudio; por lo tanto, es difícil usar sus resultados para un punto de comparación. Además, no se cuenta con los códigos de programación para replicarlos y adecuarlos a la mismos objetivos de estudio. Sin embargo, existen protocolos de transporte como el TCP y sus variantes, que son ampliamente utilizados en las redes de comunicaciones. TCP, como se explicó en la sección 2.3.2, es un protocolo que tiene funciones de control de congestión que busca evitarla, y si no es posible ejecuta acciones para su control. Una de las ventajas de TCP es su diseño de aplicación general a cualquier tipo de red; por lo cual puede ser utilizado en las redes WSN. Además, el simulador de redes NS2 tiene los códigos de programación de este protocolo, lo cual permite cualquier modificación en ellos. En nuestro caso, se modificó el código para agregar puntos de medición dentro del protocolo TCP; tales como el retardo de transmisión de paquetes, los paquetes perdidos

por desbordamiento de búfer y CAF, la cantidad de paquetes transmitidos y recibidos por los nodos. Esto nos permitió usar el mismo escenario de red para poder hacer una comparación bajo los mismos términos y situaciones de carga de tráfico.

Por estas consideraciones se propone el uso del protocolo TCP para nuestro segundo caso de estudio. Seleccionamos tres versiones de TCP: TCP-Tahoe, TCP-NewReno y TCP-Vegas. TCP-Tahoe es la versión original de TCP y cumple con lo establecido en la sección 2.3.2 y en la figura 2.12. El resto de las versiones son modificaciones que intentan mejorar los mecanismos de detección y resolución de la congestión. Por ejemplo, TCP-NewReno puede detectar más rápidamente la congestión en la red y su mecanismo de resolución no es tan agresivo [Floyd et al., 2004]. Ya que, cuando se detecta congestión, TCP-Tahoe pasa a la fase slow-start, que reinicia el envió de datos a 1 paquete, como se aprecia en la figura 2.12; mientras que TCP-NewReno pasa a la fase de prevención de congestión, que reduce el envió de paquetes al valor del umbral de congestión, que es mayor a 1 paquete como lo muestra la misma figura. Esto permite que la red se recupere más rápido de un período de congestión. Por su parte, TCP-Vegas, si bien modifica algunos aspectos de los algoritmos anteriores, el aspecto más relevante es la forma como actúa contra la congestión [Brakmo et al., 1994]. La estrategia de TCP-Vegas es calcular la cantidad de tráfico que puede inyectar al enlace sin causar congestión. Para ello mide la tasa de datos actual del enlace, y lo compara con la tasa de datos que el enlace debería soportar. La estrategia le permite ajustar el tamaño de la ventana de transmisión y controlar el flujo de datos que puede enviar un nodo; evitando así, saturar los búfers de los nodos intermedios a lo largo del camino. Por ello, TCP-Vegas se considera como un protocolo *proactivo* que busca evitar la congestión; a diferencia de las versiones anteriores que necesitan esperar a que se pierda un paquete para activar sus mecanismos.

En suma las variantes de TCP difieren en los mecanismos que utilizan para el control de congestión, pero en general, están diseñado para ofrecer una comunicación confiable extremo a extremo para la entrega de datos.

Entonces, la configuración del segundo caso de pruebas consiste en utilizar el escenario de red y la carga de tráfico, anteriormente descritos, con cada uno de los protocolos TCP y con el protocolo *PCCQ*. Al igual que en el primer caso de estudio, para los

diferentes protocolos TCP, se utiliza un sistema de almacenamiento FIFO y el estándar IEEE 802.15.4 con su configuración por defecto.

5.2 Resultados de las Pruebas del Primer Caso de Estudio

En esta sección se presentan y discuten los resultados obtenidos en las métricas definidas para este caso de estudio, bajo el escenario planteado.

Retardo de transmisión promedio extremo a extremo (*RTP*)

La figura 5.3 muestra el retardo promedio de transmisión por paquete (*RTP*), para el tráfico tipo 1; *con* y *sin* el protocolo *PCCQ*. Además se incluyen, en el eje X1 el tiempo de simulación y en el eje X2 el valor de carga de tráfico de la red. Esta medición fue realizada a nivel de nodo, entre el Nodo 1 y el Sink, con la intención de observar con más detalle la operación del protocolo *PCCQ*, especialmente del módulo de Priorización y del control dinámico del acceso al medio. En general se observa que el valor de *RTP* es mayor cuando se utiliza *PCCQ*, que cuando no se utiliza; principalmente debido a la configuración del umbral *Rtn* a 22 ms (Retardo de salida del paquete dentro del nodo), en los parámetros del módulo de Detección (Fig.4.23). Esto hace que *PCCQ* permita valores de retardo de hasta 22 ms para que un paquete salga de un nodo; cumpliendo la propuesta de QoS para los tráficos tipo 1.
Observe en la figura 5.3 el efecto de *PCCQ* y este umbral en el Nodo 1. Cuando inicia el Nodo 1 a enviar paquetes (el único en toda la red que lo hace), alcanza un valor de *RTP* de al rededor de 15 ms, menor al umbral establecido. Por lo cual, *PCCQ* asume baja congestión y mantiene al Nodo en el Estado I. Conforme el trafico en la red se incrementa (más nodos transmiten), el tiempo que le toma al nodo 1 enviar sus paquetes es mayor; ya que necesita competir por el acceso al medio con más nodos. *PCCQ* permite este incremento hasta que se se alcanza el umbral *Rtn*; por ejemplo, a los 152 seg de simulación el valor de *RTP* es de 25 ms, 3 ms más de lo marcado por el umbral. Aquí vale la pena una precisión, el valor de *RTP* es medido en el nodo destino

Sink, en modo extremo a extremo, y el valor *Rtn* es medido dentro del Nodo. Por ello, siempre se cumple que $RTP > Rtn$, por que se adiciona el tiempo de transmisión del paquete por el canal, más el tiempo que le toma la nodo Sink llevar el paquete de su capa Física a su capa de Transporte. Entonces, si queremos obtener un menor tiempo de *RTP* es necesario reducir el valor del umbral *Rtn*.

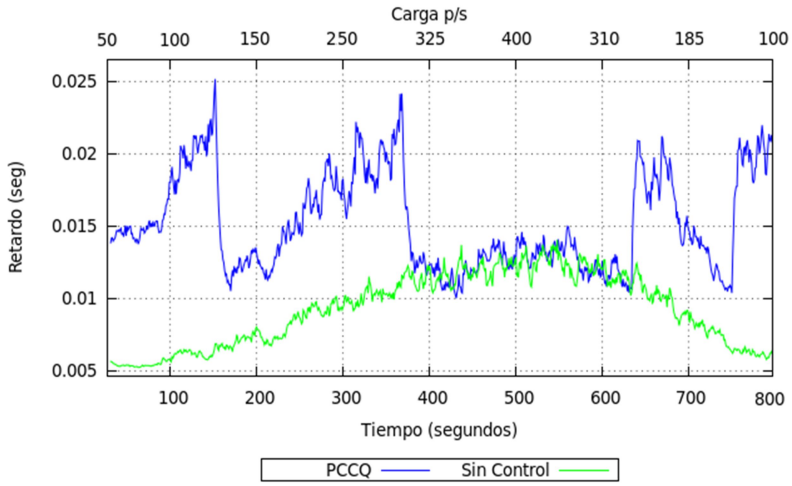

Figura 5.3: Comparación del Retardo Extremo a Extremo, con y sin *PCCQ*.

Cuando el Nodo 1 alcanza el umbral *Rtn*, *PCCQ* asume que el nivel de congestión en la red se incrementó, lo cual es correcto ya que la red paso de 50 a 115 p/s (generada por 5 nodos) a los 152 seg de simulación. Por lo cual, *PPCQ* hace un ajuste en su control de congestión y pasa al Nodo al Estado II (Fig.4.23); modificando los parámetros del módulo de Priorización y del algoritmo CSMA-CA. Con lo primero, damos mayor prioridad a los traficos tipo 1 y con CSMA-CA agilizamos el acceso al medio para reducir el tiempo de transmisión de los paquetes. Esto se aprecia en la figura 5.3, ya que a los 162.5 seg se simulación el valor de *RTP* baja a 10.5 ms. El nodo 1 se mantiene en este estado de control de congestión mientras no se alcance el umbral *Rtn*; lo cual se da a los 360 seg de simulación, con un valor de *RTP* de 24.15 ms, debido a que la carga de la de la red se incremento de 115 a 285 p/s (generados por 14 nodos). Entonces, *PCCQ* cambia el control de congestión del Nodo 1 al Estado III o congestión crítica, priorizando aún más los paquetes tipo 1 y deteniendo la salida de los paquetes tipo 3; mientras ajusta los valores del algoritmo CSMA-CA para reducir el tiempo de salida

de los paquetes (Fig.4.23). Este ajuste se aprecia a los 382 seg de simulación, cuando el valor de *RTP* baja a 12.35 ms.

En la parte de máxima congestión de la red, al rededor de los 400 p/s (20 nodos) o 500 seg de simulación, *PCCQ* alcanza un valor de *RTP* por debajo de los 15 ms (menor al umbral de *Rtn*), similar al valor alcanzado por la configuración de los nodos para las pruebas "Sin Control"; con las ventajas de la consciencia del valor límite de retardo del tráfico más crítico, la reducción de paquetes perdidos y mayor cantidad de paquetes enviados al Sink, como se vera en esta sección.

Siguiendo con la figura 5.3, conforme la red empieza a reducir la carga de tráfico, después de los 500 seg, *PCCQ* detecta que el valor de *Rtn* esta por debajo del umbral inferior (fijado a 10 ms, Fig.4.23); por lo cual, *PCCQ* puede relaja las restricciones del control de congestión, y pasa el Nodo 1 al Estado II; donde permite seguir enviado paquetes tipo 3. Este cambio se aprecia con el incremento del valor de *RTP* en la figura 5.3 (a los 634 seg de simulación). Este mismo efecto se presenta a los 750 seg de simulación, dado que la carga sigue reduciéndose, al pasar el Nodo 1 del Estado II al I. Cabe resaltar que, en ambos cambios, el valor de *RTP* nuca fue mayor a 22 ms.

Al respecto del trato ofrecido por *PCCQ* a los tráficos tipo 2 y 3, éstos obtienen los beneficios del control de congestión conforme la carga en la red se incrementa; sin embargo no hay una vigilancia sobre ellos. Las pruebas evidenciaron que en cargas bajas a medias tienen buenos resultados en los valores alcanzados de *RTP*, como se aprecia en la figura 5.4. Sin embargo, en congestión crítica o Estado III, *PCCQ* necesita controlar el tráfico que se envía a la red; por lo cual, detiene el envío de paquetes del tráfico menos importante o tipo 3. Por ello, en el tramo de 360 a 670 seg de simulación (Fig.5.4) no se observa un valor de *RTP* para este tipo de tráfico.

En conclusión, para esta métrica, observamos que *PCCQ* mantiene un buen control de congestión en relación a la carga presente en la red, mientras mantiene consciencia del valor límite requerido para el retardo extremos a extremo de los paquetes tipo 1. Ya que *PCCQ*, cambia entre los diferentes Estados de congestión según la carga de tráfico; sin presentar el efecto *ping-pong*. En estas pruebas, se concluye que *PCCQ* puede ser utilizado para garantizar un cierto valor límite a los tráficos de mayor prioridad. Además, se observa que el mayor aporte del valor final de *RTP* esta dado por el tiempo

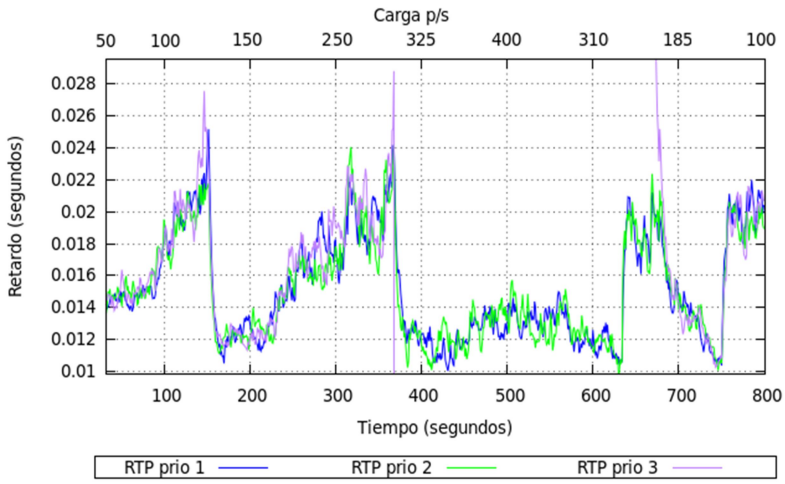

Figura 5.4: Retardo Extremo a Extremo para cada tipo de tráfico, con *PCCQ*.

que le toma al paquete salir del nodo; y este tiempo es en mayor parte debido al algoritmo de acceso al medio CSMA-CA y sus parámetros de configuración *macMinBE*, *macMacBE* y *macMaxCSMABackoff*.

Porcentaje promedio de paquetes perdidos (*PPP*)

Los resultado obtenidos para esta métrica se muestran en la figura 5.5; en ésta, se muestra el porcentaje promedio de paquetes perdidos por todos los nodos en la red, para el tráfico tipo 1 (eje Y); tanto para el protocolo *PCCQ* como para la configuración "Sin Control". El valor que se muestra en esta figura es la suma de los paquetes perdidos por desbordamiento de búfer, por falla de acceso al medio (CAF) y por colisión de datos. Los resultados, durante toda la simulación, muestran que el desempeño de *PCCQ* es mejor que la configuración *Sin Control*; especialmente cuando la red presenta una congestión crítica. Por ejemplo, a los 500 seg de simulación, la red tiene una carga de 400 p/s (o 20 nodos) que rebasan el límite teórico de la WSN de 390.62 p/s (Ec.4.3). En este punto, *PCCQ* alcanza un valor de *PPP* de 30%; mientras que *Sin Control* cree hasta un 40%. Esta diferencia en la mejora obedece a que *PCCQ* cuenta con el módulo de Detección y de Resolución; que le permite saber el grado de congestión presente en la red y activar los mecanismos de control de acuerdo al nivel de congestión, para

administrar la cantidad y tipos de paquetes que un nodo puede transmitir.

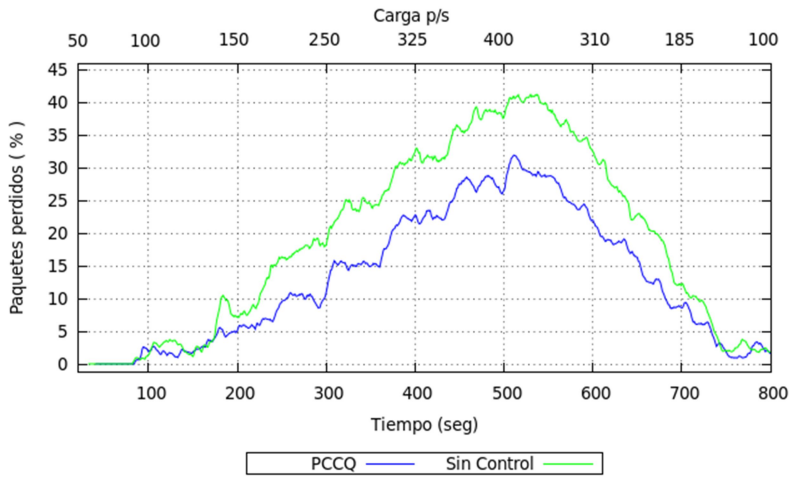

Figura 5.5: Comparación del Porcentaje de paquetes perdidos, con y sin *PCCQ*.

Al respecto del valor de *PPP*, es importante notar que esta ligado al nivel de carga de la red; ya que más nodos intentando enviar paquetes simultáneamente incrementan la probabilidad de pérdida de éstos. Por ejemplo, cuando la red tiene un solo nodo transmitiendo (hasta los 85 seg de simulación), el valor de *PPP* es 0%, para ambos casos. Sin embargo, a los 300 seg (o 250 p/s, de 12 nodos) *PCCQ* alcanza un máximo de 10%, mientras que *Sin Control* tiene 18%. Evidenciando con estos resultados la dependencia entre *PPP* y carga. Pero también se demuestra el desempeño del protocolo *PCCQ* en el control de la congestión a diferentes niveles de carga. Ya que *PCCQ* aplica diferentes acciones de resolución en cada nodo, de acuerdo al grado de congestión (Estados I, II y III, Sec.4.3.4). Entonces, si las necesidades sobre esta métrica es mantener un cierto valor límite, por ejemplo menor al 30%; es necesario reducir la cantidad de nodos en la red y/o ajustar los parámetros del módulo de Resolución de *PCCQ* (Sec.4.3.4). Recuerde que *PCCQ* esta diseñado para no enviar paquetes tipo 3, cuando el nodo entra en Estado III; sin embargo, puede hacerse lo mismo con los paquetes tipo 2, haciendo que la carga de la red se reduzca. Además, cuando un nodo esta en Estado III, éste emite un mensaje de notificación crítica (NC = 1); con lo cual, los nodos tiene que reducir un 25% y un 50% los tráficos tipo 2 y 3 a nivel de aplicación, respectivamente. Por lo tanto, se podrían incrementar estos porcentajes con el fin de lograr una menor

tasa de paquetes perdidos.

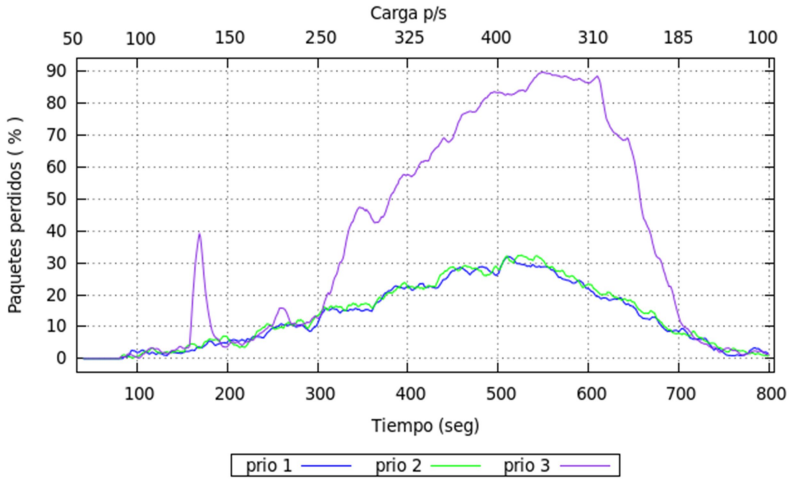

Figura 5.6: Comparación del Porcentaje de paquetes perdidos, con y sin *PCCQ*.

La figura 5.6 muestra el efecto del control de congestión de *PCCQ* para los diferentes tipos de tráficos, con los diferentes parámetros de configuración de la figura 4.23. Observe que, como estaba previsto, *PCCQ* ofrece los mejores niveles de *PPP* a los tráficos tipo 1 y 2, cuando la carga de la red esta en congestión media o alta. Por ejemplo, el trafico tipo 3 alcanza los 90% de paquetes perdidos contra un 30% para los tipo 1 y 2. Este comportamiento, como ya se mencionó, es el resultado del control de congestión de *PCCQ*, que deja de enviar paquetes tipo 3 durante congestión crítica. Esto a primera vista es drástico, sin embargo, se intenta mantener la operación de la red para los tráficos más importantes. Además, como se constato en las pruebas, la mayor cantidad de paquetes perdidos tipo 3 se dan por desbordamiento de búfer; los cuales son los menos graves. Esto es, un paquete perdido por colisión tiene un gasto mayor de energía del nodo (la cual es limitada); ya que el nodo necesita usar su módulo de transmisión, que es la etapa que más energía consume. Además, transmitir un paquete en la red, cuando ésta se encuentra saturada, no es lo más adecuado; ya que contribuye al grado de congestión y al aumentando de la probabilidad de colisione, afectando a otros nodos dentro de la red. Entonces, es preferible perder un paquete a nivel de búfer que a nivel de la red, y más si son los paquetes menos importantes. Por el contrario, las pruebas para la configuración "Sin Control", no tienen paquetes perdidos por des-

bordamiento de búfer, el 100% son paquetes perdidos por falla de acceso (CAF) y por colisión de datos. Esto por que el protocolo UDP y la configuración por omisión del estándar IEEE 802.15.4, seleccionados para la opción "Sin Control", tienen el objetivo de enviar los paquetes con la mayor rapidez posible (vea la Fig.5.3); lo cual provoca mayor probabilidad de paquetes perdidos (Fig.5.5).
En conclusión, *PCCQ* obtiene mejores resultados de *PPP* que la configuración "Sin Control". Además, nos permite modificar su operación a través de los diferentes parámetros de configuración; ya sea para obtener un menor retardo de transmisión o reducir la tasa de paquetes perdidos en la red.

Desempeño o Throughput de la red (*TH*)

A continuación se analizan los resultados obtenidos en la métrica TH para el protocolo *PCCQ* y para la configuración "Sin Control".
La figura 5.7 muestra la cantidad promedio de paquetes por segundo (*TH*) que recibe el nodo Sink de todos los nodos de la red, sólo del tráfico tipo 1 de cada nodo. Esto último es importante recalcarlo, ya que por configuración de la capa de aplicación (Sec.5.1.1), si un nodo genera 16 paquetes por segundo, significa que el nodo *crea* 4 paquetes de cada tipo por segundo; entonces, los valores de la figura 5.7 corresponden sólo a una cuarta parte del tráfico total. En esta figura se observa que se obtienen los mismos valores de *TH*, tanto para *PCCQ* como para "Sin Control", cuando la red presenta una carga baja de tráfico. Esto corresponde a los extremos de la figura (160 y 730 seg), ya que en estos puntos la red tiene a penas 5 nodos generando tráfico; aquí ambas configuraciones de prueba alcanzan un *TH* de 32 p/s. Conforme la red empieza a recibir más tráfico el desempeño de *PCCQ* presenta los mejores resultados en comparación con la configuración "Sin Control". Por ejemplo, cuando la red tiene congestión crítica (en el rango de 360 a 630 seg) o carga máxima, *PCCQ* esta por encima de los 80 p/s de *TH*; mientras que la configuración "Sin Control" no alcanza los 60 p/s. Esto representa una mejora de más del 25%, en la cantidad total de paquetes tipo 1 que recibe el nodo Sink de la red, durante la mayor parte del tiempo de simulación. Esta mejora se explica, debido a que *PCCQ* activa los mecanismos de reducción para los tráficos tipo 2 y 3, conforme a lo establecido para el Estado III (figura 4.23). Esto permite reducir la carga

de la red a costa de los tráficos menos importantes; y ofrecer una mayor oportunidad de transmisión a los paquetes tipo 1 de cada nodo (como se verá en la figura 5.8). Este tipo de ajustes y consciencia del QoS de cada tipo de tráfico no se obtienen con el protocolo UDP ni la operación típica del IEEE 802.15.4, que corresponden a la configuración "Sin Control".

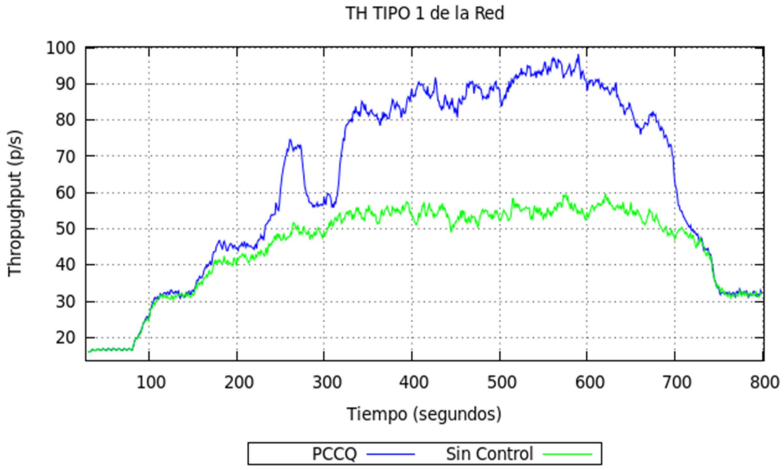

Figura 5.7: Comparación del Throughput de la red, con y sin *PCCQ*.

En la misma figura 5.7, se observa un pico del valor de *TH* de 74.2 p/s a los 250 seg, cuando la red esta en carga media (o Estado II). Este comportamiento obedece a un ajuste de control de congestión de Estado III, activado por la llegada de un paquete con una notificación de *congestión crítica* (NC) de algún nodo de la red. Por lo cual, *PCCQ* actúa enviando los nodos al Estado III mientras se sigan recibiendo este tipo de mensajes; como de hecho sucede hasta los 280 seg, cuando el valor de *TH* baja de nuevo. Este efecto se aprecia mejor en la figura 5.8, la cual muestra el valor de *TH* alcanzado por cada tipo de tráfico al utilizar el protocolo *PCCQ*. Observe que a los 250 seg el tráfico tipo 3 baja a 33.5 p/s, el tipo dos se mantiene 56 p/s y el tipo 1 sube 74.2 p/s; cumpliendo uno de los objetivos del protocolo *PCCQ*. Después, cuando los nodos dejan de recibir la notificación de *congestión crítica*, *PCCQ* regresa el control de congestión al Estado II, donde se ofrece un control más equilibrado para los tres tipos de paquetes, como se ve a los 300 seg de simulación. A diferencia de la configuración "Sin Control", que no hace una priorización del tipo de tráfico, ni ajustes en la transmisión

de paquetes para controlar la congestión. De hecho, las pruebas realizadas para este caso, muestran que los tres tipo de tráficos presentan el mismo valor de *TH* durante toda la simulación (igual al resultado de la figura 5.7, para el caso Sin Control).

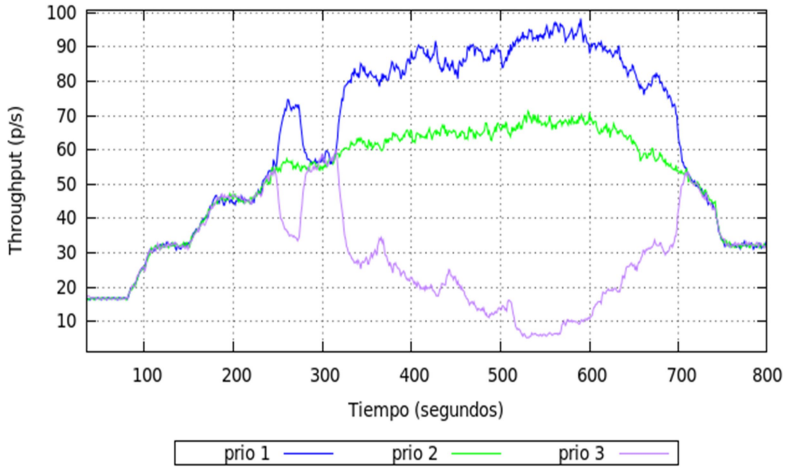

Figura 5.8: Throughput de la red para los diferentes tráficos con *PCCQ*.

En general las figuras 5.8 y 5.7 muestran que, mientras la red se encuentre con una carga de tráfico de baja a media, *PCCQ* ofrece un desempeño similar para los tres tipos de tráfico. Sin embargo, cuando la red presenta una carga de tráfico alta, *PCCQ* prioriza la atención que le ofrece a los tráficos y consigue ofrecer los mejores rendimientos al tráfico de mayor prioridad (tipo 1). A diferencia de cuando la red utiliza la configuración "Sin Control", que no prioriza ni controla la congestión de la red; ya que sólo alcanza un desempeño similar a *PCCQ* en cargas bajas, cuando la red tiene la capacidad de atender la carga de la red.

En conclusión a este caso de estudio, se observa que *PCCQ* obtiene mejores resultados en la comunicación de datos sobre el escenario de red planteado para la WSN; con respecto a la configuración "Sin Control". Ya que puede entregar más paquetes por segundo (*TH*) al nodo Sink, con un porcentaje menor de paquetes perdidos (*PPP*); mientras mantiene un retardo de transmisión de paquete extremo a extremo aceptable, cercano a los 22 ms marcados como límite QoS para el tráfico tipo 1. Esto se traduce en un mejor servicio de monitorización de señales médicas de los pacientes de forma remota; ya que ofrece los mejores niveles de QoS a las aplicaciones más criticas como la

señal ECG (tipo 1) y el ritmo cardíaco (tipo 2). Demostrando que *PCCQ* administra de mejor manera la comunicación de datos en la WSN, sobre todo en situaciones de alta demanda de comunicación; ya que *PCCQ* considera el grado de congestión presente en la red y las necesidades QoS del tráfico, para ajustar dinámicamente su control de congestión.

5.3 Resultados de las Pruebas del Segundo Caso de Estudio

Debido a que en el caso de estudio anterior se utilizó un protocolo de transporte que no tiene control de congestión, como lo es el protocolo UDP; ahora se evaluará el desempeño de *PCCQ* con protocolos de red que consideren esta función de control. Como se mencionó en la descripción del segundo caso de estudio (Sec.5.1.3) se utilizaron los protocolos de capa de Trasporte *TCP-Tahoe*, *TCP-NewReno* y *TCP-Vegas*, los cuales son ampliamente utilizados en las redes de comunicaciones y pueden ser adaptados a cualquier tipo de red; como fue el caso del protocolo de red IEEE 802.15.4. Entonces los siguientes resultados corresponden a la combinación de un protocolo TCP más el estándar IEEE 802.15.4 con su configuración por defecto. En esto último, es importante mencionar que se habilitó la función de recuperación de paquetes a nivel de la subcapa MAC (retransmisión de datos, Sec.2.5.2), debido a que los protocolos TCP lo necesitaron para establecer su sesión de datos; puesto que si los paquetes de inicio de sesión se pierden, el nodo no inicia el envío de datos y nuestro escenario de red no puede ser evaluado.

Retardo de transmisión promedio extremo a extremo (*RTP*)

La figura 5.9 muestra el resultado promedio de 100 simulaciones para el retardo de transmisión de paquete (tipo 1) extremo a extremo (*RTP*), para los tres protocolos TCP y para *PCCQ*. En esta figura se observa que los peores resultados se obtienen para los protocolos *TCP-Tahoe* y *TCP-NewReno* durante toda la simulación; ya que sus valores de *RTP* estan por encima de los 52.3 ms En segundo lugar lo ocupa el pro-

tocolo *TCP-Vegas*, con valores de *RTP* menores a los 29 ms Los mejores resultados los obtiene el protocolo *PCCQ*, ya que sus valores de *RTP* se mantienen por debajo de los 22 ms durante los diferentes grados de congestión de la red. Los resultados de *PCCQ* son mejores en un 58.65% con respecto de *TCP-Tahoe* y *TCP-NewReno*, y un 24.14% para *TCP-Vegas*. Sólo *TCP-Vegas* es mejor hasta los 80 seg de simulación, cuando la red tiene una carga baja (menor a los 65 p/s con 2 nodos en la red). Una causa de estas diferencias es que *PCCQ* cuenta con el módulo de Priorización de paquetes y el control de congestión, que agiliza la salida de los paquetes tipo 1 y mantiene consciencia del valor de QoS requerido por este tipos de paquetes. Además, otra causa, es la necesidad de los protocolos TCPs de *esperar* por un paquete de *confirmación* en cada paquete que envían, aumentando el valor de retardo de transmisión. Por lo cual, *PCCQ* no implementa esta función.

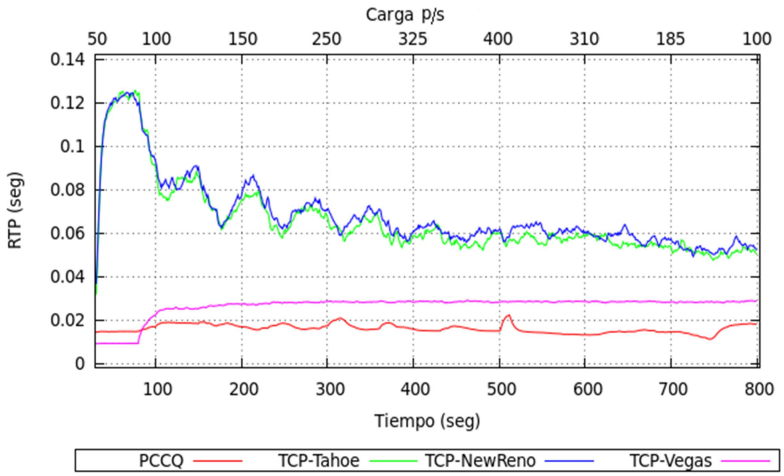

Figura 5.9: Comparación del Retardo Extremo a Extremo.

El valor de *RTP*, para los protocolos *TCP-Tahoe* y *TCP-NewReno*, presenta un pico importante a los 80 seg, esto se explica por la forma en que los protocolos envían sus paquetes (Sec.2.3.2, Fig.2.12). Ya que TCP inicia con el envío de un sólo paquete en la primer venta de transmisión, valor que va incrementando exponencialmente mientras no se detecte una falla en la entrega. Entonces, al inicio de la simulación TCP pasa rápidamente del envío de un paquete a una cantidad mayor, puesto que sólo existe un nodo transmitiendo (a los 30 seg). En consecuencia, cuando hay más paquetes in-

tentando salir del nodo, el tiempo que le toma salir a los paquetes tipo 1 es mayor. Después, cuando se activa el segundo nodo (a los 80 seg), existen problemas de colisión de datos; lo cual hace que los nodos pierdan sus paquetes, obligando a los protocolos TCPs a reducir la ventana de transmisión a un paquete. Por ello, se aprecia en la imagen, para estos protocolos, una señal con forma de "dientes de sierra". Esta explicación es importante mantenerla en consideración, ya que impacta el desempeño de las tres métricas evaluadas en esta sección.

En conclusión a los resultados de esta métrica de evaluación, podemos decir que *PCCQ* mantiene el mejor desempeño y una estabilidad en sus valores alcanzados durante las diferentes pruebas de simulación; aún con niveles críticos de congestión en la red. Entonces, cualquier aplicación que tenga necesidades QoS, sobre el retardo de transmisión extremo a extremo, menores a 22 ms por paquete, puede utilizar el protocolo *PCCQ*.

Porcentaje promedio de paquetes perdidos (*PPP*)

La figura 5.10 muestra los resultados para la métrica *PPP* del tráfico tipo 1, para los diferentes protocolos TCPs y *PCCQ*. Estos resultados son un indicador del grado de eficiencia de la red en la transmisión de paquetes. De entrada, los resultados muestran que *PCCQ* es mejor que el protocolo *TCP-Vegas* en cualquier nivel de congestión de la red. Para el caso de *TCP-Tahoe* y *TCP-NewReno*, *PCCQ* es mejor que éstos cuando la red presenta cargas bajas y medias de tráfico, y sólo es superado en cargas altas. Por ejemplo, a los 200 seg de simulación cuando la red tiene una carga de 150 p/s (o 7 nodos), *PCCQ* tiene un *PPP* de 5%, *TCP-Tahoe* y *TCP-NewReno* tiene un 12.5%, y *TCP-Vegas* un 24%. Esta diferencia es sustancial y repercute en la QoS de las aplicaciones. Estos valores son el resultado de los mecanismos de control de congestión que usa cada protocolo.

Conforme la cantidad de carga de tráfico en la red se incrementa, la probabilidad de perder paquetes también lo hace; como lo muestra la figura 5.10. *PCCQ* es mejor que el resto hasta cuando la red presenta una carga menor a los 300 p/s. como se observa a los 380 y 600 seg de simulación. Por encima de este valor de carga, *TCP-Tahoe* y *TCP-NewReno* son mejores que *PCCQ*. Sin embargo, esta mejora es relativa, ya que *PCCQ* envía más paquetes por segundo (como se vera en los resultados de la siguiente

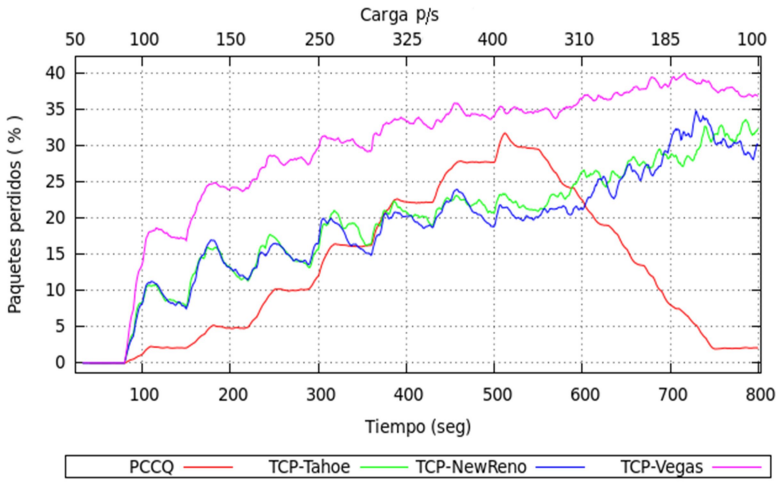

Figura 5.10: Comparación del Porcentaje de paquetes perdidos.

métrica) para la misma carga de red; lo cual hace que se incremente la probabilidad de paquetes perdidos. Considerando esta situación, se demuestra que *PCCQ* tiene mejores resultados que los diferentes protocolos TCPs.

De la figura 5.10 se observa que sólo *PCCQ* muestra una forma de "campana", en sus valores de *PPP*, siguiendo el comportamiento de la carga de la red. Esto obedece al control de congestión de cada protocolo de Transporte, ya que *PCCQ* actúa conforme la carga presente en la red; mientras que los protocolos TCPs actúan cuando se presenta una falla en la entrega de un paquete. Entonces, los resultados de los TCPs están dominados por el ajuste de la tasa de transmisión de sus mecanismos de control de congestión (Sec.2.3.2).

PCCQ puede detectar si la pérdida de un paquete es por desbordamiento de búfer o por saturación de la red (Sec.4.3.2). Con ello decidir si administra los búfer cuando se detecte que están congestionados, o regular el acceso al canal de comunicaciones cuando se encuentre saturado. Además, *PCCQ* permite cambiar los valores de configuración de los parámetros de operación del módulo de Resolución, para reducir con mayor impacto los tráficos tipo 2 y 3, con el fin de mejorar los valores de *PPP*.

En conclusión, *PCCQ* supera los resultados y posibilidades de acción que los protocolos TCPs pueden ofrecer. Además, tiene la ventaja de ser consciente, en todo momento, de la importancia de cada tipo de tráfico.

Desempeño o Throughput de la red (*TH*)

La figura 5.11 presenta los resultados de la cantidad promedio de paquetes tipo 1 que recibe el nodo Sink (*TH*), de todos los nodos de la red y para cada protocolo de Transporte. En ésta se observa que el desempeño alcanzado por *PCCQ* supera el desempeño de los diferentes protocolos TCPs, desde cargas medias hasta cargas altas de tráfico. Por ejemplo, *PCCQ* es mejor desde los 152 seg de simulación (115 p/s de carga con 5 nodos) hasta los 750 seg (100 p/s de carga con 4 nodos); prácticamente todo el tiempo del período de pruebas. Sólo es superado cuando existe una carga baja en la red menores a 100 p/s), como se observa a los 100 y 750 seg de simulación. Sin embargo, la magnitud de mejora es importante en el resto del tiempo. Por ejemplo, a los 500 seg cuando la red presenta máxima carga (400 p/s o 20 nodos), *PCCQ* alcanza un valor de *TH* de 88 p/s; mientras que *TCP-Vegas* tiene 31.5 p/s, y *TCP-Tahoe* y *TCP-NewReno* un 28.34 p/s. Estos valores representan un 64.20% y un 67.79%, respectivamente; lo cual demuestran claramente el desempeño del protocolo *PCCQ*.

Derivado de los resultados anteriores, vamos a explicar el por qué de la contundencia del desempeño de *PCCQ*. *PCCQ* cuenta con un módulo de Detección que le mantiene informado del grado de congestión de la red y del tipo de congestión (desbordamiento de búfer y/o red saturada), por lo cual, puede aplicar controles de solución precisos; lo que no pueden hacer los protocolos TCPs por su falta de información. Además, *PCCQ* cuenta con el módulo de Priorización, que le permite identificar el tipo de prioridad del paquete para administrarlo en consecuencia, junto con el grado de congestión de la red. Esto le permite al nodo enviar más paquetes tipo 1 con respecto de los paquetes tipo 2 y 3, provocando que la capacidad total de la red sea aprovechada por los paquetes más prioritarios. Estas características, le permiten a *PCCQ* ajustar suave y específicamente la reducción del tráfico que cada nodo puede enviar a la red durante los diferentes estados de congestión.

A diferencia de *PCCQ*, los protocolos TCPs usan mecanismos más agresivos y "planos" de control del flujo de tráfico en los nodos, como se ha mencionado en las pruebas anteriores. Si bien, los TCPs tienen diferencias en su control de congestión, aplican en esencia lo siguiente: asumen que cualquier perdida de paquetes es por una red congestionada, pudiendo ser una baja instantánea de la calidad del canal inalámbrico

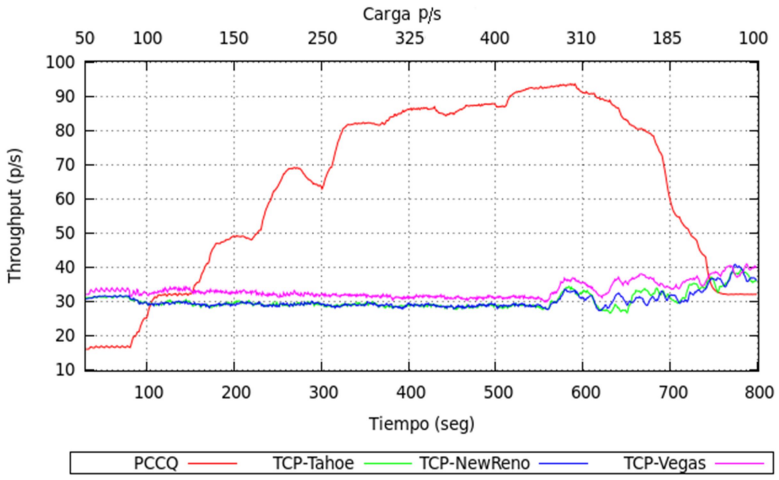

Figura 5.11: Comparación del Throughput de la red.

(lo cual es muy común) y no por la saturación de paquetes en los búfers. Entonces, cuando ésto sucede, los TCPs reducen agresivamente la tasa de envío de paquetes, buscando resolver el problema de congestión cuando no lo es. Lo anterior se traduce en los resultados que presenta la figura 5.11, donde se observa que el TH alcanzado para los diferentes protocolos TCPs es muy bajo. Este ajuste recurrente de la tasa de datos se aprecia de mejor manera en la figura 5.12. Observe, por ejemplo, que el valor de *TH* a los 172 seg es de 28.2 p/s, posteriormente este valor comienza a subir hasta 30.3 p/s a los 219 seg. En este lapso de tiempo no se activaron más nodos en la red, el tráfico se mantuvo en 150 p/s hasta antes de los 220seg (ver código en el Anexo A). Lo cual permitió mantener y/o incrementar la cantidad de paquetes en cada ventana de transmisión; ya que si bien tiene oscilaciones en su valor de *TH*, mantiene una tendencia a la alza. Sin embargo, después de los 219 seg el valor de *TH* tiene una tendencia a la baja; esto coincide con la activación de tres nodos más en la red (a los 220, 230 y 240 seg), que incrementan la carga en la red hasta 200 p/s (Fig.5.12). Entonces, contrariamente a lo esperado, cuando se generan más paquetes por segundo, *TCP-Tahoe* obtiene un decremento en su valor de *TH*. Ya que más nodos en la red incrementan la probabilidad de colisiones, provocando la activación repetida del mecanismo de control y por consecuencia la reducción del *TH*. Este efecto provoca que el desempeño alcanzado de *TH* no pueda crecer, y se mantenga oscilando al rededor de los 29 p/s.

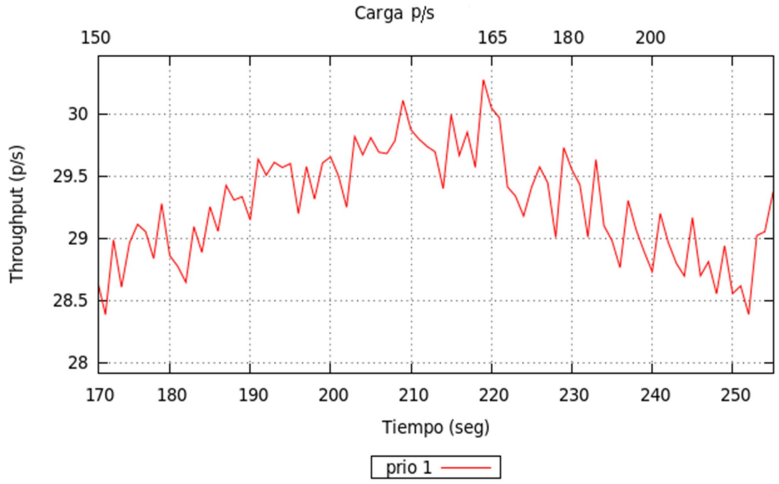

Figura 5.12: Throughput de la red para el protocolo *TCP-Tahoe*.

5.4 Discusión de los resultados

En conclusión, a los resultados obtenidos en las pruebas de comparación de los protocolos, se observó que *PCCQ* presenta un control de congestión más eficiente; ya que supera, en la mayor parte del tiempo, a los protocolos *TCP-Tahoe*, *TCP-NewReno* y *TCP-Vegas*, al respecto de los valores alcanzados en las métricas de *RTP*, *PPP* y *TH*. Por ejemplo, para *RTP*, *PCCQ* se mantiene por debajo de los 22 ms, como se propuso en el umbral de retardo *Rtn* dentro del módulo de Detección (Fig.4.23); a diferencia de los 29 ms de *TCP-Vegas* y de los más de 52.3 ms de *TCP-Tahoe* y *TCP-NewReno*. En cuanto al desempeño de *TH*, *PCCQ* presenta los mejores resultados, alcanzando un valor máximo de 93.5 p/s a los 590 seg; contra 37.61 p/s para *TCP-Vegas*, 31.1 p/s para *TCP-Tahoe* y *TCP-NewReno*. En cuanto a la métrica de *PPP*, *PCCQ* obtiene los mejores resultados cuando la red presenta una carga no mayor a 300 p/s. Por el contrario, *PCCQ* tiene un mayor porcentaje de *PPP* en la parte de mayor congestión de la red (a los 500 seg de simulación). Sin embargo, en esta parte, *PCCQ* tiene un desempeño superior a los protocolos TCPs, al respecto de las métricas de *TH* y *RTP*. Estos resultados tiene varios beneficios dentro de la red WSN; ya que *PCCQ* puede permitir un mayor número de nodos (usuarios o pacientes) dentro de la misma red, para un valor dado de QoS de la aplicación. Además, permite diferenciar los tipos de

tráficos en la red, para darles una calidad de comunicación diferencial de acuerdo a su importancia. Otro beneficio es referente al ahorro de energía, al reducir la cantidad de paquetes perdidos por colisión de datos, que son los que más energía consumen en su envío. Todo ésto, permite una mejor cobertura de servicios de supervisión médica a través de las WSNs. Por ejemplo, en el trabajo de Ikram se menciona que una red utilizada para el servicio de transmisión de datos de una señal ECG debe ofrecer ciertos valores mínimos de QoS: una tasa de datos de 4 kbps y un retardo máximo extremo a extremo de 500 ms [Ikram et al., 2007]. Por su parte, en el trabajo de Cervantes menciona que la tasa de un ECG de tres puntas debe estar entre 1 y 6.4 kbps [Cervantes, 2014]. Entonces, si seleccionamos una tasa de 6.4 kbps y un retardo máximo de 500 ms, la señal ECG digitalizada cabe en 10 paquetes de 80 bytes (Sec.5.1.1 y 2.1.2). Por lo tanto, considerando los resultados alcanzados por *PCCQ*, un retardo de transmisión por paquete *RTP* de 22 ms produce un retardo total de 220 ms al enviar los 10 paquetes; lo cual esta por debajo del límite QoS en más de un 50%. En el caso de los protocolos TCPs, sólo *TCP-Vegas* esta por debajo del limite de los 500 ms con 290 ms; sin embargo, *TCP-Tahoe* y *TCP-NewReno* necesitan 600 ms para enviar los 10 paquetes.

En cuanto al TH, *PCCQ* alcanza un máximo de 93.5 p/s (o 59.84 kbps) para el tráfico tipo 1; con lo cual *PCCQ* puede soportar hasta 9.35 usuarios que porten un sensor ECG a 6.4 kbps. Sin embargo, considerando los protocolos TCPs, donde *TCP-Vegas* tiene un máximo de 37.61 p/s (o 24.07 kbps) y los otros dos protocolos tienen 31.1 p/s (o 19.90 kbps); soportan 3.76 y 3.11 usuarios, respectivamente. Resultando que *PCCQ* soporta un 50% más de usuarios portando una señal ECG con un QoS de 6.4 kbps.

Al respecto del porcentaje de paquetes perdidos, idealmente se buscaría que la red mantenga un cero porciento de paquetes perdidos; sin embargo, es complicado lograrlo y más en las redes inalámbricas. Por ejemplo, en nuestras pruebas de la figura 5.10 se observa que hasta los 88 seg de simulación, cuando la red tenia una carga menor a 100 p/s y sólo dos nodos activos, se mantuvo en 0% el valor de paquetes perdidos (*PPP*). Por otro lado, aplicar algoritmos para recuperar los paquetes perdidos, normalmente a través de la retrasmisión de los mismos, no es adecuado en ambientes donde el tiempo de entrega es crítico [Gama et al., 2008]. Ya que estos algoritmos provocan que se

incremente notablemente el retardo de transmisión, y por ende que se incumpla con los límites de QoS de la aplicación médica. Además, dado que las señales se recolectan continuamente, es preferible esperar la siguiente señal ECG, si la señal previa trae un alto porcentaje de paquetes perdidos. Estudios sobre la transmisión de señales ECG, demuestran que una transmisión continua de datos ECG, durante 30 seg, que tenga hasta 8% de paquetes perdidos puede ser funcional para una interpretación médica [Henrion et al., 2004]. Entonces, fijar un valor limite depende del grado de satisfacción del usuario de la aplicación en particular. Lo que si se puede establecer es, que el protocolo *PCCQ* permite incrementar el retardo de transmisión de los paquetes con el fin de reducir el porcentaje de paquetes perdidos; ya que podemos retenerlos en los búfers del módulo de Priorización y configurar los parámetros del algoritmo CSMA-CA para ampliar el tiempo de espera de acceso al canal y el umbral de falla de acceso. Reduciendo la probabilidad de colisiones en la red y de fallas por CAF. Por ejemplo, mantener por más tiempo la configuración de los parámetros del Estado I (Fig.4.23); el cual, como se ve en la figura 5.10 (corresponde a los extremos), reduce la cantidad de paquetes perdidos a costa del incremento del retardo. Además, dado que nuestro valor de *RTP* es 50% menos que el QoS de 500 ms, existe espacio para lo anterior.
Entonces, finalmente resaltamos la flexibilidad del protocolo *PCCQ* para adaptarse a los diferentes grados de congestión y permitir en cualquier momento el cambio de los parámetros de operación del propio protocolo. Gracias al diseño de interacción Cross-Layer, entre la capa de Transporte y la subcapa MAC del nodo. Además de que *PCCQ* es consiente del QoS del tráfico de las diferentes aplicaciones, realizando el control de congestión acorde a ello.

5.5 Conclusiones del capítulo

En este capítulo se presentó la evaluación del desempeño del protocolo *PCCQ*, para lo cual se describió el escenario de red para una WSN con aplicaciones de monitorización de señales médicas. Se diseño la generación del tipo de tráfico de forma tal que, el protocolo manejara múltiples aplicaciones en la misma red y creará una carga de tráfico con los

tres niveles de congestión, en la misma prueba. Además, se detallaron las métricas utilizadas en las pruebas y el diseño de los dos casos de estudio para la evaluación del protocolo.

Finalmente se presentaron los resultados de cada caso de estudio, donde en general, se mostró que el desempeño de *PCCQ* supera los casos de comparación. Primero, *PCCQ* es mejor que cuando se usa una red con la configuración "por omisión" de los parámetros del algoritmo CSMA-CA, propuesta por el estándar IEEE 802.15.4 y sin interacción entre las capa de Transporte y la subcapa MAC. Y en el segundo caso, *PCCQ* demuestra que puede ofrecer mejores niveles de QoS en las métricas propuestas, que los protocolos TCPs utilizados.

Con este capítulo se valida el funcionamiento del protocolo *PCCQ* para el control de congestión en una red WSN, con aplicaciones de supervisión de la salud; cumpliendo el principal objetivos de este trabajo de investigación.

CAPÍTULO 6

CONCLUSIONES

Con la intención de resolver el problema planteado de la *congestión* presente en la redes inalámbricas de sensores, este trabajo de tesis hace una propuesta de un protocolo de Transporte llamado *PCCQ*. *PCCQ*, además, cumple con el objetivo general propuesto en nuestro trabajo: *Desarrollar un protocolo de Transporte para el control de congestión, basado en en el diseño Cross-Layer, para redes WSNs-IEEE 802.15.4, consciente de la QoS de las aplicaciones y del grado de congestión de la red.*. Ya que el protocolo *PCCQ*, como se mostró en los resultados de evaluación, puede controlar la congestión de forma distribuida en cada nodo de la red, y de forma adaptable de acuerdo al grado de congestión experimentado por el nodo y a la prioridad del tipo de tráfico que atiende.

Para su operación, y derivado del análisis de las necesidades planteadas, el protocolo *PCCQ* esta compuesto por cuatro módulos interconectados (Fig.4.4, Sec.4.3): *(i)* módulo de Priorización de tráfico, *(ii)* módulo de Detección de Congestión, *(iii)* módulo de Notificación de Congestión, y *(iv)* módulo de Resolución de Congestión. Estos módulos se insertan en cada nodo de la red para cumplir el objetivo del control

de la congestión. Esta propuesta responde la primer pregunta de investigación: *¿Qué módulos deben formar el control de la congestión, que consideren el QoS de las aplicaciones?*.

Para optimizar la operación de *PCCQ*, y cumplir con el objetivo planteado, se diseño bajo el concepto de Cross-Layer; ya que, como lo asumen diferentes autores (Sec.2.4), este tipo de diseños supera el desempeño alcanzado por los protocolos de comunicación, al respecto de los que no lo hacen. La interacción múlticapa le permite aprovechar las funciones de cada capa para trabajar en sintonía a los mismos objetivos de optimización; haciendo consciente a las capas de información valiosa, que sin este diseño seria imposible conocer. Para este diseño, se propuso la interacción entre la capa de Transporte y la subcapa de acceso al medio MAC (Sec.4.3). Ya que estas entidades son las que más impacto tienen para controlar la congestión y el desempeño de la comunicación de datos; como se expuso en la "evaluación del impacto de la congestión en la QoS" (Sec.2.2.2). Esta consideración responde la segunda pregunta de investigación *¿Qué capas de la "pila" de protocolos de red deben interactuar en el diseño Cross-Layer?*.

La tercer, y última, pregunta de investigación tiene que ver con la operación interna del protocolo *PCCQ*, *¿Qué parámetros de éstas capas se deben compartir en el diseño Cross-layer para instrumentar la solución?*. Estos parámetros tienen que ver con la forma en que el protocolo mide el grado de congestión y hace el control de la misma (Sec.4.3.5 y Fig.4.23). La subcapa MAC envía a la capa de Transporte: la cantidad de paquetes perdidos por falla de acceso al medio (CAF), la cantidad de paquetes que llegan y salen de ésta, y el tiempo de salida de cada paquete. Por su parte, la capa de Transporte le envía los valores de configuración de los parámetros de operación del algoritmo CSMA-CA de la subcapa MAC (macMinBE, macMaxBE y macMaxCSMABackoff), de acuerdo al grado de congestión presente. El resto de los parámetros necesarios para la operación de los cuatro módulos de *PCCQ* se comparten en la misma capa de Transporte, entre los diferentes módulos. Lo cual hace evidente que es un intercambio de información sencillo entre capas, pero realmente útil. Ya que la capa de transporte es consciente de la congestión a nivel de red, gracias a la información de la subcapa MAC; y por su parte, la subcapa MAC sabe que valores son los más adecuados para su algoritmo CSMA-CA, de acuerdo al grado de congestión y al tipo,

ya sea congestión del búfer del nodo o del enlace (Sec.4.3.2 y Ec.4.6).

Con la intención de informar acerca del logro de los objetivos planteados, se presentan las siguientes conclusiones; para el primer objetivo particular: *Establecer el marco de actuación del trabajo de investigación, que incluye el diseño del escenario (requerimientos) y la determinación de las métricas de evaluación para el análisis del desempeño del protocolo*. Para éste, su cumpliendo se inscribe en el capitulo 4, específicamente en la definición del *Modelo de Sistema* (Sec.4.2), que incluye las especificaciones del modelo de RED y de NODO utilizado en nuestra propuesta; y en las especificaciones de operación de cada módulo del protocolo *PCCQ* (Sec.4.3).

Al respecto del objetivo 2: *Delimitar la información y funcionalidades principales de la capa de transporte y subcapa MAC, que serán incluidas en la propuesta de diseño Cross-Layer*. Este fue cubierto en las respuestas a la preguntas de investigación 2 y 3, aquí explicadas.

De igual forma, el objetivo 3, *Desarrollar el protocolo de transporte Cross-Layer bajo las especificaciones de diseño definidas y utilizando el estándar IEEE 802.15.4*, tiene que ver con las respuestas a las tres preguntas de investigación aquí expuestas.

Finalmente, el objetivo 4, *Desarrollar el marco de trabajo para evaluar el desempeño del protocolo de transporte, que incluya una red de sensores y el estándar IEEE 802.15.4*; tiene que ver con lo expuesto en el capítulo 5. En el cual se expone la configuración del escenario de red, las características del tráfico y el diseño de dos casos de estudio, para la evaluación del desempeño del protocolo *PCCQ*. Los casos de estudio tienen que ver con las pruebas del escenario de red, con y sin el protocolo *PCCQ*; y con la comparación contra otros protocolos de control de congestión de capa de Transporte, al respecto del desempeño alcanzado (Sec.5.2 y 5.3, respectivamente). De los resultados obtenidos, en las diferentes pruebas, se demuestra que el protocolo *PCCQ* realiza el control de la congestión de forma dinámica, de acuerdo al grado de congestión presente en la red; y es consciente del tipo de tráfico que atiende. Además, los resultados de *PCCQ* superan el desempeño alcanzado por los protocolos utilizados en la comparación.

Entonces, a través del cumplimiento de cada objetivo particular y de los resultados obtenidos en la pruebas de evaluación, se concluye que el protocolo *PCCQ* cumple el objetivo general del trabajo de tesis y da respuesta a la problemática planteada.

6.1 Aportaciones

La contribución principal de esta tesis es un protocolo de transporte que puede soportar múltiples aplicaciones en la misma red, proporcionar un control de congestión variable, distribuido, que es consiente del grado de congestión y el QoS del tráfico presente en la red, que mantiene un valor establecido de retardo extremo a extremo, maximizando la entrega de datos y alcanzando valores aceptables de paquetes perdidos.
El protocolo propuesto *PCCQ* se diferencia de otras soluciones, presentadas en el estado del arte, por las siguientes características:

- Un diseño Cross-Layer, utilizando interacción entre la capa de Transporte y la subcapa MAC. Con lo cual se obtiene mejores resultados del control de congestión. Además, de mejorar la operación del estándar IEEE 802.15.4, tan sólo con una administración diferente de sus parámetros de acceso al medio.

- Cuenta con un módulo de Priorización de paquetes, que incluye un algoritmo de Ordenamiento Cíclico Ponderado para WSNs (WWRR); el cual permite manejar varios niveles de prioridad de forma simple y una gran velocidad de respuesta (Sec.4.3.1).

- Tiene consciencia del grado de congestión presentes en la red y del QoS requerido de las aplicaciones, al momento de aplicar el control de congestión.

- No genera paquetes de control para la operación del protocolo, evitando contribuir a la congestión y al consumo de energía; haciendo un uso óptimo de la red de comunicaciones.

- El protocolo fue diseñado considerando las capacidades y limitaciones de las tecnologías usadas para el desarrollo de las WSNs, como las que utilizan el estándar IEEE 802.15.4. Esto permite la posibilidad de incrustar el protocolo en dispositivos de baja capacidad hardware y software.

Por lo tanto, *PCCQ* puede contribuir al desafío técnico de habilitar una comunicación eficiente, que permita materializar los beneficios potenciales de las WSNs en la vigilancia de la salud de forma remota.

6.2 Trabajo futuro

Como trabajo futuro de investigación se visualizan las siguientes acciones principales:

1. Incorporar las funciones de recuperación de paquetes perdidos al protocolo *PCCQ*, para cubrir los dos principales requisitos de un protocolo de transporte. Manteniendo la consciencia de no rebasar los limites impuestos de retardo de transmisión extremo a extremo.

2. Contemplar la movilidad entre clúster dentro de la WSN, mediante la incorporación de mecanismos de "traspaso de clúster" vinculados al protocolo de Transporte.

3. Estudiar la posibilidad de extender el diseño Cross-Layer a la capa de Aplicación, con la intención de tener una interacción coordinada con las aplicaciones y el control de congestión. Sin embargo, es necesario considerar que ésto incrementa la complejidad del diseño, por ello, se requiere un análisis cuidadoso.

4. Se propone instrumentar el protocolo *PCCQ* en una red de pruebas físicas, con dispositivos de comunicación inalámbrica bajo el estándar IEEE 802.15.4; para validar de forma real la operación y los beneficios del protocolo.

ACRÓNIMOS

Ack Acknowledgement (Reconocimiento)

AIAD Additive Increase Additive Decrease (Incremento Aditivo y Decremento Aditivo)

BAN Body Area Network (Red de Área Corporal)

Beacons Beacons (Balizas)

bps bits por segundo

CAF Channel Access Failure (Falla de Acceso al Canal)

CBR Constant Bit Rate (Tasa de Bit Constante)

CCA Clear Channel Assessment (Evaluación de canal Libre)

CRC Cyclic Redundancy Check (Verificación Cíclica de Redundancia)

CSMA-CA Carrier Sense Multiple Access with Collision Avoidance (Acceso Múltiple por Detección de Portadora con Prevención de Colisiones)

ECG Electrocardiogram (Electrocardiograma)

ED Energy Detection (Detección de Energía)

EMG Electromyogram (Electromiografía)

EWMA Exponential Weighted Moving Average (Promedio móvil Ponderado exponencial)

FCS Frame Check Sequence (Secuencia de Verificación de Marco)

FIFO First in-first out (Primero en Llegar-Primero en Salir)

Hz Hertz

LLC Logical Link Control (Control de Enlace Lógico)

LQI Link Quality Indication (Indicación de la calidad del enlace)

MAC Medium Access Control (Control de Acceso al Medio)

MCPS-SAP MAC Common Part sublayer-SAP (Punto de Acceso al Servicio de la Parte Común de la Subcapa MAC)

MFR MAC Footer (Fin del Marco MAC)

MHR MAC Header (Encabezado MAC)

MLME-SAP MAC sublayer Management Entity-SAP (Punto de Acceso al Servicio de la Entidad de Gestión de la Subcapa MAC)

MPDU MAC Protocol Data Unit (Unidad de Datos del Protocolo MAC)

MSDU MAC Service Data Unit (Unidad de Datos de Servicio MAC)

NC Notificación de Congestión

OSI Open Systems Interconnection (Interconexión de Sistemas Abiertos)

PAN Personal Area Network (Red de Área Personal)

PCCQ Protocolo de transporte para el control de Congestión, consciente del QoS

PD-SAP PHY Data-SAP (Punto de Acceso al Servicio de Datos de la Capa Física)

PER Packet Error Rate (Tasa de Paquetes Perdidos)

PHR PHY Header (Encabezado PHY)

PHY Physical (Física)

PLME-SAP PHY Layer Management Entity-SAP (Punto de Acceso al Servicio de la Entidad de Gestión de la Capa Física)

POS Personal Operating Space (Espacio de Operación Personal)

PPDU PHY Protocol Data Unit (Unidad de Datos del Protocolo PHY)

PSDU PHY Service Data Unit (Unidad de Datos de Servicio PHY)

QoS Quality of Service (Calidad de Servicio)

RTP Retardo promedio de Transmisión de Paquete extremo a extremo

SAP Service Access Point (Punto de Acceso al Servicio)

SFD Start of Frame Delimiter (Delimitador de Inicio de Marco)

SHR Synchronization Header (Encabezado de Sincronización)

SSCS Service-Specific convergence sublayer (subcapa de servicios específicos de convergencia)

TCP Transmission Control Protocol (Protocolo de Control de Transmisión)

TH Throughput (Desempeño de la red)

UDP User Datagram Protocol (Protocolo de Datagrama de Usuario)

WLAN Wireless Local Area Network (Red Inalámbrica de Área Local)

WLANs Wireless Local Area Networks (Redes Inalámbricas de Área Local)

WMAN Wireless Metropolitan Area Network (Red Inalámbrica de Área Metropolitana)

WPAN Wireless Personal Area Network (Red Inalámbrica de Área Personal)

WPANs Wireless Personal Area Networks (Redes Inalámbricas de Área Personal)

WSN Wireless Sensor Network (Red Inalámbrica de Sensores)

WSNs Wireless Sensor Networks (Redes Inalámbricas de Sensores)

WWRR WSNs Weighted Round Robin (Ordenamiento Cíclico Ponderado para WSNs)

ANEXO A

CÓDIGO DE PROGRAMACIÓN DE LAS PRUEBAS DE SIMULACIÓN

A continuación se presenta el código de programación, en lenguaje TCL, que incluye las configuraciones de la WSN, tanto de la red como de cada uno de los nodos. En este código se incluyen las diferentes capas de un nodo: capa de Aplicación, con el código CBR; la capa de Transporte, que incluyen los diferentes protocolos utilizados (PCQQ, UDP, TCP-Tahoe, TCP-NewReno y TCP-Vegas); la capa de Red, con un la configuración de un protocolo estático; la subcapa MAC y la capa Física, estas parte del estándar IEEE 802.15.4. Además, se agregan la activación de los nodos para generar el tráfico segun el diseño planeado en las diferentes pruebas de evaluación.

A.1 Código TCL de la configuración de las pruebas

```
# OPCIONES GLOBALES QUE SON UTILIZADAS
set val(chan) Channel/WirelessChannel
```

```
set val(prop) Propagation/TwoRayGround
set val(netif) Phy/WirelessPhy/802_15_4
set val(mac) Mac/802_15_4
set val(ll) WSNLL
set val(ant) Antenna/OmniAntenna
set val(ifqlen) 21
set val(nn) 21
set val(rp) DumbAgent
set val(x) 35
set val(y) 35
################################
set val(nam) wpan_demo4b.nam
set val(traffic1) cbr
set val(agent) pccq # poner el tipo de Protocolo
if { "$val(agent)" == "pccq" } {
    set val(ifq) Queue/WWRR
} else {
    set val(ifq) Queue/DropTail/PriQueue
}
################################
proc getCmdArgu {argc argv} {
    global val
    for set i 0 {$i ¡ $argc} {incr i} {
        set arg [lindex $argv $i]
        if {[string range $arg 0 0] != "-"} continue
        set name [string range $arg 1 end]
        set val($name) [lindex $argv [expr $i+1]]
        }
}
getCmdArgu $argc $argv
set appTime1 30.0 ;# in seconds
```

```
set appTime2 50.0 ;# in seconds
set stopTime 800.0 ;# in seconds
set bandera 1
set ns_ [new Simulator]
set tracefd [open ./wpan_demo4b.tr w]
$ns_ trace-all $tracefd
if { "$val(nam)" == "wpan_demo4b.nam" } {
    set namtrace [open ./$val(nam) w]
    $ns_ namtrace-all-wireless $namtrace $val(x) $val(y)
}
$ns_ puts-nam-traceall {# nam4wpan#}
Mac/802_15_4 wpanCmd verbose on
Mac/802_15_4 wpanNam namStatus on

if { "$val(agent)" == "tcp" } {
    Mac/802_15_4 wpanCmd ack4data on
} else {
    Mac/802_15_4 wpanCmd ack4data off
}
# For model 'TwoRayGround'
set dist(30m) 2.13643e-07
set dist(35m) 1.56962e-07
set dist(40m) 1.20174e-07
Phy/WirelessPhy set CSThresh_ $dist(40m)
Phy/WirelessPhy set RXThresh_ $dist(40m)
# set up topography object
set topo [new Topography]
$topo load_flatgrid $val(x) $val(y)
# Create God
set god_ [create-god $val(nn)]
set chan_1_ [new $val(chan)]
```

```
# configure node, ESTA APAGADO LA TRAZA DE AGENTE $ns_ node-config-
adhocRouting $val(rp) \
    -llType $val(ll) \
    -macType $val(mac) \
    -ifqType $val(ifq) \
    -ifqLen $val(ifqlen) \
    -antType $val(ant) \
    -propType $val(prop) \
    -phyType $val(netif) \
    -topoInstance $topo \
    -agentTrace ON \
    -routerTrace OFF \
    -macTrace ON \
    -movementTrace OFF \
    -channel $chan_1_
for {set i 0} {$i ¡ $val(nn) } {incr i} {
    set node_($i) [$ns_ node]
    $node_($i) random-motion 0
}
# 21 nodos del 0 al nodo 20, y de los cuales hay 20 nodos sensores que pueden generar
datos
source ./wpan_demo4b.scn
$ns_ at 0.0 "$node_(0) NodeLabel \"PAN Coor\""
$ns_ at 0.0 "$node_(0) sscs startCTPANCoord 0"
$ns_ at 0.5 "$node_(1) sscs startCTDevice 0 0"
$ns_ at 1.8 "$node_(9) sscs startCTDevice 0 0"
$ns_ at 3.1 "$node_(13) sscs startCTDevice 0 0"
$ns_ at 4.4 "$node_(19) sscs startCTDevice 0 0"
$ns_ at 5.8 "$node_(2) sscs startCTDevice 0 0"
$ns_ at 7.1 "$node_(7) sscs startCTDevice 0 0"
$ns_ at 8.4 "$node_(11) sscs startCTDevice 0 0"
```

```
$ns_ at 9.5 "$node_(16) sscs startCTDevice 0 0"
$ns_ at 10.3 "$node_(3) sscs startCTDevice 0 0"
$ns_ at 10.5 "$node_(6) sscs startCTDevice 0 0"
$ns_ at 11.8 "$node_(12) sscs startCTDevice 0 0"
$ns_ at 12.1 "$node_(17) sscs startCTDevice 0 0"
$ns_ at 12.6 "$node_(20) sscs startCTDevice 0 0"
$ns_ at 12.8 "$node_(5) sscs startCTDevice 0 0"
$ns_ at 13.0 "$node_(10) sscs startCTDevice 0 0"
$ns_ at 14.3 "$node_(14) sscs startCTDevice 0 0"
$ns_ at 14.8 "$node_(18) sscs startCTDevice 0 0"
$ns_ at 15.0 "$node_(4) sscs startCTDevice 0 0"
$ns_ at 15.3 "$node_(8) sscs startCTDevice 0 0"
$ns_ at 15.7 "$node_(15) sscs startCTDevice 0 0"
Mac/802_15_4 wpanNam PlaybackRate 4ms
for {set i 0} {$i ¡ $val(nn)} {incr i} {
    if { "$val(agent)" == "pccq" } {
        puts "Tipo de agente: PCCQ"
        set agent_($i) [new Agent/PCCQ]
    } elseif { "$val(agent)" == "udp" } {
        puts "Tipo de agente: UDP"
        set agent_($i) [new Agent/UDP]
    } else {
        set agent_($i) [new Agent/TCP]
        puts "Tipo de agente: TCP"
        eval $agent_($i) set packetSize_ 80
    }
    eval $ns_ attach-agent \$node_($i) \$agent_($i)
    eval \$agent_($i) get_macObject [$node_($i) set mac_(0)]
    eval \$agent_($i) get_ifqObject [$node_($i) set ifq_(0)]
    if { "$val(agent)" == "pccq" } {
    eval \$agent_($i) change MaxBE 6
```

```
    eval \$agent_($i) change MinBE 6
    eval \$agent_($i) change MaxBackoffs 7
    eval \$agent_($i) set_resolucion 1
    eval \$agent_($i) set_queue 4 2 1
    }
    if {[string compare [lindex $argv 0] all] == 0 } {
    eval \$agent_($i) imprimir true
    } elseif [ string is integer [lindex $argv 0]] {
        if { [expr [lindex $argv 0]] == $i } {
            eval $agent_($i) imprimir true          }
    }
    if {[ string is double [lindex $argv 1]]} {
        eval \$agent_($i) change tiempo_espera [lindex $argv 1] }
}
eval \$agent_(1) imprimir_especial true
######## CREA UN TRAFICO CBR #########
proc cbrtraffic { src dst tampack starttime interval numpaq prioridad stop_time} {
    global ns_ node_ agent_ nuestro_protocolo val stopTime
    if { "$val(agent)" == "tcp" } {
        set null_($dst) [new Agent/TCPSink]
        if { $src == 1 } {
            eval \$null_($dst) imprimir true
        }
    } else {
        set null_($dst) [new Agent/MYPCCQ]
        if { $src == 1 } {
            eval \$null_($dst) imprimir true
        }
    }
    eval $ns_ attach-agent \$node_($dst) \$null_($dst)
    eval $ns_ connect \$agent_($src) \$null_($dst)
```

```
    set cbr($src) [new Application/Traffic/CBR]
    eval \$cbr($src) set packetSize_ $tampack
    eval \$cbr($src) set interval_ $interval
    eval \$cbr($src) set random_ 0
    eval \$cbr($src) set_secuencia 4 4 4
    if {$numpaq != 0 } {
        eval \$cbr($src) set maxpkts_ $numpaq
    }
    eval \$cbr($src) attach-agent \$agent_($src)
    if { $prioridad != 0 } {
        eval \$cbr($src) set prio_ $prioridad
    }
    set paq_seg [expr 1/$interval]
    $ns_ at $starttime "$cbr($src) start"
#para detener el envió de datos del nodo a nivel de Aplicación
    if { $stop_time == 0 } {
        set tiempo_parar $stopTime
    } else {
        set tiempo_parar $stop_time
    }
    $ns_ at $tiempo_parar "$cbr($src) stop"
    puts "\nPara el tiempo: $tiempo_parar trafico: $paq_seg"
}
# Activación del tráfico de los nodos, invoca el proc, cbrtraffic(); if { "$val(traffic1)"
== "cbr"} {
    ${val(traffic1)}traffic 1 0 80 [expr $appTime1] 0.02 0 0 0
    ${val(traffic1)}traffic 2 0 80 [expr $appTime1+50.02] 0.066666667 0 0 0
    ${val(traffic1)}traffic 3 0 80 [expr $appTime1+60.0] 0.066666667 0 0 0
    ${val(traffic1)}traffic 4 0 80 [expr $appTime1+70.0] 0.05 0 0 0
    ${val(traffic1)}traffic 5 0 80 [expr $appTime1+120.02] 0.066666667 0 0 710
    ${val(traffic1)}traffic 6 0 80 [expr $appTime1+130.0] 0.066666667 0 0 720
```

```
    ${val(traffic1)}traffic 7 0 80 [expr $appTime1+140.0] 0.05 0 0 730
    ${val(traffic1)}traffic 8 0 80 [expr $appTime1+190.0] 0.066666667 0 0 670
    ${val(traffic1)}traffic 9 0 80 [expr $appTime1+200.0] 0.04 0 0 680
    ${val(traffic1)}traffic 10 0 80 [expr $appTime1+210.0] 0.066666667 0 0 690
    ${val(traffic1)}traffic 11 0 80 [expr $appTime1+260.02] 0.066666667 0 0 630
    ${val(traffic1)}traffic 20 0 80 [expr $appTime1+270.06] 0.04 0 0 640
    ${val(traffic1)}traffic 13 0 80 [expr $appTime1+280.0] 0.066666667 0 0 650
    ${val(traffic1)}traffic 14 0 80 [expr $appTime1+330.0] 0.04 0 0 590
    ${val(traffic1)}traffic 15 0 80 [expr $appTime1+340.02] 0.066666667 0 0 600
    ${val(traffic1)}traffic 16 0 80 [expr $appTime1+350.0] 0.066666667 0 0 610
    ${val(traffic1)}traffic 17 0 80 [expr $appTime1+400.0] 0.05 0 0 550
    ${val(traffic1)}traffic 18 0 80 [expr $appTime1+410.0] 0.066666667 0 0 560
    ${val(traffic1)}traffic 19 0 80 [expr $appTime1+420.0] 0.066666667 0 0 570
    ${val(traffic1)}traffic 12 0 80 [expr $appTime1+470.02] 0.028571429 0 0 740
for {set i 0} {$i ¡ $val(nn) } {incr i} {
    $ns_ at $stopTime "$node_($i) reset"
}
$ns_ at $stopTime "stop"
$ns_ at $stopTime "puts \"\nNS EXITING...\n\""
$ns_ at $stopTime "$ns_ halt"
proc stop {} {
    global ns_ tracefd appTime1 val env
    $ns_ flush-trace
    close $tracefd
    set hasDISPLAY 0
    foreach index [array names env] {
        if { ("$index" == "DISPLAY") && ("$env($index)" != "") } {
            set hasDISPLAY 1
        }
    }
    if { ("$val(nam)" == "wpan_demo4b.nam") && ("$hasDISPLAY" == "1") } {
```

```
        exec nam wpan_demo4b.nam &
    }
}

#agrega aleatorieidad a cada simulacion
$defaultRNG seed 2
puts "\nStarting Simulation..."
$ns_ run
```

BIBLIOGRAFÍA

[Abed et al., 2010] Abed, A., Ali, A., and Aslam, N. (2010). Building an hmi and demo application of wsn-based industrial control systems. In *Energy, Power and Control (EPC-IQ), 2010 1st International Conference on*, pages 302–306.

[Akyildiz et al., 2008] Akyildiz, I., Melodia, T., and Chowdhury, K. (2008). Wireless multimedia sensor networks: Applications and testbeds. *Proceedings of the IEEE*, 96(10):1588–1605.

[Akyildiz and Vuran, 2010] Akyildiz, I. and Vuran, M. C. (2010). *Wireless Sensor Networks*. John Wiley & Sons, Inc., New York, NY, USA.

[Akyildiz et al., 2002] Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., and Cayirci, E. (2002). Wireless sensor networks: A survey. *Comput. Netw.*, 38(4):393–422.

[Alam and Hong, 2009] Alam, M. M. and Hong, C. S. (2009). CRRT: congestion-aware and rate-controlled reliable transport in wireless sensor networks. *IEICE TRANSACTIONS on Communications*, E92-B(1):184–199.

[Alemdar and Ersoy, 2010] Alemdar, H. and Ersoy, C. (2010). Wireless sensor networks for healthcare: A survey. *Computer Networks*, 54(15):2688–2710.

[Ameen et al., 2008] Ameen, M., Nessa, A., and Kwak, K. S. (2008). QoS issues with focus on wireless body area networks. In *Convergence and Hybrid Information Technology, 2008. ICCIT '08. Third International Conference on*, volume 1, pages 801–807.

[Anastasi et al., 2011] Anastasi, G., Conti, M., and Di Francesco, M. (2011). A comprehensive analysis of the MAC unreliability problem in IEEE 802.15.4 wireless sensor networks. *IEEE Transactions on Industrial Informatics*, 7(1):52–65.

[ANSI/IEEE, 1984] ANSI/IEEE (1984). Ansi/ieee std 802.2: Logical link control.

[Arnon et al., 2003] Arnon, S., Bhastekar, D., Kedar, D., and Tauber, A. (2003). A comparative study of wireless communication network configurations for medical applications. *IEEE Wireless Communications*, 10(1):56–61.

[Braden et al., 2003] Braden, R., Faber, T., and Handley, M. (2003). From protocol stack to protocol heap: role-based architecture. *ACM SIGCOMM Computer Communication Review*, 33(1):17–22.

[Brakmo et al., 1994] Brakmo, L. S., O'Malley, S. W., and Peterson, L. L. (1994). TCP vegas: New techniques for congestion detection and avoidance. In *Proceedings of the Conference on Communications Architectures, Protocols and Applications*, volume 24 of *SIGCOMM '94*, pages 24–35, New York, NY, USA. ACM.

[Buenrostro et al., 2013] Buenrostro, M. R., Iván, N. H. J., Antonio, G. I. J., María, C. L., Mabel, V. B., and de Dios., S. L. J. (2013). Análisis de factores que afectan el QoS que ofrecen las WSN aplicado a entornos de salud. *Revista de Difución Científica*, 7(2):54–62.

[Buratti et al., 2009] Buratti, C., Conti, A., Dardari, D., and Verdone, R. (2009). An overview on wireless sensor networks technology and evolution. *Sensors*, 9(9):6869–6896.

[Castillo-Effer et al., 2004] Castillo-Effer, M., Quintela, D., Moreno, W., Jordan, R., and Westhoff, W. (2004). Wireless sensor networks for flash-flood alerting. In *Devices,*

Circuits and Systems, 2004. Proceedings of the Fifth IEEE International Caracas Conference on, volume 1, pages 142–146.

[Cervantes, 2014] Cervantes, de Ávila, H. (2014). *ARQUITECTURA DE E-SALUD BASADA EN REDES INALÁMBRICAS DE SENSORES*. PhD thesis, UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA, Ensenada, B.C.

[Cypher et al., 2006] Cypher, D., Chevrollier, N., Montavont, N., and Golmie, N. (2006). Prevailing over wires in healthcare environments: benefits and challenges. *Communications Magazine, IEEE*, 44(4):56–63.

[Darwish and Hassanien, 2011] Darwish, A. and Hassanien, A. E. (2011). Wearable and implantable wireless sensor network solutions for healthcare monitoring. *Sensors (Basel, Switzerland)*, 11(6):5561–5595. PMID: 22163914 PMCID: PMC3231450.

[Edirisinghe and Zaslavsky, 2013] Edirisinghe, R. and Zaslavsky, A. (2013). Cross-layer contextual interactions in wireless networks. *IEEE Communications Surveys Tutorials*, Early Access Online.

[Egbogah and Fapojuwo, 2011] Egbogah, E. E. and Fapojuwo, A. O. (2011). A survey of system architecture requirements for health care-based wireless sensor networks. *Sensors (Basel, Switzerland)*, 11(5):4875–4898. PMID: 22163881 PMCID: PMC3231387.

[Fernández et al., 2009] Fernández, R., Ordieres, J., Martínez de Pisón, A., González, A., Alba, F., Lostado, R., and Pernía (2009). *REDES INALÁMBRICAS DE SENSORES: TEORÍA Y APLICACIÓN PRÁCTICA*, volume 26 of *Material Didáctico. Ingenierías.* Universidad de La Rioja, España, 1 edition. ISBN: 978-84-692-3007-7.

[Floyd et al., 2004] Floyd, S., Gurtov, A., and Henderson, T. (2004). Rfc 3782: The NewReno modification to TCP's fast recovery algorithm.

[Francini et al., 2001] Francini, A., Chiussi, F., Clancy, R., Drucker, K., and Idirene, N. (2001). Enhanced weighted round robin schedulers for accurate bandwidth distribution in packet networks. *Computer Networks*, 37(5):561 – 578.

[Fu et al., 2014] Fu, B., Xiao, Y., Deng, H., and Zeng, H. (2014). A survey of cross-layer designs in wireless networks. *IEEE Communications Surveys Tutorials*, 16(1):110–126.

[Gallego et al., 2005] Gallego, J., Hernandez-Solana, A., Canales, M., Lafuente, J., Valdovinos, A., and Fernandez-Navajas, J. (2005). Performance analysis of multiplexed medical data transmission for mobile emergency care over the umts channel. *Information Technology in Biomedicine, IEEE Transactions on*, 9(1):13–22.

[Gama et al., 2008] Gama, O., Carvalho, P., Afonso, J., and Mendes, P. (2008). Quality of service support in wireless sensor networks for emergency healthcare services. In *Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE*, pages 1296–1299.

[Giancoli et al., 2008] Giancoli, E., Jabour, F., and Pedroza, A. (2008). CTCP: reliable transport control protocol for sensor networks. In *International Conference on Intelligent Sensors, Sensor Networks and Information Processing, 2008. ISSNIP 2008*, pages 493 –498.

[Gungor et al., 2008] Gungor, V. C., Akan, A. B., and Akyildiz, I. F. (2008). A real-time and reliable transport (RT) ^{2} protocol for wireless sensor and actor networks. *IEEE/ACM Trans. Netw.*, 16(2):359–370.

[Gutiérrez et al., 2011] Gutiérrez, J., Callaway, E., and Barrett, R. (2011). *Low-rate wireless personal area networks: enabling wireless sensors with IEEE 802.15.4*. Number ISBN-13: 978-0738162850 in IEEE Standard Information Network,. IEEE Press., New York, third edition edition.

[Haas, 2001] Haas, Z. (2001). Design methodologies for adaptive and multimedia networks. *Communications Magazine, IEEE*, 39(11):106–107.

[Halsall, 2006] Halsall, F. (2006). *Redes de Computadoras e Internet*. Madrid, España, 5 edition. ISBN:978-84-7829-083-3.

[Hanson et al., 2009] Hanson, M., Powell, H., Barth, A., Ringgenberg, K., Calhoun, B., Aylor, J., and Lach, J. (2009). Body area sensor networks: Challenges and opportunities. *Computer*, 42(1):58–65.

[Henrion et al., 2004] Henrion, S., Mailhes, C., and Castanié, F. (2004). Transmitting critical biomedical signals over unreliable connexionless channels with good QoS using advanced signal processing. *proc. of WSEAS*, pages 694–700.

[Ikram et al., 2007] Ikram, M., Seppanen, S., Sliz, R., Hamalainen, M., and Pomalaza-Ráez, C. (2007). Implementation issues for wireless medical devices. In *International Symposium on Medical Information and Communication Technology (ISMICT'07)*.

[Institute, 1980] Institute, I. S. (1980). User datagram protocol. http://www.ietf.org/rfc/rfc0768.txt.

[Institute, 1981] Institute, I. S. (1981). Transmission control protocol. http://www.ietf.org/rfc/rfc793.txt.

[Institute, 2008] Institute, I. S. (2008). The network simulator - ns-2. http://www.isi.edu/nsnam/ns/.

[ISO/IEC, 1994] ISO/IEC (1994). Osi reference model 7498-1: Architecture for open systems interconnection.

[Iyer et al., 2005] Iyer, Y., Gandham, S., and Venkatesan, S. (2005). STCP: a generic transport layer protocol for wireless sensor networks. In *14th International Conference on Computer Communications and Networks, 2005. ICCCN 2005. Proceedings*, pages 449 – 454.

[Jafari et al., 2005] Jafari, R., Encarnacao, A., Zahoory, A., Dabiri, F., Noshadi, H., and Sarrafzadeh, M. (2005). Wireless sensor networks for health monitoring. In *Proceedings of the The Second Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services*, MOBIQUITOUS '05, pages 479–781, Washington, DC, USA. IEEE Computer Society.

[Kawadia and Kumar, 2005] Kawadia, V. and Kumar, P. (2005). A cautionary perspective on cross-layer design. *Wireless Communications, IEEE*, 12(1):3–11.

[Konstantas and Herzog, 2003] Konstantas, D. and Herzog, R. (2003). Continuous monitoring of vital constants for mobile users: the mobihealth approach. In *Engineering in Medicine and Biology Society, 2003. Proceedings of the 25th Annual International Conference of the IEEE*, volume 4, pages 3728–3731.

[Kurose and Ross, 2013] Kurose, J. F. and Ross, K. W. (2013). *Computer Networking: A Top-Down Approach*. Addison-Wesley, 6 edition. ISBN-13: 978-0-13-285620-1.

[LAN/MAN Standard, 2006] LAN/MAN Standard, C. (2006). Ieee standard for information technology– local and metropolitan area networks– specific requirements– part 15.4: Wireless medium access control (mac) and physical layer (phy) specifications for low rate wireless personal area networks (wpans).

[Latré et al., 2011] Latré, B., Braem, B., Moerman, I., Blondia, C., and Demeester, P. (2011). A survey on wireless body area networks. *Wireless Networks*, 17(1):1–18.

[Lorincz et al., 2004] Lorincz, K., Malan, D., Fulford-Jones, T., Nawoj, A., Clavel, A., Shnayder, V., Mainland, G., Welsh, M., and Moulton, S. (2004). Sensor networks for emergency response: challenges and opportunities. *Pervasive Computing, IEEE*, 3(4):16–23.

[Malindi, 2011] Malindi, P. (2011). QoS in telemedicine. In Graschew, G., editor, *Telemedicine Techniques and Applications*. InTech.

[Mendes and Rodrigues, 2011] Mendes, L. D. and Rodrigues, J. J. (2011). A survey on cross-layer solutions for wireless sensor networks. *Journal of Network and Computer Applications*, 34(2):523 – 534. Efficient and Robust Security and Services of Wireless Mesh Networks.

[min LIN et al., 2011] min LIN, Q., chuan WANG, R., GUO, J., and juan SUN, L. (2011). Novel congestion control approach in wireless multimedia sensor networks. *The Journal of China Universities of Posts and Telecommunications*, 18(2):1 – 8.

[Nasser et al., 2013] Nasser, N., Karim, L., and Taleb, T. (2013). Dynamic multilevel priority packet scheduling scheme for wireless sensor network. *IEEE Transactions on Wireless Communications*, 12(4):1448–1459.

[Otto et al., 2005] Otto, C., Milenkovic, A., Sanders, C., and Jovanov, E. (2005). System architecture of a wireless body area sensor network for ubiquitous health monitoring. *Journal of Mobile Multimedia*, 1(4):307–326.

[Rahim et al., 2011] Rahim, M., Rashid, R., Ariffin, S. H. S., Fisal, N., Sarijari, M., and Hadi Fikri Abdul Hamid, A. (2011). Testbed design for wireless biomedical sensor network (wbsn) application. In *Computer Applications and Industrial Electronics (ICCAIE), 2011 IEEE International Conference on*, pages 284–289.

[Rahman et al., 2008a] Rahman, M., Monowar, M., and Hong, C. S. (2008a). A QoS adaptive congestion control in wireless sensor network. In *Advanced Communication Technology, 2008. ICACT 2008. 10th International Conference on*, volume 2, pages 941 –946.

[Rahman et al., 2008b] Rahman, M. A., Saddik, A. E., and Gueaieb, W. (2008b). Wireless sensor network transport layer: State of the art. In Mukhopadhyay, S. and Huang, R., editors, *Sensors*, volume 21 of *Lecture Notes in Electrical Engineering*, pages 221–245. Springer Berlin Heidelberg.

[Rajba et al., 2013] Rajba, S., Raif, P., Rajba, T., and Mahmud, M. (2013). Wireless sensor networks in application to patients health monitoring. In *Computational Intelligence in Healthcare and e-health (CICARE), 2013 IEEE Symposium on*, pages 94–98.

[Rao and Marandin, 2006] Rao, V. P. and Marandin, D. (2006). Adaptive backoff exponent algorithm for zigbee (IEEE 802.15.4). In Koucheryavy, Y., Harju, J., and Iversen, V. B., editors, *Next Generation Teletraffic and Wired/Wireless Advanced Networking*, number 4003 in Lecture Notes in Computer Science, pages 501–516. Springer Berlin Heidelberg.

[Rathnayaka et al., 2010] Rathnayaka, A. J. D., Potdar, V., Sharif, A., Sarencheh, S., and Kuruppu, S. (2010). Wireless sensor network transport protocol - a state of the art. In *Broadband, Wireless Computing, Communication and Applications (BWCCA), 2010 International Conference on*, pages 812–817.

[Rezaee et al., 2014] Rezaee, A. A., Yaghmaee, M. H., Rahmani, A. M., and Mohajerzadeh, A. H. (2014). Hoca: Healthcare aware optimized congestion avoidance and control protocol for wireless sensor networks. *Journal of Network and Computer Applications*, 37(1):216 – 228.

[Rohm et al., 2009] Rohm, D., Goyal, M., Xie, W., Polepalli, B., Hosseini, H., Divjak, A., and Bashir, Y. (2009). Dynamic reconfiguration in beaconless IEEE 802.15.4 networks under varying traffic loads. In IEEE, editor, *IEEE Global Telecommunications Conference, 2009. GLOBECOM 2009*, pages 1–8.

[Shaikh et al., 2010] Shaikh, F., Khelil, A., Ali, A., and Suri, N. (2010). TRCCIT: tunable reliability with congestion control for information transport in wireless sensor networks. In *Wireless Internet Conference (WICON), 2010 The 5th Annual ICST*, pages 1 –9.

[Sharif et al., 2010] Sharif, A., Potdar, V., and Rathnayaka, A. (2010). LCART: a cross-layered transport protocol for heterogeneous WSN. In *2010 IEEE Sensors*, pages 793 –796.

[Shnayder et al., 2005] Shnayder, V., Chen, B., Lorincz, K., Fulford-Jones, T., and Welsh, M. (2005). Codeblue: Wireless sensors for medical care. In *3rd International Conference on Embedded Networked Sensor Systems*.

[Skorin-Kapov and Matijasevic, 2010] Skorin-Kapov, L. and Matijasevic, M. (2010). Analysis of QoS requirements for e-health services and mapping to evolved packet system QoS classes. *Int. Journal of Telemedicine and Appl.*, 2010(628086):9:1–9:18.

[Sridevi and Usha, 2011] Sridevi, S. and Usha, M. (2011). Taxonomy of transport protocols for wireless sensor networks. In *Recent Trends in Information Technology (ICRTIT), 2011 International Conference on*, pages 467–472.

[Srivastava and Motani, 2005] Srivastava, V. and Motani, M. (2005). Cross-layer design: a survey and the road ahead. *Communications Magazine, IEEE*, 43(12):112–119.

[Stine, 2006] Stine, J. (2006). Cross-layer design of manets: The only option. In *Military Communications Conference, 2006. MILCOM 2006. IEEE*, pages 1–7.

[Tian et al., 2005] Tian, Y., Xu, K., and Ansari, N. (2005). Tcp in wireless environments: problems and solutions. *Communications Magazine, IEEE*, 43(3):S27–S32.

[Vuran and Akyildiz, 2010] Vuran, M. and Akyildiz, I. (2010). XLP: a cross-layer protocol for efficient communication in wireless sensor networks. *IEEE Transactions on Mobile Computing*, 9(11):1578 –1591.

[Wan et al., 2003] Wan, C.-Y., Eisenman, S. B., and Campbell, A. T. (2003). CODA: congestion detection and avoidance in sensor networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems*, SenSys '03, pages 266–279, New York, NY, USA. ACM.

[Wang et al., 2007] Wang, C., Li, B., Sohraby, K., Daneshmand, M., and Hu, Y. (2007). Upstream congestion control in wireless sensor networks through cross-layer optimization. *Selected Areas in Communications, IEEE Journal on*, 25(4):786–795.

[Wang and Abu-Rgheff, 2003] Wang, Q. and Abu-Rgheff, M. (2003). Cross-layer signalling for next-generation wireless systems. In *Wireless Communications and Networking, 2003. WCNC 2003. 2003 IEEE*, volume 2, pages 1084–1089 vol.2.

[Wood et al., 2008] Wood, A., Stankovic, J., Virone, G., Selavo, L., He, Z., Cao, Q., Doan, T., Wu, Y., Fang, L., and Stoleru, R. (2008). Context-aware wireless sensor networks for assisted living and residential monitoring. *Network, IEEE*, 22(4):26–33.

[Wu et al., 2003] Wu, H., Zhang, Q., and Zhu, W. (2003). Design study for multimedia transport protocol in heterogeneous networks. In *Communications, 2003. ICC '03. IEEE International Conference on*, volume 1, pages 567–571.

[Yang et al., 2013] Yang, P., Shao, J., Luo, W., Xu, L., Deogun, J., and Lu, Y. (2013). Tcp congestion avoidance algorithm identification. *Networking, IEEE/ACM Transactions on*, (99):1–14.

[Yick et al., 2008] Yick, J., Mukherjee, B., and Ghosal, D. (2008). Wireless sensor network survey. *Comput. Netw.*, 52(12):2292–2330.

[Yigitel et al., 2011] Yigitel, M. A., Incel, O. D., and Ersoy, C. (2011). QoS-aware MAC protocols for wireless sensor networks: A survey. *Comput. Netw.*, 55(8):1982–2004.

[Yunus et al., 2011] Yunus, F., Ismail, N.-S., Ariffin, S. H. S., Shahidan, A. A., Fisal, N., and Syed-Yusof, S. (2011). Proposed transport protocol for reliable data transfer in wireless sensor network (wsn). In *Modeling, Simulation and Applied Optimization (ICMSAO), 2011 4th International Conference on*, pages 1–7.

[Zimmermann, 1980] Zimmermann, H. (1980). Osi reference model–the iso model of architecture for open systems interconnection. *Communications, IEEE Transactions on*, 28(4):425–432.