



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

**METODOLOGÍA DE DISEÑO PARA MINIMIZAR EL
CONSUMO DE POTENCIA Y LOS RETRASOS EN REDES
INALÁMBRICAS DE SENSORES Y ACTUADORES CON
RETRASOS ACOTADOS**

Tesis Doctoral

Diego Martínez Castro

Directores

D. José Enrique Simó Ten
D. Alfons Crespo i Lorente



Valencia, España
Julio de 2011

AGRADECIMIENTOS

Deseo agradecer a todas las personas e instituciones que colaboraron en el desarrollo de esta tesis.

Agradezco a todos los profesores que me orientaron durante mis estudios de posgrado y en especial a mis directores de tesis José Enrique Simó y Alfons Crespo. Su colaboración, apoyo y enseñanza han sido fundamentales para la realización de este trabajo.

Quiero reconocer y agradecer a la Universidad Autónoma de Occidente por la comisión de estudios, la cual me permitió concentrarme totalmente en el desarrollo de esta tesis. Igualmente a la Universidad Politécnica de Valencia y en particular a los miembros del Grupo de Informática Industrial del DISCA, por su acogida y por facilitarme la posibilidad de realizar este trabajo.

A mi esposa, mis padres y en general a toda mi familia, por la paciencia que han tenido y la motivación generada para que este trabajo fuese concluido.

ÍNDICE GENERAL

LISTA DE FIGURAS	iv
LISTA DE TABLAS	vi
LISTA DE ACRÓNIMOS.....	vii
RESUM	ix
RESUMEN	xi
ABSTRACT	xiii
MOTIVACIÓN Y OBJETIVOS DE LA TESIS	1
1.1 OBJETIVOS DE LA TESIS	4
1.2 CONTRIBUCIONES	4
1.3 CONTENIDO DE LA TESIS	5
REVISIÓN DEL ESTADO ACTUAL EN REDES INALÁMBRICAS DE SENSORES Y ACTUADORES	7
2.1 APLICACIONES DE LAS REDES INALÁMBRICAS DE SENSORES Y ACTUADORES	7
2.1.1 Aplicaciones en el sector salud	8
2.1.2 Aplicaciones en el sector medioambiental	10
2.1.3 Aplicaciones en el sector agrícola	12
2.1.4 Aplicaciones en el sector industrial	13
2.1.5 Clasificación de las aplicaciones	13
2.2 ARQUITECTURA DE LOS NODOS DE WSAN	16
2.3 PROTOCOLOS DE COMUNICACIONES PARA WSAN	19
2.3.1 Protocolos de control de acceso al medio (MAC)	20
2.3.1.1 Estándar de comunicación IEEE 802.15.4	23
2.3.2 Algoritmos de encaminamiento	26
2.4 ALGORITMOS DE PLANIFICACIÓN DE TAREAS Y MENSAJES	28
2.5 ESCALADO DINÁMICO DE VOLTAJE	30
2.6 METODOLOGÍAS DE DISEÑO DE REDES INALÁMBRICAS DE SENSORES Y ACTUADORES	31
2.7 CONCLUSIONES	34

AGRADECIMIENTOS

Deseo agradecer a todas las personas e instituciones que colaboraron en el desarrollo de esta tesis.

Agradezco a todos los profesores que me orientaron durante mis estudios de posgrado y en especial a mis directores de tesis José Enrique Simó y Alfons Crespo. Su colaboración, apoyo y enseñanza han sido fundamentales para la realización de este trabajo.

Quiero reconocer y agradecer a la Universidad Autónoma de Occidente por la comisión de estudios, la cual me permitió concentrarme totalmente en el desarrollo de esta tesis. Igualmente a la Universidad Politécnica de Valencia y en particular a los miembros del Grupo de Informática Industrial del DISCA, por su acogida y por facilitarme la posibilidad de realizar este trabajo.

A mi esposa, mis padres y en general a toda mi familia, por la paciencia que han tenido y la motivación generada para que este trabajo fuese concluido.

ÍNDICE GENERAL

LISTA DE FIGURAS	iv
LISTA DE TABLAS	vi
LISTA DE ACRÓNIMOS	vii
RESUM	ix
RESUMEN	xi
ABSTRACT	xiii
MOTIVACIÓN Y OBJETIVOS DE LA TESIS	1
1.1 OBJETIVOS DE LA TESIS	4
1.2 CONTRIBUCIONES	4
1.3 CONTENIDO DE LA TESIS	5
REVISIÓN DEL ESTADO ACTUAL EN REDES INALÁMBRICAS DE SENSORES Y ACTUADORES	7
2.1 APLICACIONES DE LAS REDES INALÁMBRICAS DE SENSORES Y ACTUADORES	7
2.1.1 Aplicaciones en el sector salud	8
2.1.2 Aplicaciones en el sector medioambiental	10
2.1.3 Aplicaciones en el sector agrícola	12
2.1.4 Aplicaciones en el sector industrial	13
2.1.5 Clasificación de las aplicaciones	13
2.2 ARQUITECTURA DE LOS NODOS DE WSAN	16
2.3 PROTOCOLOS DE COMUNICACIONES PARA WSAN	19
2.3.1 Protocolos de control de acceso al medio (MAC)	20
2.3.1.1 Estándar de comunicación IEEE 802.15.4	23
2.3.2 Algoritmos de encaminamiento	26
2.4 ALGORITMOS DE PLANIFICACIÓN DE TAREAS Y MENSAJES	28
2.5 ESCALADO DINÁMICO DE VOLTAJE	30
2.6 METODOLOGÍAS DE DISEÑO DE REDES INALÁMBRICAS DE SENSORES Y ACTUADORES	31
2.7 CONCLUSIONES	34

ARQUITECTURA DE LOS NODOS	35
3.1 ARQUITECTURA PROPUESTA PARA LOS NODOS	35
3.2 ANÁLISIS DE LOS REQUISITOS DE CÓMPUTO EN UN CASO PARA LA DETECCIÓN DE ARRITMIAS	37
3.3 SELECCIÓN DEL PROTOCOLO DE COMUNICACIONES	39
VALIDACIÓN DEL DISEÑO	43
4.1 NOTACIÓN Y MODELO DEL SISTEMA	43
4.2 ANÁLISIS DE PLANIFICABILIDAD EXTREMO-EXTREMO PARA WSAN CON DISTANCIA ENTRE NODOS ACOTADA A UN ÚNICO SALTO	44
4.2.1 Caso de estudio: NCS con distancia entre nodos acotada a un único salto	46
4.3 ANÁLISIS DE PLANIFICABILIDAD EXTREMO-EXTREMO PARA WSAN MULTISALTO	54
4.4 VALIDACIÓN DE PROPIEDADES ESTRUCTURALES Y DE COMPORTAMIENTO EN WSAN	56
4.4.1 Componentes	56
4.4.2 Validación de propiedades estructurales y de comportamiento del diseño	57
4.4.3 Representación de componentes en CPN	59
METODOLOGIA DE DISEÑO DE WSAN	65
5.1 FUNCIONES DE COSTO	68
PRUEBAS Y VALIDACIÓN	71
6.1 CASO DE ESTUDIO: WSAN CON DISTANCIA ENTRE NODOS ACOTADA A UN ÚNICO SALTO	71
6.1.1 Consideraciones del sistema de detección de fallos	72
6.1.2 Consideraciones del NCS	73
6.1.3 Búsqueda del modo de funcionamiento óptimo	73
6.1.4 Resultados de simulación	81
6.1.5 Modelado y validación empleando CPN	84
6.2 CASO DE ESTUDIO 3: WSAN CON COMUNICACIÓN MULTISALTO	87
6.2.1 Consideraciones generales.....	87
6.2.2 Búsqueda del modo de funcionamiento óptimo	89
6.2.3 Resultados de simulación	91

6.3 CONCLUSIONES.....	94
VERIFICACIÓN EXPERIMENTAL	95
CONCLUSIONES Y LÍNEAS FUTURAS DE INVESTIGACIÓN	103
8.1 CONCLUSIONES.....	103
8.2 LÍNEAS FUTURAS DE INVESTIGACIÓN	105
BIBLIOGRAFÍA.....	106

Lista de figuras

Figura 1.1. Esquema general de una WSN.....	2
Figura 2.1. Imágenes del proyecto <i>The Great Duck Island</i>	11
Figura 2.2. Imagen de uno de los nodos utilizados en el proyecto LOFAR.....	12
Figura 2.3. Arquitectura de una aplicación industrial que utiliza WirelessHART	13
Figura 2.4. Diagrama comparativo del consumo de potencia en los diferentes modos de operación del TR1000, CC1000 y CC2420, (Li, 2008)	22
Figura 2.5. Topología Estrella	24
Figura 2.6. Topología <i>Peer-to-Peer</i>	24
Figura 2.7. Estructura de una supertrama	25
Figura 3.1. Arquitectura lógica de los nodos de alto desempeño	36
Figura 3.2. Diagrama de bloques del algoritmo de Pan y Tompkins	37
Figura 3.3. Caso de estudio propuesto para analizar el desempeño de una WSN que utiliza la ecuación (1) para el establecimiento de rutas.....	41
Figura 4.1. Representación general del NCS considerado	48
Figura 4.2. Representación del NCS como sistema sincronizado	48
Figura 4.3. Respuesta del sistema discreto, $T_s=110\text{ ms}$ y $\tau_r=80\text{ ms}$. Referencia tren de pulsos unitario	50
Figura 4.4. Planificación de los mensajes en la red TDMA.....	51
Figura 4.5. Planificación de tareas en el nodo que ejecuta la tarea <i>Actuación</i> del lazo de control 1	52
Figura 4.6. Respuesta del los tres lazos de control para una referencia tren de pulsos y una red TDMA	52
Figura 4.7. Planificación de los mensajes en la red Zigbee	53
Figura 4.8. Respuesta del los tres lazos de control para una referencia tren de pulsos y una red Zigbee	53
Figura 4.9. Envío de información en modo multisalto	54
Figura 4.10. Representación en CPN de un componente periódico	60
Figura 4.11. Representación en CPN de un componente aperiódico	60
Figura 4.12. Esquema general de la interacción entre una tarea y el planificador de tareas	61
Figura 4.13. Representación en CPN de un nodo con dos tareas periódicas y un planificador EDF	62

Figura 4.14. Esquema general de acceso de una red TDMA con 35 ranuras y dos GTS	63
Figura 4.15. Representación en CPN de una red TDMA con dos nodos	64
Figura 5.1. Metodología de diseño de WSAAN	65
Figura 6.1. Diagrama de componentes del caso de estudio: WSAAN con distancia entre nodos acotada a un único salto	72
Figura 6.2. Frente de Pareto para el escenario 3	80
Figura 6.3. Resultados de simulación del escenario 3, modo de funcionamiento 16	82
Figura 6.4. Resultados de simulación del escenario 3, modo de funcionamiento 19	83
Figura 6.5. Representación en CPN del caso de estudio: WSAAN con distancia entre nodos acotada a un único salto	84
Figura 6.6. Modelo en CPN del sistema a controlar	85
Figura 6.7. Modelo en CPN de la tarea de control	85
Figura 6.8. Modelo en CPN del actuador	86
Figura 6.9. Modelo en CPN del medidor	86
Figura 6.10. Esquema general del caso de estudio: WSAAN con comunicación multisalto	88
Figura 6.11. Respuesta del sistema de control para una señal de entrada cuadrada, con $T_s = 40ms$ y $\tau_r = 40ms$	88
Figura 6.12. Comparación del consumo de potencia en los nodos que ejecutan las tareas <i>Controlador</i> , para la configuración óptima y una sub-óptima	92
Figura 6.13. Periodos entre la generación de eventos en el lazo de control 2	93
Figura 6.14. Consumo de potencia en los nodos que ejecutan las tareas <i>Controlador</i>	93
Figura 6.15. Resultados de simulación. Lazo de control 1 periódico y lazo de control 2 por eventos	94
Figura 7.1. Arquitectura del NCS implementado	96
Figura 7.2. Esquema de ejecución de las tareas del NCS implementado	96
Figura 7.3 Respuesta simulada del sistema de orden 2 a una entrada escalón unitario	99
Figura 7.4. Resultados experimentales y de simulación del NCS	101
Figura 7.5. Resultados experimentales, PID por eventos	102

Lista de tablas

Tabla 2.1. Análisis de los requisitos para cada tipo de aplicación	15
Tabla 2.2. Características de los nodos sensores	18
Tabla 3.1. Tiempo de cómputo para desarrollar el algoritmo de Pan y Tompkins	38
Tabla 3.2. Indicador P_A*U en el desarrollo el algoritmo de Pan y Tompkins	38
Tabla. 3.3. Resultados del análisis del desempeño de una WSN que utiliza la ecuación (1) para el establecimiento de rutas	42
Tabla 4.1. Parámetros para el análisis de planificabilidad extremo-extremo, en redes con distancia entre nodos acotada a un único salto	45
Tabla 4.2. Resultados del análisis de planificabilidad.....	51
Tabla 4.3. Conjunto mínimo de componentes para el diseño de WSAN	57
Tabla 4.4. Secuencia de marcas en el lugar CPU entre los instantes de tiempo 40 y 43	61
Tabla 6.1. Tiempos de cómputo para el subsistema de diagnóstico	73
Tabla 6.2. Tiempos de cómputo para el subsistema de control	73
Tabla 6.3. Valores de las funciones de costo para el escenario 1	75
Tabla 6.4. Valores de las funciones de costo para el escenario 2	76
Tabla 6.5. Valores de las funciones de costo para el escenario 3.....	77
Tabla 6.6. Tiempos de cómputo de las tareas para cada uno de los lazos de control	89

Lista de acrónimos

AES	Advanced Encryption Standard
AODV	Ad-hoc On-demand Distance Vector
CAP	Periodo de acceso por contención
CFP	Periodo libre de contención
CPN	Rede de Petri coloreadas
CSMA	Acceso múltiple por sensado de portadora
CSMA-CA	Acceso múltiple por sensado de portadora con evitación de colisión
DSR	Dynamic Source Routing
DVS	Dynamic Voltage Scaling
ED	Energy detection
EDCA	Enhanced Distributed Channel Access
EDF	Earliest Deadline First
FCFS	First-Come First-Served
GTS	Guaranteed Time Slot
HAN	Home Area Networks
ICI	Intervalo crítico inicial
ICI _N	Intervalo crítico inicial de la red
LEACH	Low-Energy Adaptive Clustering Hierarchy
LQI	Link quality indication
MAC	Control de acceso al medio
NCS	Networked Control Systems
RM	Rate Monotonic

SISO	Single Input, Single Output
TDMA	Acceso múltiple por división de tiempo
WSAN	Wireless Sensors and Actuators Networks
WSN	Wireless Sensors Networks

Resum

La naturalesa i complexitat de les noves aplicacions de monitorització i control, junt amb els nous avanços en miniaturització, la comunicació ubíqua i la convergència digital, han impulsat el desenvolupament de les xarxes de sensors i actuadors permetent concebre noves aplicacions que anteriorment eren poc viables degut a les limitacions tecnològiques.

Els nodes de la xarxa es distribueixen sobre l'àrea de cobertura de l'aplicació i cooperen per a resoldre els problemes específics. Aquestes arquitectures de sistemes augmenten la flexibilitat dels propis sistemes, contribueixen a disminuir l'impacte produït per les fallades en un component i els costos d'implementació. A més a més de facilitar les activitats de diagnòsi, manteniment i traçabilitat en els processos, propicien nous nivells de seguretat, confort i productivitat en totes les àrees. Això permet cobrir des dels entorns industrials fins als individus, com per exemple aplicacions de domòtica i de l'atenció de la salut.

En general, aquestes aplicacions demanen un elevat nivell de garantia de bon funcionament i seguretat, i la seva naturalesa concurrent i no determinista fa que l'anàlisi i disseny siguin complexos. Açò ha generat que en alguns casos no s'obtinga una bona correspondència entre els resultats experimentals i els objectius de funcionament proposats, la qual cosa és conseqüència de l'ús de models imprecisos per a analitzar i dissenyar estos sistemes, mètodes de validació poc elaborats i plataformes que no suporten els models utilitzats. El que és desitjable es que el procés de construcció d'aquests sistemes siga clar i compregua els elements necessaris per a la seua realització.

Aquesta nova àrea integra disciplines que han tingut desenvolupaments aïllats, com ara instrumentació, xarxes de comunicacions, teoria de control, processament de senyals, computació i informàtica; pel que cada una emprava mètodes de disseny i anàlisi amb formalismes de modelatge i eines distintes, on cada representació posa de manifest certes característiques sense contemplar els altres subsistemes. És a dir, que els diferents components del sistema es dissenyen i analitzen amb diferents eines, la qual cosa limita l'anàlisi de la interacció entre els mateixos i l'ús de criteris metodològics de disseny concurrent que permeten optimitzar alguns dels paràmetres crítics en estes aplicacions.

Adicionalment, un dels grans desafiaments en el desenvolupament d'aquestes aplicacions se centra en que l'optimització del disseny en funció d'un factor pot ocasionar la disminució de la qualitat de l'aplicació en funció d'un altre, amb la qual cosa s'ha d'aconseguir un compromís entre els mateixos.

Davant de les condicions anteriors, en aquesta tesi s'han realitzat diverses aportacions orientades a establir una metodologia de disseny d'aplicacions de xarxes de sensors i actuadors amb retards tancats. Aquesta metodologia està basada en la coordinació de les

activitats dels diferents nivells de l'arquitectura dels nodes d'aquests sistemes, amb la qual cosa s'ha contribuït a la solució d'alguns dels desafiaments presents en esta àrea.

La metodologia proposada utilitza una arquitectura de node en què es consideren els requisits de les aplicacions, permet la verificació de les restriccions temporals extrem-
extrem, i troba una distribució dels components de l'aplicació per a obtindre un balanç entre el mínim consum de potència i els retards mínims en la generació d'accions.

Adicionalment es presenta un conjunt mínim de components en xarxes de Petri Colorejades que permeten analitzar el funcionament d'aquests sistemes i verificar les seves propietats estructurals i de comportament.

Els resultats obessos mitjançant simulació i experimentació corroboren la validesa i l'efectivitat de la metodologia de disseny proposada, i permeten apreciar la importància d'obtindre configuracions de funcionament que realitzen un balanç entre la potència consumida en els nodes i els retards presents en aplicacions de control sobre WSN, mantenint els retards tancats.

Resumen

La naturaleza y complejidad de las nuevas aplicaciones de monitorización y control, junto con los nuevos avances en miniaturización, la comunicación ubicua y la convergencia digital, han impulsado el desarrollo de las redes inalámbricas de sensores y actuadores, lo que ha permitido concebir nuevas aplicaciones que anteriormente eran poco viables debido a limitantes tecnológicos.

Los nodos de la red se distribuyen sobre el área de cobertura de la aplicación y cooperan para la solución de problemas específicos. Estas arquitecturas de sistemas aumentan la flexibilidad de los mismos, contribuyen a disminuir el impacto producido por los fallos en un componente y los costes de implementación, facilitan las actividades de diagnóstico, mantenimiento y trazabilidad en los procesos, y propician nuevos niveles de seguridad, confort y productividad en todas las áreas, cubriendo desde los entornos industriales hasta los individuos, como por ejemplo aplicaciones de domótica y del cuidado de la salud.

En general, estas aplicaciones demandan un elevado nivel de garantía de buen funcionamiento y seguridad, y su naturaleza concurrente y no determinista hace que su análisis y diseño sean complejos. Esto ha generado que en algunos casos no se obtenga una buena correspondencia entre los resultados experimentales y los objetivos de funcionamiento propuestos, lo cual es consecuencia del uso de modelos imprecisos para analizar y diseñar estos sistemas, métodos de validación poco elaborados y plataformas que no soportan los modelos empleados. Por lo que es deseable que el proceso de construcción de estos sistemas sea claro y comprenda los elementos necesarios para su realización.

Esta nueva área integra disciplinas que han tenido desarrollos aislados, tales como instrumentación, redes de comunicaciones, teoría de control, procesamiento de señales, computación e informática; por lo que cada una emplea métodos de diseño y análisis con formalismos de modelado y herramientas distintas, en donde cada representación pone de manifiesto ciertas características sin contemplar los demás subsistemas. Es decir, que los distintos componentes del sistema se diseñan y analizan en herramientas diferentes, lo que limita el análisis de la interacción entre los mismos y el uso de enfoques metodológicos de diseño concurrente que permitan optimizar algunos de los parámetros críticos en estas aplicaciones.

Adicionalmente, uno de los grandes desafíos en el desarrollo de estas aplicaciones se centra en que la optimización del diseño en función de un factor, puede ocasionar la disminución de la calidad de la aplicación en función de otro, con lo cual se debe lograr un compromiso entre los mismos.

Ante las condiciones anteriores, en esta tesis se han realizado varias aportaciones orientadas a establecer una metodología de diseño de aplicaciones de redes inalámbricas de sensores y actuadores con retrasos acotados. Esta metodología se soporta en la

coordinación de las actividades de los diferentes niveles de la arquitectura de los nodos de estos sistemas, con lo cual se ha contribuido a la solución de algunos de los desafíos presentes en esta área.

La metodología propuesta utiliza una arquitectura de nodo en la que se consideran los requisitos de las aplicaciones, permite la verificación de las restricciones temporales extremo-extremo, y encuentra una distribución de los componentes de la aplicación para obtener un balance entre el mínimo consumo de potencia y los retrasos mínimos en la generación de acciones.

Adicionalmente se presenta un conjunto mínimo de componentes en redes de Petri coloreadas que permiten analizar el funcionamiento de estos sistemas y verificar sus propiedades estructurales y de comportamiento.

Los resultados obtenidos por medio de simulación y experimentación corroboran la validez y la efectividad de la metodología de diseño propuesta, y permiten apreciar la importancia de obtener configuraciones de funcionamiento que realicen un balance entre la potencia consumida en los nodos y los retrasos presentes en aplicaciones de control sobre WSN, manteniendo los retrasos acotados.

Abstract

The nature and complexity of new applications for monitoring and control, along with new advances in miniaturization, ubiquitous communication and digital convergence, have driven the development of wireless sensors and actuators networks, this has allowed to devise new applications which were previously impractical due to technological limitations.

The network nodes are distributed over the coverage area of the application and cooperate to solve specific problems. This system architecture increases the flexibility thereof, helps to reduce the impact caused by faults in a component and implementation costs, facilitates the diagnosis, maintenance and traceability processes, and fosters new levels of safety, comfort and productivity in all areas, covering from industrial environments to individuals, such as home automation applications and healthcare.

In general, these applications demand a high level of assurance of proper operation and safety, and its concurrent non-deterministic nature makes their analysis and design complex. This has led in some cases not get a good match between experimental results and the proposed performance specifications, which is a result of imprecise models for analyzing and designing these systems are used, and to make use of inadequate validation methods and platforms that do not support the models utilized. It is desirable to have a complete process to develop these systems which contains all the elements necessary for its realization.

This new area includes disciplines that have been isolated developments, such as instrumentation, communications networks, control theory, signal processing, computer and information, so each one uses different methods for analysis and design with own modeling formalisms and tools, where each representation highlights certain characteristics without considering the other subsystems. That is, the different system components are designed and analyzed by different tools, limiting the analysis of the interaction between them and the use of concurrent design methodological approaches to optimize some of the critical parameters in these applications.

Additionally, one of the greatest challenges in developing these applications is focused on the design optimization based on a factor, could result in decreased quality of implementation in terms of another, then a compromise between them should be established.

Given the above conditions, this thesis has made several contributions aimed at establishing a methodology for designing applications of wireless sensors and actuators network with bounded delays. This methodology is supported in coordinating the activities of different levels of the architecture of the nodes of these systems, thereby contributing to solving some of the challenges present in this area.

The proposed methodology uses a node architecture which considers the application requirements, allows the verification of end-to-end time constraints, and finds a distribution of application components to provide a balance between the minimum power consumption and minimal delays in generating actions.

Additionally is presented a minimal set of components in Colored Petri Nets to analyze the performance of these systems and verify their structural and behavioral properties.

The results obtained by simulation and experimentation confirm the validity and effectiveness of the design methodology proposed, and allow to appreciate the importance of to obtain operation configurations in which they make a balance between the power consumed at the nodes and delays in control applications implemented on WSN, while maintains bounded delays.

Capítulo 1

MOTIVACIÓN Y OBJETIVOS DE LA TESIS

La naturaleza y complejidad de las nuevas aplicaciones de monitorización y control, junto con los nuevos avances en miniaturización, la comunicación ubicua y la convergencia digital, han impulsado a la distribución de estas aplicaciones entre diferentes sistemas empotrados separados geográficamente, denominados nodos, los cuales se comunican a través de redes de comunicaciones y cooperan para la solución de problemas específicos.

Estas arquitecturas de sistemas aumentan la flexibilidad de los mismos, contribuyen a disminuir el impacto producido por los fallos en un componente y los costes de implementación, facilitan las actividades de diagnóstico, mantenimiento y trazabilidad en los procesos, y propician nuevos niveles de seguridad, confort y productividad en todas las áreas, cubriendo desde los entornos industriales hasta los individuos.

Desde finales de la década de los 90 ha surgido un gran interés por el uso de redes inalámbricas para la interconexión de los nodos, permitiendo concebir nuevas aplicaciones que anteriormente eran poco viables debido a limitaciones en la portabilidad de la aplicación, la movilidad de los nodos y la autoconfiguración del sistema. Inicialmente se desarrollaron aplicaciones de monitorización, dando origen al término redes inalámbricas de sensores (*Wireless Sensors Networks* - WSN). Estas investigaciones estuvieron motivadas principalmente por aplicaciones militares. Los requisitos de estas aplicaciones dieron lugar a la definición de una WSN como una red de gran escala (miles de nodos, que cubren grandes áreas geográficas), inalámbrica, *ad hoc*, multisalto, de nodos sensores con arquitectura homogénea, pequeños, principalmente estáticos (después de la implementación), que pueden ser dispersados de manera aleatoria sobre el área de interés (Römer, 2004).

Posteriormente se han considerado dominios de aplicaciones civiles, como medioambientales y de monitorización de especies, agrícolas, industriales, de atención sanitaria, etc. Los proyectos desarrollados en estas áreas han indicado que la definición anterior de una WSN no necesariamente se ajusta a estas aplicaciones. Estas WSN pueden estar compuestas de nodos heterogéneos y móviles, la topología de la red puede ser tan simple como una topología en estrella, etc. Adicionalmente, los nodos han sido dotados de interfaces de actuación, dando origen a las redes inalámbricas de sensores y actuadores (*Wireless Sensors and Actuators Networks* - WSAN), con lo cual se han agregado nuevos requisitos de calidad de servicio en las aplicaciones, principalmente relacionados con los retrasos en la generación de acciones sobre los sistemas físicos.

Algunas actividades de la aplicación pueden tener una asignación preestablecida, como es el caso de las funciones de medición y actuación, debido a la conexión física de

ciertos nodos con el sistema con el que se interactúan; mientras que las demás funciones se asignan a un nodo siguiendo razonamientos relacionados con el cumplimiento de algún criterio, como por ejemplo la optimización en el consumo de potencia, la minimización de los retrasos en el sistema, entre otras. En la figura 1.1 se presenta de manera general una aproximación a la estructura de estas aplicaciones.

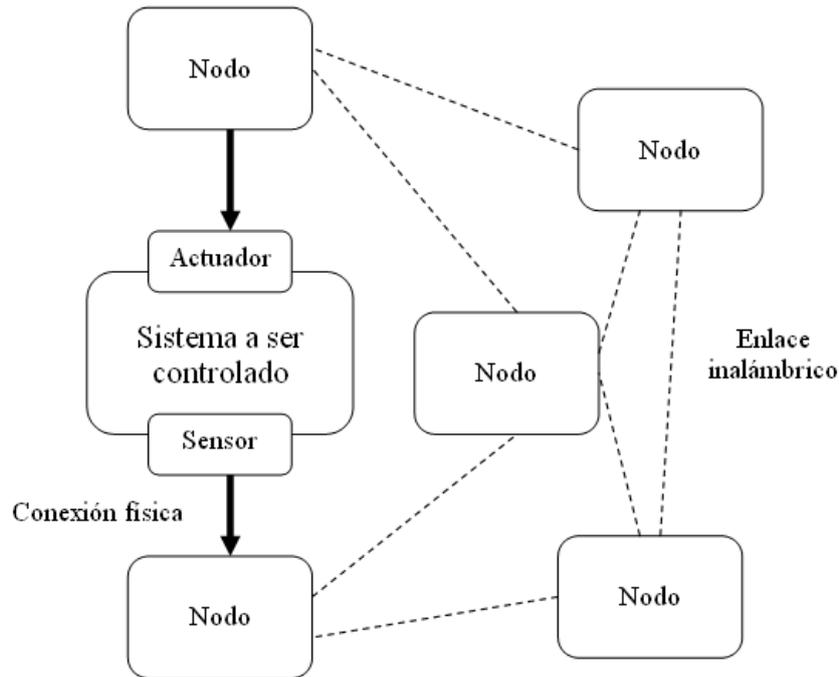


Figura 1.1. Esquema general de una WSN

Los parámetros más relevantes en el diseño de estas aplicaciones son el tiempo de vida, la escalabilidad, el tiempo de respuesta, la sincronización de los nodos, la protección de la información, y el cumplimiento de plazos de tiempo real, (Lester, 2003), (Martínez, 2008). Dependiendo del tipo de aplicación varía la importancia de estos objetivos en el diseño. Uno de los grandes desafíos en el desarrollo de estas aplicaciones se centra en que la optimización del diseño en función de uno de los anteriores factores, puede ocasionar la disminución de la calidad de la aplicación en función de otro factor, razón por la cual se debe lograr un compromiso entre los mismos.

En general, estas aplicaciones demandan un elevado nivel de garantía de buen funcionamiento y seguridad, y su naturaleza concurrente y no determinista hace que su análisis y diseño sean complejos. Estos sistemas pueden evolucionar por diferentes estados, y dada su magnitud es muy probable que se presenten condiciones de funcionamiento no contempladas durante las fases de diseño, que conlleven a un desempeño no deseado del sistema. Esto ha generado que en algunos casos no se obtenga una buena correspondencia entre los resultados experimentales y los objetivos de funcionamiento propuestos (Martínez, 2009b), lo cual es consecuencia del uso de

modelos imprecisos para analizar y diseñar estos sistemas, métodos de validación poco elaborados y plataformas que no soportan los modelos empleados.

Los requisitos propuestos por estas aplicaciones han llevado a un cambio significativo en las metodologías de diseño, llevando a la interacción de varios equipos de trabajo y la automatización de algunas etapas del diseño. Sin embargo aún existe una gran diferencia entre los resultados reales, y los análisis teóricos y de simulación. Las técnicas, lenguajes y metodologías empeladas no son integrales y al pasar de una a otra en las diferentes etapas del desarrollo de un sistema, nunca se obtiene un modelo completo y menos evaluado y analizado.

La construcción de un sistema completo de WSN es complicado y exige mucho tiempo debido a la dependencia de muchos parámetros del sistema relacionados entre sí; además, el espacio de diseño asociado a la optimización de las configuraciones de WSN es enorme debido a la gran cantidad de configuraciones posibles del sistema y la complejidad de las interacciones entre los parámetros de los componentes y del sistema, por lo que la optimización efectiva en este espacio es un reto, (Shen, 2007).

Esta nueva área integra disciplinas que han tenido desarrollos aislados, tales como instrumentación, redes de comunicaciones, teoría de control, procesamiento de señales, computación e informática; por lo que cada una emplea métodos de diseño y análisis con formalismos de modelado y herramientas distintas, en donde cada representación pone de manifiesto ciertas características sin contemplar los demás subsistemas. Es decir, que los distintos componentes del sistema se diseñan y analizan en herramientas diferentes, lo que limita el análisis de la interacción entre los mismos y el uso de enfoques metodológicos de diseño concurrente que permitan optimizar algunos de los parámetros críticos en estas aplicaciones.

Estas aplicaciones plantean varios retos relacionados con las restricciones temporales en sistemas basados en eventos, los retrasos variables en el tiempo, los fallos, la reconfiguración, sistemas distribuidos de soporte a la toma de decisiones, diseño de protocolos soportados en redes inalámbricas que garanticen la calidad de servicio en aplicaciones con restricciones de tiempo real, soluciones de compromiso entre el diseño de las leyes de control y la ejecución en tiempo real. Por lo cual se hace necesario el desarrollo de *frameworks*, algoritmos, metodologías de diseño sistemático y herramientas para satisfacer la alta fiabilidad y los requisitos de seguridad para los componentes heterogéneos que cooperan a través de un medio complejo, y con restricciones espaciales y temporales, (Tariq, 2011).

Los actuales desafíos están principalmente relacionados con el soporte de los niveles de calidad de servicio propios de estas aplicaciones, la protección de la información, la sincronización entre los nodos, el planteamiento de nuevas arquitecturas de nodos y protocolos de comunicaciones que contribuyan a la disminución de las latencias y el consumo de potencia, y la proposición de estrategias de coordinación de las actividades de los niveles de la arquitectura del sistema que contribuyan a la optimización de los parámetros de mayor relevancia en el sistema, como también el desarrollo de métodos que permitan un diseño adecuado de estas aplicaciones.

En esta tesis se han realizado varias aportaciones orientadas a establecer una metodología de diseño de aplicaciones de redes inalámbricas de sensores y actuadores con retrasos acotados, que se soporta en la coordinación de las actividades de los diferentes niveles de la arquitectura de estos sistemas, con lo cual se ha contribuido a la solución de algunos de los desafíos presentes en esta área. Esta metodología utiliza una arquitectura de nodo en la que se consideran los requisitos de las aplicaciones, permite la verificación de las restricciones temporales extremo-extremo, y encuentra una distribución de los componentes de la aplicación para obtener un balance entre el mínimo consumo de potencia y los retrasos mínimos en la generación de acciones. Adicionalmente se presentan modelos de componentes en redes de Petri coloreadas que permiten la verificación de las propiedades estructurales y de comportamiento del sistema.

1.1 Objetivos de la tesis

El objetivo general de la tesis es el planteamiento de una metodología de diseño de redes inalámbricas de sensores y actuadores con retrasos acotados, que permita encontrar un balance entre la minimización del consumo de potencia y los retrasos en la generación de acciones.

Para abordar el objetivo general se enfatizó en los siguientes aspectos:

- Análisis de los requisitos de las aplicaciones.
- Análisis de las tecnologías y metodologías existentes para el diseño de WSN.
- Proposición de una arquitectura para los nodos de la red en WSN, cuyos niveles posibiliten el planteamiento de estrategias para minimizar el consumo de potencia y tener retrasos acotados.
- Proposición de métodos para verificar la planificabilidad extremo-extremo en WSN con nodos que utilicen la arquitectura propuesta.
- Proposición de un conjunto mínimo de componentes para el diseño y análisis de WSN, que permitan comprobar propiedades de comportamiento y estructurales del sistema.
- Realización de pruebas para validar los resultados obtenidos empleando la metodología de diseño propuesta.

1.2 Contribuciones

La principal contribución de esta tesis consiste en el planteamiento de una metodología de diseño para aplicaciones de redes inalámbricas de sensores y actuadores, que se soporta en una arquitectura de nodos en la que se consideran los requisitos de las aplicaciones, permite la verificación de las restricciones temporales del sistema, y encuentra una distribución de los componentes de la aplicación que minimiza el consumo de potencia y los retrasos.

Adicionalmente se presenta un conjunto mínimo de componentes en redes de Petri coloreadas que permiten diseñar el sistema y analizar sus propiedades estructurales y de comportamiento, como también realizar análisis dinámicos del sistema completo.

En los distintos ámbitos, las contribuciones que aporta este trabajo se pueden resumir en los siguientes puntos:

- Clasificación de las aplicaciones de las WSAN.
- Proposición de la arquitectura de los nodos que conforman la red.
- Evaluación del comportamiento temporal de sistemas distribuidos de control implementados sobre IEEE 802.15.4.
- Proposición de métodos para analizar la planificabilidad extremo-extremo en WSAN con distancia entre nodos acotada a un único salto y en redes multisalto.
- Proposición de un conjunto mínimo de componentes modelados en redes de Petri coloreadas (Colored Petri Nets – CPN), que permiten comprobar propiedades de comportamiento y estructurales de aplicaciones de WSAN, y realizar análisis dinámicos del sistema completo.
- Proposición de una metodología de diseño para WSAN con retrasos acotados, en la que se coordina la actividad de los niveles de la arquitectura de los nodos, con el propósito de encontrar soluciones óptimas que garanticen el cumplimiento de las restricciones temporales presentes en la aplicación, y un mínimo consumo de potencia y retrasos en el sistema.

1.3 Contenido de la tesis

Esta tesis se encuentra organizada en 8 capítulos constituidos de la siguiente manera:

Capítulo 2: Revisión del estado actual en redes inalámbricas de sensores y actuadores. Esta revisión está orientada a presentar los trabajos desarrollados y los desafíos presentes en las temáticas relacionadas con el diseño de redes inalámbricas de sensores y actuadores.

Capítulo 3: Arquitectura de los nodos. En este capítulo se realiza una propuesta para la arquitectura genérica de los nodos de las redes inalámbricas de sensores y actuadores. La arquitectura propuesta posibilita el desarrollo de sistemas sensibles a los retrasos en la comunicación, la reconfiguración del sistema en función del cambio dinámico de los requisitos de la aplicación, y al desarrollo de soluciones de compromiso entre el consumo de potencia y las latencias, garantizando la ejecución en tiempo real de los algoritmos de control en NCS.

Capítulo 4: Validación del diseño. En este capítulo se presentan los test de planificabilidad propuestos para verificar el cumplimiento de las restricciones

temporales extremo-extremo en WSAN, y un conjunto mínimo de componentes para verificar propiedades estructurales y de comportamiento en estas aplicaciones.

Capítulo 5: Metodología de diseño de WSAN. En este capítulo se presenta una metodología de diseño de redes inalámbricas de sensores y actuadores, que coordina la actividad de los niveles de la arquitectura de los nodos, encuentra una distribución óptima de los componentes de la aplicación, teniendo como criterios minimizar el consumo de potencia y los retrasos, y garantiza el cumplimiento de las restricciones temporales extremo-extremo en estas aplicaciones.

Capítulo 6: Pruebas y validación. En este capítulo se analizan dos casos de estudio en los que se aplica la metodología propuesta. Los resultados obtenidos resaltan la importancia y la validez de la propuesta realizada.

Capítulo 7: Resultados experimentales. Con el propósito de analizar el desempeño de WSAN soportadas en las tecnologías seleccionadas y las consideraciones de diseño realizadas en esta tesis, en este capítulo se presentan los resultados experimentales de la implementación de un NCS sobre una WSAN.

Capítulo 8: Conclusiones y líneas futuras de investigación. En este capítulo se verifica el cumplimiento de los objetivos y se indican las líneas futuras de investigación que se pueden abordar a partir de los resultados obtenidos en esta tesis.

Capítulo 2

REVISIÓN DEL ESTADO ACTUAL EN REDES INALÁMBRICAS DE SENSORES Y ACTUADORES

La arquitectura de los nodos y de la red que conforman estas aplicaciones, está directamente influenciada por el cumplimiento de las restricciones impuestas por la aplicación. Esto conlleva el análisis de diferentes aspectos que influyen en la realización de un diseño adecuado, como protocolos de redes de comunicaciones inalámbricas, análisis de tiempo real en la ejecución de tareas y en el envío de mensajes, estrategias de ahorro del consumo de potencia, como el escalado dinámico de voltaje, y metodologías de diseño de redes de sensores inalámbricas. En este capítulo se realiza una caracterización de las aplicaciones de WSN, y se analizan los trabajos relacionados en las temáticas mencionadas.

2.1 Aplicaciones de las redes inalámbricas de sensores y actuadores

Los parámetros más relevantes en el diseño de WSN son (Lester, 2003), (Martínez, 2008):

- **Tiempo de vida.** Se espera que la aplicación funcione de forma autónoma por largos periodos de tiempo. El factor que más incide en el tiempo de vida es el consumo de potencia. En términos generales, la actividad que más consume potencia en este tipo de aplicaciones es la comunicación, de allí el interés por la implementación de estrategias que disminuyan el consumo de potencia en dicha actividad.
- **Cobertura.** En algunas aplicaciones se requiere cubrir grandes áreas en espacios abiertos. Para lo cual es importante contar con nodos móviles y estrategias adecuadas para la escalabilidad de la red y la transmisión de información en modo multisalto.
- **Coste reducido.** En algunas aplicaciones se distribuye un elevado número de nodos que al final no se pueden recuperar en su totalidad, de allí la importancia en disponer de nodos de bajo coste.
- **Tiempo de respuesta.** En aplicaciones de control es importante acotar los tiempos de envío de información por cuanto este parámetro puede afectar el desempeño de las estrategias de control. Por otro lado, en aplicaciones de vigilancia y seguridad es importante minimizar el tiempo en que se transmite la información.
- **Sincronización.** Existen aplicaciones en donde es importante correlacionar temporalmente la información suministrada por diferentes nodos, para lo cual es

importante sincronizar la actividad de los nodos y dotar las muestras de una marca temporal.

- Protección de la información. Debido a que el medio de comunicaciones a través del cual se desarrolla la comunicación es inalámbrico, el tema de la autenticidad de información toma un mayor interés.

Uno de los grandes desafíos en el desarrollo de estas aplicaciones se centra en que la optimización del diseño en función de uno de los anteriores factores, puede ocasionar la disminución de la calidad de la aplicación en función de otro factor, con lo cual se debe lograr un compromiso entre los mismos.

Para el desarrollo del proyecto *The Great Duck Island Experiment*, (Li, 2008), se utilizaron nodos Mica2Dot, ejecutando el sistema operativo TinyOS. Los nodos fueron abastecidos por baterías de litio, y el tiempo de vida esperado fue de 100 días aproximadamente, el cual se afectó considerablemente por la sobrecarga del protocolo de comunicaciones multisalto. La movilidad esperada de los nodos fue muy baja. La arquitectura de la red fue jerárquica multinivel y se emplearon técnicas de comunicación de un único salto y multisalto. También se implementaron estrategias para disminuir el tiempo en que el dispositivo de comunicaciones permaneció activo, con el propósito de aumentar la duración de la energía almacenada en los nodos.

En el proyecto SWIM (Li, 2008) se empleó un procesador MSP430. El transmisor de radio fue un Silicon Labs Si4112. El área cubierta fue bastante grande y la movilidad de los nodos fue muy alta. Por cuanto en este proyecto no hubo restricciones en cuanto al tiempo de envío de la información, los nodos almacenaban información hasta detectar que se encontraban dentro del radio de cobertura de un nodo recolector de información, por lo cual fue necesario dotar estos nodos de memoria suficiente para almacenar los datos. Se implementó un esquema de comunicación de un único salto para transmitir la información desde los nodos sensores hasta los nodos recolectores.

Debido a la gran incidencia de los requisitos de las aplicaciones en el diseño de las WSN, en este trabajo se ha acotado el análisis a dominios de aplicaciones civiles, principalmente en los sectores salud, medioambiental, agrícola, e industrial. A continuación se analiza cada una de ellas.

2.1.1 Aplicaciones en el sector salud

El gran interés en el desarrollo de dispositivos económicos y portables, orientados a la monitorización, registro y análisis de variables fisiológicas, para supervisar pacientes continuamente y analizar la presencia o el riesgo de desarrollar alguna enfermedad, se fundamenta en que actualmente estas actividades están bastante limitadas tanto por el tipo de información como por los instantes en que se captura, por lo que las anomalías transitorias no pueden ser detectadas en la totalidad de los casos. Sin embargo, muchas enfermedades, tales como las asociadas a patologías cardiovasculares, se asocian más bien a episodios transitorios que a continuas anomalías. La ocurrencia de estos estados no se puede predecir, por cuanto se descarta su análisis a través de una supervisión controlada. La tendencia es la integración de la tecnología

médica existente con nuevas aplicaciones en WSAN, las cuales aumentarán la cantidad de datos analizados y permitirán respuestas en tiempo real.

Se han realizado varios proyectos orientados al desarrollo de WSN para la detección de problemas de salud. Los signos vitales que se monitorizan más comúnmente son el ECG, oximetría de pulso, temperatura corporal, frecuencia cardíaca y la presión arterial. Algunos de estos proyectos son (Alemdar, 2010):

- UbiMon (*Ubiquitous Monitoring Environment for Wearable and Implantable Sensors*). Tiene por objeto abordar tópicos generales relacionados con el uso de sensores portables de medidas no invasivas, que capturen eventos transitorios que amenacen la vida de las personas y faciliten la intervención.
- SAPHE (*Smart and Aware Pervasive Healthcare Environments*). Este proyecto tiene como objetivo desarrollar una plataforma inteligente, de supervisión no invasiva, para el análisis de tendencias de datos a largo plazo.

Uno de los principales objetivos de SAPHE es asegurar que los pacientes con enfermedades crónicas sean tratados en un ámbito comunitario o con el apoyo del hogar, siempre que sea posible, lo que permite reducir la demanda de asistencia sanitaria y libera capacidad, al mismo tiempo que posibilita la atención de más pacientes y permite un uso más eficaz de los recursos, además de ofrecer una atención más personalizada y mejorar la calidad de vida de los pacientes.

SAPHE busca abordar la necesidad de una nueva generación de sistemas de vigilancia de la salud, que permitan la detección temprana del deterioro de la salud mediante la identificación de cambios fisiológicos y metabólicos a lo largo del tiempo.

- MobiHealth es uno de los primeros proyectos que integra todos los dispositivos sensores portátiles que una persona lleva consigo durante el día, como teléfonos móviles y relojes. Los sensores miden continuamente y transmiten los datos fisiológicos a los proveedores de servicios de salud a fin de proporcionar una asistencia remota, rápida y fiable en caso de accidente. Este proyecto es uno de los primeros estudios que propone la convergencia de los sistemas de redes diferentes como BAN (*Body Area Network*), PAN (*Personal Area Network*) y WAN (*Wide Area Network*), para permitir un servicio de cuidado de la salud móvil y personalizado.
- CodeBlue: *Wireless Sensor Networks for Medical Care, Remote Arrhythmia Monitoring System Developed*, es una plataforma de hardware y software desarrollado en la Universidad de Harvard. La parte de diseño de hardware incluye el diseño y desarrollo de un oxímetro de pulso, ECG de dos vías, y un tablero sensor para el análisis de movimiento. La arquitectura de software se basa en un modelo *publish/subscribe*. CodeBlue tiene por objetivo proporcionar la coordinación y comunicación entre los dispositivos médicos inalámbricos de manera *ad hoc*. Adicionalmente se utiliza un protocolo de detección para que los nodos descubran las capacidades de sensado que poseen cada uno. También se ha

integrado un sistema de localización en los nodos, denominado MoteTrack, el cual es utilizado para localizar los pacientes y los profesionales del cuidado de la salud.

- AlarmNet es una red inalámbrica de sensores médicos compuesta de cinco componentes. La red de área corporal es responsable de monitorizar variables fisiológicas y de la ubicación del paciente, esta incluye sensores de frecuencia cardíaca, saturación de oxígeno y ECG desarrollados en el proyecto CodeBlue. La red de sensores fijos proporciona información de contexto espacial y del medio ambiente como la temperatura, el movimiento y la humedad.
- LifeGuard, fue desarrollado inicialmente para los astronautas, puede ser utilizado para monitorizar signos vitales. El sistema se compone de tres componentes. La parte de sensores puede soportar diferentes tipos de sensores tales como ECG, frecuencia respiratoria, oxímetro y presión arterial. Los datos de los sensores son recogidos y almacenados por un dispositivo portátil. Cuenta con acelerómetros de 3 ejes y sensores de temperatura de la piel.
- FireLine está diseñado para supervisar las medidas cardíacas de los bomberos para poder tomar las acciones necesarias en el caso de presentar alguna anomalía. El dispositivo se compone de un sensor inalámbrico, un sensor de ritmo cardíaco y tres electrodos.
- Baby Glove, consiste de dos sensores colocados en la parte superior del torso del bebé, que contienen un sensor de temperatura y electrodos para monitorizar la frecuencia del pulso del bebé y la hidratación.

Otros proyectos relacionados son: HUMAN++, HealthService 24, WsHC - *Wireless Health and Care*, *Fraunhofer Body Area Network*, MobiHealth, Wealthy, HEARTS, MyHeart, AUBADE: *A wearable EMG augmentation system for robust emotional understanding*.

La primera generación de redes de sensores inalámbricos para la asistencia sanitaria ha demostrado su potencial en el desarrollo de sistemas de monitorización continua y alarma temprana. La tendencia de este tipo de aplicaciones se orienta a garantizar la integridad y la privacidad de la información, y la capacidad de desplegar sistemas de gran escala que cumplen los requisitos de las aplicaciones, utilizando dispositivos con recursos reducidos y de bajo consumo (Ko 2010).

2.1.2 Aplicaciones en el sector medioambiental

En los últimos años se ha presentado un gran interés por el desarrollo de técnicas y estrategias que permitan disminuir el impacto medioambiental. La conservación de las fuentes de aguas o cuencas hidrográficas, la optimización del uso del agua, el control biológico de plagas, el manejo racional de los agroquímicos, el control de emisiones atmosféricas, etc., son temas de gran impacto en nuestra sociedad.

Las WSN permiten el desarrollo de aplicaciones orientadas a la continua monitorización de variables medioambientales en grandes extensiones de terrenos, lo

que facilita actividades como la caracterización de cuencas hidrográficas, la detección de contaminantes, el análisis de condiciones microclimáticas, el estudio de factores de riesgo para especies en peligro, la monitorización de especies en vía de extinción, sistemas de detección, monitorización y control de incendios forestales, etc.

Algunos de los proyectos medioambientales abordados con WSN son:

- *The Great Duck Island Experiment*. Esta red de sensores, compuesta por más de 150 nodos, fue diseñada para sensar, de forma no invasiva, la ocupación de los nidos de una especie de pato y la incidencia de los microclimas en su hábitat. En la figura 2.1 se aprecian 5 tipos de dispositivos. Los biólogos ponen los nodos sensores en los nidos subterráneos (1) y sobre pilotes de 4 pulgadas colocados justo fuera de los nidos (2). Estos dispositivos registran los datos y los envían a un nodo *gateway* (3), que transmite la información a una computadora portátil en la estación de investigación (4), luego a través de una antena parabólica (5) la información es transmitida a un laboratorio en California.



Figura 2.1. Imágenes del proyecto *The Great Duck Island*. (Tomado de: Martha Baer, *The Ultimate on-the-fly network*. Wired Magazine, 11.12, Dec. 2003).

- *The Shared Wireless Infostation Model (SWIM)*. Por cuanto actualmente las ballenas y grandes mamíferos marinos se encuentran en peligro de extinción, éste proyecto fue desarrollado para estudiar el impacto de los cambios medioambientales naturales y aquellos provocados por el hombre, en el comportamiento y movilidad de estas especies.
- *The ZebraNet Wildlife Tracker*. Este proyecto se desarrolló con el objetivo de reunir datos sobre una amplia gama de especies de animales terrestres, con el fin de comprender sus interacciones y la influencia de una sobre otras. El objetivo era también comprender los patrones de la migración de esos animales salvajes, y la forma en que pueden verse afectados por cambios en los patrones climáticos y otros provocados por el hombre.

Otros proyectos relacionados son: WaterNet, Smart Mesh Weather Forecasting, GlacsWeb, Wireless Sensors for Wildfire Monitoring.

2.1.3 Aplicaciones en el sector agrícola

Las WSN pueden contribuir a analizar el estado de un cultivo por medio de la monitorización y el análisis regular de las condiciones fisiológicas de las plantas en sitios específicos, como también vigilar el estado del suelo obteniendo mediciones sobre propiedades químicas (cantidad de nutrientes, presencia de minerales como el nitrógeno, fósforo, potasio, calcio, azufre, magnesio, hierro, cobre o aluminio, niveles de pH para medir acidez o basicidad, etc.), propiedades físicas (porosidad del suelo, densidad), o su capacidad de campo (disponibilidad de agua, pérdidas por escorrentía e infiltración, etc.).

De igual forma, el uso de WSN facilita el análisis de la biota, es decir, todo el conjunto de especies de plantas, animales y otros organismos encontrados en el cultivo. Adicionalmente, se puede recopilar información sobre diversas variables climáticas, como temperatura, precipitación, humedad, horas luz, niveles de radiación, viento y evaporación.

Esta información posibilita el desarrollo de pronósticos sobre la presencia de plagas y el rendimiento del cultivo. Es este tipo de disponibilidad de información para sitios particulares en instantes de tiempo específicos, lo que hace posible evaluar densidades óptimas de siembra, estimar cantidad óptima de fertilizante y predecir la producción en cada punto del cultivo. A esto se le conoce como agricultura de precisión, es decir, la posibilidad de observar, medir y controlar las prácticas agrícolas con diferenciación entre distintas partes de un mismo cultivo.

En el proyecto LOFAR-Agro, figura 2.2, se desarrolló una WSN de aproximadamente 100 nodos para una aplicación de agricultura de precisión. Por medio de la medición de humedad y temperatura, se pretendía detectar el momento propicio para combatir las fitóftoras en los cultivos de papa. Los resultados experimentales mostraron que el crecimiento del cultivo reduce drásticamente el radio de cobertura de los dispositivos de comunicaciones y afecta el desempeño de la aplicación (Langendoen, 2006).



Figura 2.2. Imagen de uno de los nodos utilizados en el proyecto LOFAR-Agro

En (López, 2009) se describe una aplicación de WSN para el cultivo de repollos en la región semiárida de Murcia, la cual permite medir las diferentes características del suelo, tales como la temperatura, la humedad volumétrica y la salinidad. El sistema permite controlar con éxito el crecimiento de repollos ecológicos con la precisión requerida.

Otros proyectos relacionados son: GoodFood.

2.1.4 Aplicaciones en el sector industrial

A nivel industrial, las WSN se orientan a la monitorización, registro y control de variables. En el año 2007 surgió un nuevo estándar, WirelessHART, el cual se ha orientado a suplir las necesidades de la industria de procesos, en la figura 2.3 se presenta la arquitectura de una aplicación industrial que utiliza este estándar.

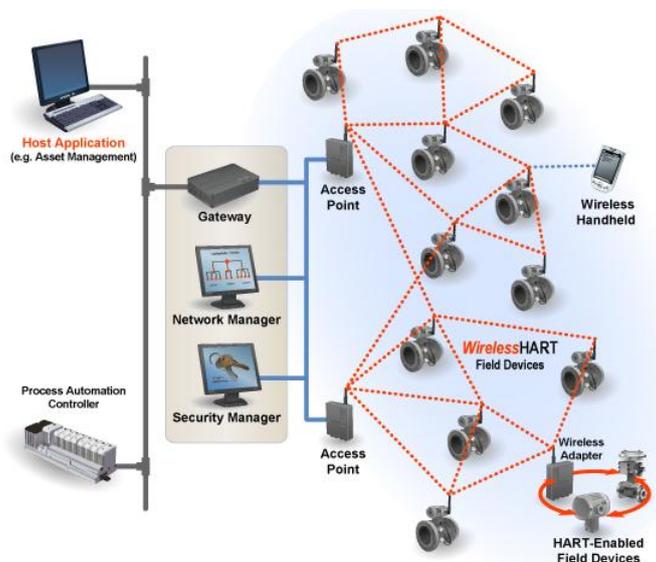


Figura 2.3. Arquitectura de una aplicación industrial que utiliza WirelessHART (tomado de <http://www.hartcomm.org>)

Las ventajas que ofrecen estos sistemas repercuten en mejoras en la calidad de los productos y disminución de costes de producción y mantenimiento. Además, la utilización de redes inalámbricas facilita la instalación y adaptación del sistema y reduce considerablemente los costes de implementación. El desarrollo de nuevas aplicaciones sobre WSN abre un nuevo panorama en el que se integran redes alámbricas e inalámbricas para aumentar la flexibilidad y confiabilidad de las aplicaciones.

2.1.5 Clasificación de las aplicaciones

En esta sección se presentan dos clasificaciones de las aplicaciones de redes de sensores y actuadores inalámbricas en los sectores descritos, cuyo objetivo fue la organización de los requisitos derivados de las aplicaciones para la toma de decisiones en el proceso de diseño y desarrollo de WSN.

En la primera aproximación a la clasificación se consideraron los siguientes requisitos de las aplicaciones: cómputo requerido, tamaño de memoria, ancho de banda, ubicación de los nodos, movilidad de los nodos, restricciones de tiempo real, área de cobertura, autonomía energética y sincronización. La clasificación realizada fue la siguiente (Martínez, 2008):

- Aplicaciones tipo 1. Se caracterizan porque los periodos de muestreo de las variables oscilan entre un segundo y algunas horas, y los plazos de tiempo para la generación de los resultados de los algoritmos no son estrictos. Adicionalmente, estas aplicaciones se desarrollan en espacios abiertos y deben cubrir grandes extensiones. A esta categoría pertenecen aplicaciones agrícolas y medioambientales orientadas principalmente a medir, registrar y analizar variables medioambientales.

Los algoritmos empleados en los nodos para procesar las señales son principalmente filtros digitales de bajo orden, y extracción de valores medios y máximos entre muestras. Dichos valores son enviados a una estación central en donde, utilizando técnicas de fusión de datos, se aplican algoritmos de estimación de valores entre puntos separados espacialmente y se toman algunas acciones como por ejemplo la apertura de bombas de agua o la aplicación de agroquímicos y fertilizantes en cultivos.

Es necesaria la sincronización de las mediciones de diferentes nodos. La autonomía energética esperada en cada nodo oscila entre días y meses; en algunas aplicaciones, en sitios sin acceso a fuentes de energía convencionales, se utilizan sistemas de energía alternativos basados principalmente en fotoceldas solares, los cuales reabastecen las baterías de los nodos.

- Aplicaciones tipo 2. Los algoritmos empleados en estas aplicaciones se desarrollan en espacios cerrados y demandan una mayor capacidad de cómputo que en las aplicaciones tipo 1, aunque al igual que en estas últimas no existen plazos estrictos de respuesta. Al tratarse de espacios cerrados y nodos fijos, se disminuyen las restricciones de consumo de potencia puesto que se pueden utilizar fuentes de energía convencionales, sin embargo se justifica el uso de redes inalámbricas por cuanto se facilita la instalación, adaptabilidad y portabilidad de la aplicación, además de disminuir los costes de implementación.

Este tipo de aplicaciones se encuentran principalmente en los sectores industrial y agrícola, y se orientan a la medición y registro de variables, diagnóstico de fallos en línea de plantas y procesos industriales, detección de plagas y control de microclimas en invernaderos. Los periodos de muestreo van desde un milisegundo, para la captura de medidas requeridas para la ejecución de algoritmos de diagnóstico, hasta un segundo, para la monitorización de algunas variables. En estas aplicaciones los algoritmos de diagnóstico no requieren un funcionamiento continuo, por tanto las muestras pueden ser almacenadas para luego ser procesadas.

- Aplicaciones tipo 3. En esta categoría se agrupan las aplicaciones en las que, además de medir y procesar variables con requisitos similares a las aplicaciones tipo 2, se requiere procesar imágenes y la aplicación se desarrolla en espacios abiertos, algunas de estas aplicaciones se encuentran en el sector agrícola destinadas a la detección de plagas, y en el sector medioambiental orientadas a la detección de incendios.

Algunos de los nodos se encuentran en movimiento, y debido a que se requieren autonomías energéticas de algunas horas, las restricciones en cuanto al consumo de

potencia son grandes. También es necesaria la sincronización de los nodos. Debido a los algoritmos empleados para el procesamiento de imágenes, los requerimientos de cómputo, cantidad de memoria y ancho de banda en la comunicación son más exigentes que en las aplicaciones tipo 2.

- Aplicaciones tipo 4. Estas se diferencian de las aplicaciones tipo 3 en que se desarrollan en espacios cerrados, con lo cual la cobertura de la red no es muy grande; las autonomías energéticas requeridas también son mayores, llegando a ser cercanas a una semana. En esta categoría se agrupan las aplicaciones médicas orientadas a la detección de enfermedades.

Dependiendo del número de pacientes que se desea monitorizar en una misma red, en este tipo de aplicaciones es recomendable el tratamiento de las muestras en el nodo, debido a que algunas variables requieren un muestreo elevado, como la presión arterial 100 muestras/segundo, el ECG 240 muestras/segundo y el EEG 500 muestras/segundo. De tal forma que solo se transmita un segmento de las medidas cuándo se detecte algún problema.

- Aplicaciones tipo 5. En este tipo de aplicaciones se debe enviar una muestra cada periodo de muestreo, por esta razón los periodos de muestreo están limitados por la separación mínima entre las tramas de datos de los protocolos de comunicaciones.

En esta categoría se encuentran las aplicaciones de control de procesos industriales, las cuales se desarrollan en espacios cerrados, por lo que la distancia entre los nodos no es muy grande, y las restricciones respecto del consumo energético no son muy exigentes. Los plazos para la generación de los resultados responden a una planificabilidad extremo-extremo; su cumplimiento ha de ser estricto, por cuanto el no cumplimiento de los mismos puede degradar considerablemente el desempeño del sistema de control, generando incluso inestabilidad en el sistema, para lo cual es importante la sincronización de las actividades de los nodos que conforman el lazo de control. Adicionalmente, para mejorar el desempeño de los lazos de control, es importante limitar la variabilidad en el jitter de las tareas de actuación.

La tabla 2.1 presenta un resumen del análisis de los requisitos para cada tipo de aplicación.

Tabla 2.1. Análisis de los requisitos para cada tipo de aplicación

Aplicación	Cómputo requerido	Tamaño de Memoria	Ancho de banda	Ubicación	Movilidad de los nodos	Tiempo real	Espacio	Autonomía energética	Sincronización
Tipo 1	Bajas prestaciones	Bajo	< 256 kbps	Si	No	Sólo en la medición	Abierto 10 km	Meses	Si
Tipo 2	Bajas prestaciones	Medio	< 256 kbps	Si	No	Sólo en la medición	Cerrado 100 m	No hay restricción	Si
Tipo 3	Altas prestaciones	Alto	1 Mbps	Si	Si	Sólo en la medición	Abierto 10 km	Horas	Si
Tipo 4	Altas prestaciones	Medio	< 256 kbps	Si	Si	Sólo en la medición	Cerrado 1 km	Días	Si
Tipo 5	Bajas prestaciones	Bajo	< 256 kbps	No	No	Extremo-Extremo y Mínima variabilidad de jitter	Cerrado 100m	No hay restricción	Si

Se puede apreciar que desde el punto de vista tecnológico existen características similares entre los tipos de aplicaciones presentados, por tal razón se realizó una nueva clasificación considerando solamente dos clases de aplicaciones, para lo cual se utilizaron como criterios de agrupación el área de cobertura, las restricciones de tiempo real y la autonomía energética. La segunda clasificación de las aplicaciones fue la siguiente:

- Aplicaciones clase A. Estas aplicaciones se desarrollan en espacios abiertos y deben cubrir grandes extensiones, razón por la cual algunos de los nodos que las constituyen pueden ser móviles, también demandan un tiempo de vida elevado y sus funciones requieren principalmente un retraso mínimo mas que acotado; de allí la importancia de contar con protocolos de comunicación escalables y dotar de técnicas de ahorro de energía a todos los nodos que conforman la red. En esta clase se agrupan las aplicaciones tipo 1 y 3 de la primera clasificación.
- Aplicaciones clase B. Los algoritmos empleados en estas aplicaciones se desarrollan en espacios cerrados. Al igual que en las aplicaciones clase A, algunas de las funciones requieren un retraso mínimo, sin embargo también existen funciones que requieren cumplir con plazos estrictos de respuesta. Al tratarse de espacios cerrados, las restricciones en cuanto al escalado de la red y el consumo de potencia en algunos nodos son menores que en las aplicaciones clase A. En esta clase se agrupan las aplicaciones tipo 2, 4 y 5 de la primera clasificación.

2.2 Arquitectura de los nodos de WSN

Las arquitecturas de los nodos dependen altamente de la aplicación a la cual están dirigidos y de la función que cumpla cada uno dentro de la misma. Los nodos actuales se caracterizan por estar dotados de coprocesadores, los cuales permiten acelerar el procesamiento de ciertas funciones, tales como la comunicación y la protección de la información, al mismo tiempo que se logra un mejor balance entre el consumo de potencia y la eficiencia del sistema.

En el mercado se encuentran nodos sensores con procesadores de 8, 16 y 32 bits, dotados de interfaces para intercambiar información con el medio en el que se encuentran, para lo cual utilizan una gran variedad de sensores y actuadores, entre ellos de temperatura, luz, relés, presión barométrica, GPS, micrófono, fotorresistencia, termistores, aceleración, etc. En la tabla 2.2 se presentan las características de los nodos sensores mas reportados en las aplicaciones.

Actualmente algunos fabricantes como Microchip, Texas Instruments, Freescale y Atmel, ofrecen un conjunto amplio de dispositivos para abordar aplicaciones de WSN, entre ellos, procesadores de bajo consumo de potencia, transceptores y sistemas en chip, los cuales tienen interfaces asociadas para implementar los protocolos IEEE 802.15.4 y Zigbee, acceder a los dispositivos y gestionar tareas.

El parámetro más relevante dentro de las características de las WSN es el consumo de potencia. Debido a esto, en la selección de un procesador para este tipo de aplicaciones es necesario considerar el consumo de potencia tanto por sus características físicas

(voltajes y corrientes requeridos en operación y en modos de bajo consumo), como de programación (modos de bajo consumo SLEEP/HALT/IDLE). Adicionalmente, la posibilidad de emplear técnicas como *Dynamic Voltage Scaling* (DVS) contribuye significativamente al ahorro del consumo de potencia.

Los desarrollos se han orientado a la integración en un mismo chip de núcleos de procesadores existentes junto con coprocesadores e interfaces especializadas. Logrando, además de un mejor balance en cuanto al consumo de potencia y el tiempo de procesamiento, la disminución del coste, tamaño y tiempo de desarrollo de las aplicaciones. Tal es el caso de los dispositivos CC2431, CC2480 y CC2530 de Texas Instruments, el ATmega128RF de Atmel y el MRF24J40 de Microchip.

En lo que respecta a la protección de información, las restricciones de estos dispositivos en cuanto a la limitada capacidad de cómputo, bajo consumo de potencia y poca capacidad de almacenamiento, restringen la implementación de algoritmos complejos de seguridad. En IEEE 802.15.4 se establece un mecanismo de cifrado soportado en claves simétricas, las cuales son provistas por los niveles superiores (IEEE 802.15.4, 2006). Actualmente ya se han incorporado módulos hardware en los procesadores que facilitan el desarrollo de dichas funciones. Tal es el caso del coprocesador para implementar el *Advanced Encryption Standard* (AES), integrado en los CC2430, CC2431, CC2480 y CC2530 de Texas Instruments.

Otro aspecto importante es la sincronización de los nodos. La técnica que se emplee para sincronizar los nodos dependerá de la aplicación para la cual esté orientada la WSN. El objetivo de cualquier técnica de sincronización de tiempo es mantener un tiempo similar dentro de una cierta tolerancia entre los nodos que conforman la red. Efectos como tiempos de acceso para el envío de mensajes de sincronización y las derivas en los relojes locales de los nodos, pueden afectar de manera significativa la sincronización de toda la red. El protocolo NTP posee una carga computacional muy elevada y no tiene en cuenta estrategias de ahorro de energía, lo cual no lo hace adecuado para WSN. Además, su esquema de sincronización estableciendo una arquitectura jerárquica, puede ser afectado por la presencia de fallos en nodos servidores de tiempo intermedios dentro de la jerarquía y por el movimiento de los nodos. En (Mangharam, 2007) se implementó en cada nodo un módulo dedicado a la sincronización de tiempo por medio de un receptor de radio AM, con lo cual se logró una sincronización de tiempo global y robusta, pero de esta forma se incrementa el coste y el consumo de potencia de los nodos.

En el CC2431 también se ha incorporado un módulo especializado para abordar la problemática de la localización de nodos con base en algoritmos de triangulación.

Tabla. 2.2. Características de los nodos sensores

Características	Mica2	MicaZ	TelosB	Imote 2
				
CPU	Atmega128 8bits / 8MHz	Atmega128 8bits / 8MHz	MSP430 16bits / 8MHz	PXA271 32bits / 13(104)MHz
SRAM [kB]	4	4	10	256 - 32,000
FLASH [kB]	128 + 512	128 + 512	48 / 1024	32,000
Radio	CC1000, 916 MHz,	IEEE 802.15.4, CC2420, 2.4GHz	IEEE 802.15.4, CC2420, 2.4GHz	IEEE 802.15.4, CC2420, 2.4GHz (Bluetooth/IEEE 802.11)
Ancho de banda	38 Kbps	250 Kbps	250 Kbps	250 Kbps (720/11,000)
Consumo de Potencia Procesador/Rx/Tx	8 / 10 / 27 [mA]	8 / 20 / 18 [mA]	1 / 20 / 18 [mA]	40 / 20 / 18 [mA]
Consumo en modo SLEEP	19 [µA]	27 [µA]	6 [µA]	390 [µA]
Sistema Operativo	TinyOS	TinyOS	TinyOS	TinyOS
Interfaces	6 canales AD de 10 bits. Las interfaces en estos nodos son diseñadas en placas independientes	8 canales AD de 10 bits. Las interfaces en estos nodos son diseñadas en placas independientes	8 canales AD de 12 bits. 2 canales DA de 12 bits. Sensores de Luz, IR, Humedad, Temperatura. Interface USB.	I ² S, AC97, Camera Chip Interface, GPIOs, SDIO, USB.

2.3 Protocolos de comunicaciones para WSN

Aunque estas redes comparten algunas características con las redes inalámbricas tradicionales, en general ellas se diferencian en que:

- La conservación de energía es un factor prioritario en el diseño de protocolos para WSN.
- La topología de la red puede cambiar dinámicamente debido al movimiento de los nodos, por lo que estos podrían no tener un único identificador.
- Se pueden presentar casos en los que una WSN esté compuesta de un elevado número de nodos.
- Los datos generados en los nodos pueden estar relacionados con los datos de los nodos vecinos, lo cual puede ser aprovechado como estrategia para disminuir el consumo de potencia en la red, (Swami, 2007).

Actualmente existen diversos dispositivos comerciales que utilizan ISA100.11, 6LoWPAN, WirelessHART (WirelessHART) y Zigbee (Zigbee Specification), junto con sus definiciones para aplicaciones específicas: ZigBee Smart Energy, ZigBee Building Automation, ZigBee Home Automation, ZigBee PRO, ZigBee RF4CE, ZigBee Health Care. Los cuales presentan un menor consumo de potencia que los desarrollos soportados en IEEE 802.11.

ISA100.11a fue diseñado para aplicaciones de automatización y monitorización en aplicaciones de bajo flujo de datos. Ese estándar se enfoca en un bajo consumo de potencia, escalabilidad, infraestructura, robustez e interoperabilidad con otros dispositivos inalámbricos. Utiliza la banda de 2,4 GHz y realiza saltos en frecuencia entre canales para aumentar la fiabilidad y reducir al mínimo la interferencia. Ofrece dos tipos de topologías, estrella y *mesh*.

6lowpan es un acrónimo de “*Ipv6 over Low power Wireless Personal Area Networks*”, este grupo ha definido los mecanismos que permiten enviar paquetes Ipv6 a través de redes IEEE 802.15.4.

WirelessHART se basa en la capa física del estándar IEEE 802.15.4-2006, pero especifica nuevos niveles de Enlace de datos, Red, Transporte y Aplicación, (Lennvall, 2008). WirelessHART emplea un modo de acceso al medio soportado en TDMA, con 100 ranuras por segundo, con lo cual se evitan las colisiones y se pueden emplear estrategias para disminuir el consumo de potencia. Adicionalmente, WirelessHART forma redes con topologías *mesh* suministrando caminos de redundancia los cuales permiten encaminar los mensajes por diferentes rutas, con lo cual se puede bordear obstáculos físicos y evitar enlaces interrumpidos e interferencias.

Zigbee está soportado en el estándar IEEE 802.15.4-2003 y utiliza un modo de acceso sin sincronización para establecer arquitecturas de redes *mesh*. Las nuevas características de Zigbee son las siguientes:

- ZigBee Smart Energy ofrece servicios para la gestión de dispositivos relacionados con el consumo de potencia en hogares y edificios a través de *Home Area Networks* (HAN). La versión 2.0 soportará direccionamiento IP y se podrá implementar en los niveles de enlace IEEE 802.15.4-2006, IEEE 802.11 e IEEE 802.3.
- ZigBee Building Automation y ZigBee Home Automation definen características para el control de electrodomésticos, iluminación, el medio ambiente, gestión energética, vigilancia y seguridad, para la automatización de pequeñas oficinas y el hogar.
- ZigBee PRO incorpora características de Zigbee definidas en 2006, las cuales facilitan el desarrollo de redes con un gran número de dispositivos.
- ZigBee RF4CE define nuevas características en IEEE 802.15.4, aportando una capa de red simple y perfiles de aplicación estándar, que se pueden utilizar para crear soluciones para hogares que integran dispositivos de varios fabricantes.
- ZigBee Health Care brinda características para el desarrollo de dispositivos inalámbricos, orientados a la monitorización y gestión de enfermedades crónicas no críticas, de manera fiable y segura.

Uno de los principales desafíos de investigación relacionados con la mejora del desempeño de la red es el desarrollo de estrategias de cooperación entre niveles de la red (*cross-layer*). Por ejemplo, un protocolo MAC comparte información de la topología con el protocolo de red para ayudar en la configuración de rutas y el mantenimiento de las mismas. Se han desarrollado propuestas enfocadas a los niveles físico, enlace de datos, red y transporte. La investigación futura puede centrarse en la colaboración entre todos los niveles de la red para lograr un mayor ahorro de energía, mejorar el rendimiento de la red, y extender la vida útil de la red, (Yick, 2008).

2.3.1 Protocolos de control de acceso al medio (MAC)

Los principales requisitos de una WSN se relacionan con la eficiencia energética, escalabilidad, y en algunas aplicaciones se requiere una latencia mínima y/o acotada. Estos requisitos se alcanzan en mayor o menor medida dependiendo del tipo de protocolo MAC seleccionado, los cuales se pueden clasificar en protocolos planificados, generalmente basados en acceso múltiple por división de tiempo (TDMA), y protocolos basados en contención.

Los protocolos que utilizan TDMA requieren de un maestro que distribuye el orden en que los nodos accederán al medio, lo cual se realiza al inicio de la aplicación, disminuyendo así la sobrecarga en cuanto a paquetes de control. Además, este tipo de protocolos son libres de colisión y presentan un muy buen desempeño en redes con distancia entre nodos acotada a un único salto. Sin embargo, requieren una sincronización estricta, y una vez se realiza la inicialización el orden de acceso al medio no puede ser modificado, lo cual dificulta el acceso de nuevos nodos a la red; aunque

éste último inconveniente puede ser resuelto realizando inicializaciones frecuentes. Esto último fue implementado en el protocolo *Low-Energy Adaptive Clustering Hierarchy* (LEACH), intercambiando cada vez la función de nodo maestro entre los diferentes nodos de la red, logrando igualmente un gasto equitativo en el consumo de potencia de los nodos. No obstante, en LEACH se presentan problemas de sincronismo debido a colisiones en el medio entre grupos de nodos vecinos.

En (Hristu-Varsakelis, 2005) y (Lee, 2008) se realiza un análisis de la utilización de Bluetooth e IEEE 802.11b como redes para control, al mismo tiempo que se presenta una arquitectura de cuatro niveles para lograr un comportamiento predecible en IEEE 802.11b, debido a que este protocolo emplea un acceso al medio por contención. Bluetooth está muy restringido tanto por el rango de transmisión, aproximadamente 10 m, como por el número de nodos que se pueden conectar a una *Piconet*, un maestro y siete esclavos. La desventaja que presentan estos protocolos respecto a IEEE 802.15.4, es que su elevado consumo de potencia disminuye drásticamente el tiempo de vida de la aplicación.

En (Pantazis, 2007) se presentan diferentes métodos para disminuir el consumo de potencia en diferentes niveles de la pila de comunicaciones de redes de sensores inalámbricas, entre los cuales se analiza el modo de acceso al medio TDMA, el cual contribuye a garantizar tiempos máximos de envío de mensajes en la red, minimizar las colisiones y al ahorro de energía en los nodos.

Los resultados presentados en (Moraes, 2008) y (Cena, 2008) permiten concluir que el modo EDCA (*Enhanced Distributed Channel Access*) del 802.11e ofrece una buena alternativa para los requisitos de tiempo real en las aplicaciones industriales. Se concluye que para los escenarios presentados, el uso de los valores por defecto establecidos en el EDCA permite garantizar los requisitos en el envío de información para aplicaciones industriales, donde el número de estaciones sea menor que 10 y con periodos entre mensajes por encima de 10 ms.

Una gran ventaja de los protocolos TDMA es que facilitan la implementación de técnicas de ahorro de energía implementando periodos de apagado de los transceptores en los nodos. En (Li, 2008) se presenta una comparación de la potencia consumida en los diferentes modos de operación entre el RFM TR1000 utilizado en los nodos MICA, el Chipcon CC1000 utilizado en los nodos MICA2, y el Chipcon CC2420 utilizado en los nodos MICAz y Telos, cuyos resultados se presentan en la figura 2.4 Se puede apreciar como el consumo de potencia en modo *idle* es comparable al consumo en los modos de transmisión y recepción, y que el mayor ahorro de potencia se logra cuando se apaga el transceptor.

Los protocolos basados en contención emplean generalmente el mecanismo de acceso múltiple por sentido de portadora (CSMA), en este tipo de protocolos se facilita el ingreso de nuevos nodos a la red, lo cual contribuye a la escalabilidad de las aplicaciones. Igualmente se facilita la implementación de técnicas multsalto con relación a los protocolos TDMA.

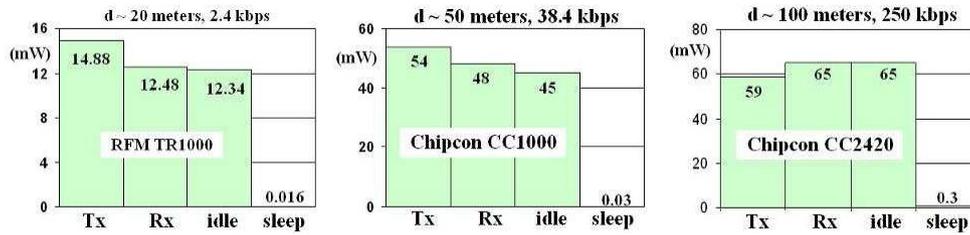


Figura 2.4. Diagrama comparativo del consumo de potencia en los diferentes modos de operación del TR1000, CC1000 y CC2420, (Li, 2008)

Algunos de los protocolos basados en contención más representativos son:

- Sensor-MAC (SMAC). Considera la eficiencia energética como el factor más importante, para ello apaga el transceptor periódicamente, lo que aumenta la latencia del sistema pero disminuye el *Duty cycle*, el cual se define como:

$$Duty_cycle = \frac{Tiempo_escuchar_medio}{Periodo}$$

SMAC introduce un tipo de paquete SYNC para realizar la tarea de sincronización. Durante el tiempo en que escuchan el medio los nodos esperan recibir paquetes SYNC, los cuales indican los instantes en que los nodos accederán al medio, posteriormente apagan el transceptor hasta su instante de activación. Para garantizar la conexión entre grupos vecinos, los nodos que reciben múltiples itinerarios de activación deben adoptar un itinerario igual a la unión entre los diferentes planes recibidos. La desventaja radica en que el itinerario debe ser establecido previamente y no puede ser ajustado para diferentes cargas de tráfico.

- Timeout-MAC (TMAC). Este protocolo intenta mejorar el ahorro de energía en el SMAC reduciendo el tiempo de escuchar el medio. Estas estrategias de ahorro de energía generan grandes retrasos en la comunicación cuando el receptor entra en un ciclo de apagado antes de que el paquete se transmita.
- Existen otros tipos de protocolos que buscan mejorar el consumo de potencia de SMAC adaptando dinámicamente el tiempo de escuchar el medio, o planteando arquitecturas de red jerárquicas y disminuyendo mensajes de control, lo cual solo presenta mejores desempeños que SMAC para casos específicos, tal es el caso de Data-gathering MAC (DMAC).

Existen otras propuestas de protocolos MAC que consideran la correlación de datos en ciertas aplicaciones, para reducir la redundancia en las transmisiones hechas por los sensores, logrando de esta forma minimizar la energía total de la red y el ancho de banda empleado. En este sentido se han realizado tres aproximaciones principales:

- Agregación de datos. Busca eliminar la redundancia de los mensajes, procesando conjuntamente los datos recibidos y los muestreados localmente.
- Codificación de fuentes distribuidas. Cada sensor realiza una compresión de datos localmente, pero logrando un porcentaje de compresión igual a como si se realizara una compresión centralizada.
- Muestreo espacial. Se emplea en los casos en que una red densa de nodos monitoriza un área, por lo que sensores en la misma vecindad pueden capturar la misma información; así que el protocolo MAC permite que solo un subconjunto de los nodos acceda al canal, y evita de esta manera emplear algunos recursos en el envío y procesamiento de información redundante.

Los requisitos de las aplicaciones en WSN han motivado a los diseñadores a utilizar estándares como Bluetooth (IEEE 802.15.1), IEEE 802.15.4 e IEEE 802.11.x. IEEE 802.11 y Bluetooth son adecuados para el envío de grandes cantidades de datos, aunque presentan un consumo de potencia elevado en comparación con IEEE 802.15.4.

2.3.1.1 Estándar de comunicación IEEE 802.15.4

Las características generales, presentadas, en el estándar de comunicación IEEE 802.15.4, son:

- Tasas de transferencias de 250Kb/S, 40Kb/S y 20Kb/S.
- Manejo de redes en estrella y malla (*peer-to-peer*).
- Direccionamiento corto con 16 bits y extendido con 64 bits.
- Ranuras de tiempo garantizado (GTS, *guaranteed time slot*).
- Detección de los niveles de energía recibidos (ED, *energy detection*).
- Indicadores de calidad en el enlace (LQI, *link quality indication*).
- Acceso al canal a través de CSMA-CA.

Desde el punto de vista de las capacidades de cómputo se establecen dos tipos de dispositivos que participan en la red, dispositivos con todas las funciones (FFD, *Full Function Device*) y dispositivos con funciones reducidas (RFD, *Reduced Function Device*). Desde el punto de vista de la función que realizan se clasifican en coordinador, enrutador y dispositivo final.

En una red existe solamente un coordinador, el cual se encarga de iniciar la red y debe ser implementado sobre un FFD, este también puede cumplir con funciones de enrutador. Los dispositivos simplemente cumplen con alguna función específica de la aplicación, pero no participan en la vinculación de nuevos nodos a la red o en el encaminamiento de información.

Dependiendo de la aplicación el estándar permite operar en una de dos topologías, la red en estrella (*Star*), figura 2.5, o la *peer-to-peer*, figura 2.6.

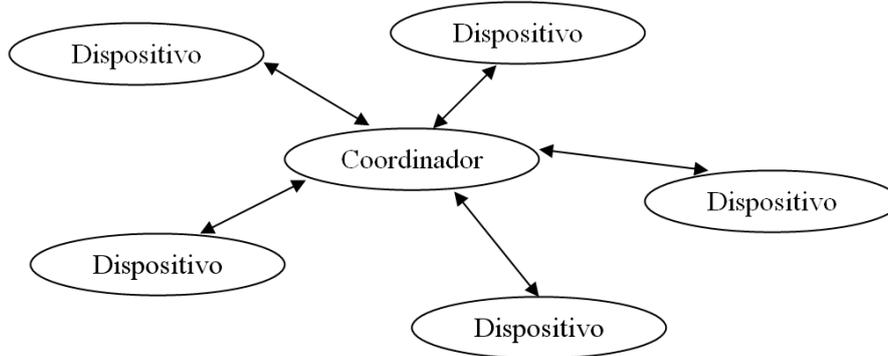


Figura 2.5. Topología Estrella

En la topología estrella la comunicación es establecida entre los dispositivos y un único coordinador de red, llamado coordinador PAN. La topología *peer-to-peer* solo tiene un coordinador PAN, sin embargo, es diferente de la topología estrella debido a que cualquier enrutador puede comunicarse con cualquier otro mientras la distancia entre ellos esté en el rango de comunicación directa, y se emplean algoritmos de encaminamiento de mensajes entre segmentos de la red. La topología *peer-to-peer* permite establecer redes más complejas, tales como la topología *cluster-tree*, en la que los enrutadores pueden cumplir funciones de coordinador dentro de un grupo de nodos, como permitir la vinculación de nuevos nodos y la sincronización de los dispositivos del grupo, aunque existe un único coordinador general de toda la red, que es el que la inicia.

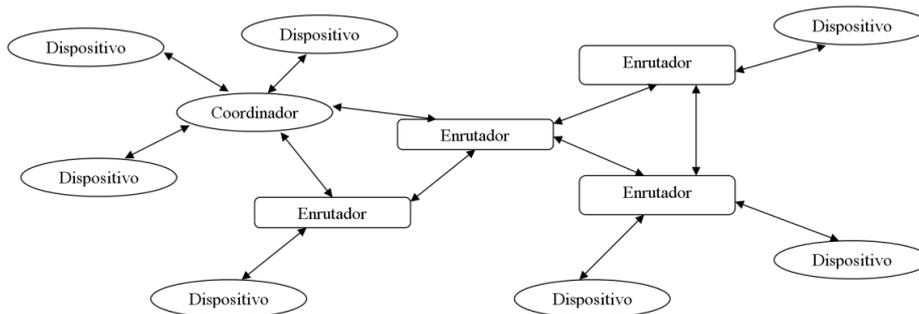


Figura 2.6. Topología *Peer-to-Peer*

El estándar IEEE 802.15.4-2003, implementa dos tipos básicos de acceso al medio, sin sincronización y con sincronización a través de *beacons*, los cuales son enviados por el coordinador. La ventaja del modo sin sincronización es que facilita la escalabilidad y la autoconfiguración de la red, aunque no ofrece garantía en cuanto al tiempo de envío de los mensajes.

En redes sin *beacon* se utiliza el algoritmo CSMA-CA para acceder al canal. Cuando algún dispositivo desea transmitir en la red primero revisa si otro dispositivo se encuentra transmitiendo sobre el canal, en cuyo caso se retira de la contienda e intenta acceder después, o indica un fallo de conexión después de varios intentos fallidos. Una trama de reconocimiento (ACK) confirma si una transmisión fue exitosa, y se envía inmediatamente después de que cada paquete de información es recibido.

Empleando el modo de acceso con sincronización de IEEE 802.15.4 es posible garantizar tiempos máximos de envío de información, asignando a los nodos un GTS dentro de una supertrama, figura 2.7, lo cual es muy importante para el desarrollo de aplicaciones de control (Martínez, 2009a). Adicionalmente existen primitivas por medio de las cuales el coordinador puede modificar la asignación de los GTS. Solamente es posible asignar un máximo de siete ranuras, con un periodo mínimo de trama de 15,36 ms, lo cual puede ser suficiente en algunos casos.

En una supertrama existen dos periodos durante los cuales los nodos pueden acceder al canal, el periodo de acceso por contención (CAP) y el periodo libre de contención (CFP, *contention-free period*), donde se asignan los GTS. Cualquier dispositivo que desee transmitir durante el CAP, espera a que empiece la siguiente ranura de tiempo (*time slot*) y si detecta que algún otro dispositivo se encuentra transmitiendo en la misma ranura, el dispositivo se retira de la contienda durante un número aleatorio de ranuras o indica un fallo en la conexión después de varios intentos.

Los GTS pueden ocupar más de un *slot*, forman el periodo libre de contención de la supertrama, el cual siempre aparece al final de la supertrama. El periodo inactivo es opcional, y en él los nodos pueden ser puestos en un estado de bajo consumo de potencia.

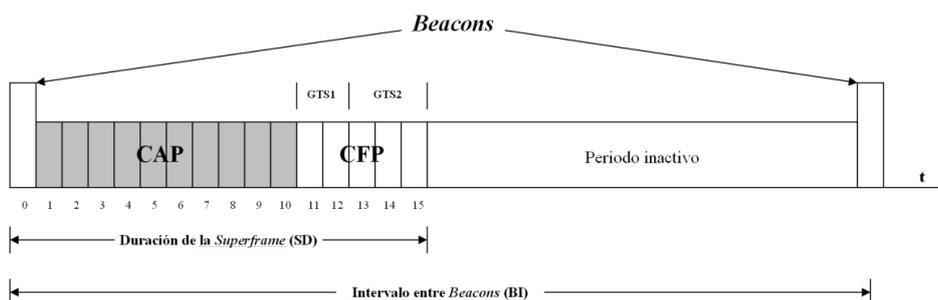


Figura 2.7. Estructura de una supertrama

La forma de calcular cada uno de estos parámetros de la supertrama es la siguiente:

- $BI = aBaseSuperframeDuration \cdot 2^{BO}$
- $SD = aBaseSuperframeDuration \cdot 2^{SO}$

- $0 \leq SO \leq BO \leq 14$
- $aBaseSuperframeDuration = 15.36ms$, para una velocidad de 250kbps en la banda de 2.4GHz.

Aunque en principio el empleo de supertramas está restringido a comunicaciones en redes estrella, lo cual limita la confiabilidad y escalabilidad de la red, en (Koubâa, 2008) se presenta un método para la asignación de los GTS en una red *cluster-tree*, denominado *Time Division Beacon Scheduling*. En este método se modela el problema de la asignación de tiempos para la supertrama como un ejecutivo cíclico, y se soporta en la aproximación por división de tiempo propuesta en IEEE 802.15.4-2006, en la que el intervalo entre *beacons* del coordinador de la red se subdivide en pequeños intervalos, dentro de los cuales se realiza la transmisión de las supertramas en cada cluster, evitando de esta manera las colisiones entre las tramas.

2.3.2 Algoritmos de encaminamiento

Los principales desafíos presentes en el desarrollo de algoritmos de encaminamiento para WSN son:

- La aleatoriedad de la topología de la red. Como consecuencia de que los nodos pueden ser dispersados de manera aleatoria sobre el área de interés, y además se pueden desplazar en tiempo de ejecución de la aplicación.
- La comunicación inalámbrica. En cuanto a las problemáticas presentes en los canales de radio respecto a las tasas de errores, velocidades de comunicación, cobertura, etc.
- La escalabilidad. Debido a la variedad del número de nodos que se puede presentar en estas aplicaciones.
- Requisitos de tráfico de las aplicaciones. El cual puede ser periódico o por ráfagas.
- Requisitos de calidad de servicio de las aplicaciones. Relacionados con parámetros como retrasos acotados, y el aumento del tiempo de vida de la aplicación, el cual es afectado principalmente con el consumo de potencia.

Los algoritmos de encaminamiento pueden ser clasificados en (Banâtre, 2008):

- Proactivos. Mantienen rutas continuamente para todos los nodos alcanzables, por lo que requieren diseminar periódicamente tablas de encaminamiento actualizadas.
- Reactivos. Establecen y mantienen una ruta solo si hay necesidad de comunicación a través de ella.
- De encaminamiento geográfico. Utilizan la localización en decisiones de encaminamiento.

- Jerárquicos. En estos algoritmos los nodos cumplen con funciones diferentes en la red.
- Centrados en el dato. Los nodos son asignados a la misma funcionalidad, cuyo objetivo es obtener información cuando se presenta un evento.

En aplicaciones donde los nodos se encuentran en posiciones geográficas fijas es posible utilizar algoritmos de encaminamiento soportados en tablas fijas. Sin embargo, la dinámica con la que se modifica la arquitectura de la red en aplicaciones donde existe movilidad de los nodos, incrementa la pérdida de mensajes y limita el cumplimiento de las restricciones de tiempo real presentes en la aplicación. Adicionalmente, la limitación de recursos como la memoria restringe el uso de grandes tablas para el almacenamiento de rutas.

Algunas de las propuestas realizadas para abordar el problema de encaminamiento en WSN son las siguientes:

- Inundación (*Flooding*): En este algoritmo se envía una copia del mensaje a todos los dispositivos de la red, por lo que no es muy eficiente en grandes redes, además de que el consumo de potencia es elevado.
- Algoritmo *Bellman-ford*. Éste algoritmo se apoya en tablas de encaminamiento las cuales se actualizan constantemente, por lo que los nodos las intercambian y comparan periódicamente. Éste algoritmo presenta un desempeño bajo en presencia de enlaces dañados, y no es adecuado para grandes redes debido al tamaño de las tablas.
- *Ad-hoc On-demand Distance Vector* (AODV): Es un algoritmo para el encaminamiento de la información en redes móviles *ad hoc* y es el algoritmo implementado en Zigbee, uno de los protocolos de red más utilizados en nodos comerciales de WSN. Es un algoritmo reactivo, es decir, solamente se establece una ruta hacia el destino si llega una petición de envío de información.

Una de las características que define a AODV es el uso de tablas de encaminamiento en cada nodo, para evitar transportar rutas en los paquetes. Cada destino de la tabla de encaminamiento lleva asociado un número de secuencia y un temporizador. El número de secuencia permite distinguir entre información nueva e información antigua, de tal manera que se evita la formación de bucles y la transmisión de rutas antiguas o caducadas por la red. La función del temporizador es evitar usar enlaces de los que no se conoce su estado desde hace mucho tiempo.

- *Dynamic Source Routing* (DSR). Al igual que AODV es reactivo, sin embargo se diferencia de este en que no utiliza tablas para encaminar los datos. En DSR el nodo emisor conoce la ruta completa hasta el destino; estas rutas son almacenadas en una cache y son enviadas en la cabecera del paquete. Cuando un nodo requiere enviar un dato y desconoce la ruta hacia el destino, inicia un proceso de descubrimiento dinámico de ruta.

En (Perkins, 2000) y (Broustis, 2006), se realiza una comparación de AODV y DSR en el contexto de redes inalámbricas MANETs. Para alta movilidad de los nodos AODV presenta un mejor porcentaje de entrega de paquetes que DSR, lo que se invierte para una baja movilidad. Para un número pequeño de nodos fuentes de información, inferior a 20, el retraso en la entrega de paquetes es igual en ambos algoritmos; sin embargo, al aumentar el número de nodos en la red y teniendo estos una alta movilidad, AODV presenta un menor retraso en la comunicación de los mensajes que DSR, lo cual nuevamente cambia para una baja movilidad en los nodos. Aunque para escenarios de WSN donde hay nodos con diferentes niveles de movilidad, los resultados presentados no permiten concluir cuál de los dos es el más adecuado.

Desde el punto de vista de la escalabilidad, DSR presenta un mejor desempeño que AODV considerando como métrica el porcentaje de paquetes entregados, pero al considerar como parámetro de comparación el retraso extremo-extremo en la entrega de paquetes, AODV se comporta mejor frente a cambios en el número de nodos de la red.

También se observó que DSR presenta una mayor carga en el nivel MAC que AODV, lo cual afecta el consumo de potencia. Aunque ninguno de los dos posee mecanismos para balance de carga, con lo cual algunos nodos de la red se pueden sobrecargar y de esta forma se afecta el tiempo de vida de la red.

Algunos otros algoritmos propuestos son:

- De encaminamiento geográfico: *Geographic Adaptive Fidelity (GAF)*, *SPAN*, *MFR - DIR and GEDIR*, *Geographic Random forwarding (GeRaF)*, *Geographic and Energy Aware Routing (GEAR)*.
- Jerárquicos: *Low Energy Adaptive Clustering Hierarchy (LEACH)*, *Small Minimum Energy Communication Networks (SMECN)*, *Virtual Grid Architecture Routing (VGA)*, *Hierarchical Power-Aware Routing (HPAR)*.
- Centrado en el dato: *Directed Diffusion (DD)*, *Sensor Protocols for Information via Negotiation (SPIN)*, *Rumor Routing (RR)*, *Minimum Cost Forwarding Algorithm (MCFA)*, *COUGAR*.

2.4 Algoritmos de planificación de tareas y mensajes

El desempeño de algunas aplicaciones puede ser afectado por la magnitud de los retrasos en el envío de la información entre los componentes del sistema, tal es el caso de los sistemas de control donde sus componentes se comunican a través de redes de comunicaciones, denominados *Networked Control Systems (NCS)*, estas aplicaciones tienen restricciones de tiempo real con plazos de respuesta extremo-extremo. Este tiempo es medido desde el instante en que se realiza la medición de las variables necesarias para calcular la acción de control, hasta cuando se actúa sobre el sistema, y depende de los tiempos de finalización de las tareas y del envío de los mensajes a través de la red.

En su formulación general, el problema de evaluar la planificabilidad de un sistema distribuido de tiempo real es NP-hard. Con el fin de superar esta dificultad se deben emplear restricciones y heurísticas. Un enfoque común es asignar las tareas estáticamente a los nodos que conforman el sistema y localmente utilizar un algoritmo de planificación como RM (*Rate Monotonic*) o EDF (*Earliest Deadline First*), (Spuri, 1996).

Las aplicaciones en sistemas distribuidos están caracterizadas por poseer relaciones de precedencia entre sus tareas. Si las tareas son estáticamente asignadas a procesadores independientes, las restricciones de tiempo extremo-extremo pueden ser analizadas por una teoría que considere la relación entre los *jitter* (Audsley, 1993).

En la literatura aparecen varios trabajos orientados al análisis de planificabilidad extremo-extremo, en los cuales se han empleado algoritmos de planificación en los nodos tales como RM y EDF, y protocolos de acceso al medio basados en TDMA, Token y Prioridades, (Tindell, 1994), (Tindell, 1995), (Spuri, 1996), (Palencia 1997). Adicionalmente, en estos trabajos se considera el uso de buffers para almacenar mensajes dentro de los nodos, y por tanto emplean métodos de planificación para ordenar la salida de los mensajes desde el buffer.

En lo que respecta a planificadores de tareas en los nodos, en el contexto de la WSN uno de los sistemas operativos más empleado ha sido el TinyOs (TinyOS), el cual está orientado para uso en sistemas de cómputo con grandes restricciones, tales como microcontroladores de 8 bits con pequeña cantidad de memoria. Se soporta en un modelo de programación basado en componentes y guiado por eventos, en el cual los manejadores de eventos tienen mayor prioridad que las tareas, las cuales se ejecutan basadas en un planificador FCFS (*First-Come First-Served*).

La principal motivación para utilizar planificadores guiados por eventos en este tipo de aplicaciones, es porque aquellas plataformas donde se han implementado están orientadas a abordar sistemas reactivos, por lo que el uso de estos esquemas de planificación posibilitan el envío del procesador a modos de bajo consumo de potencia por instantes de tiempo más prolongados, y de esta forma se incrementa el tiempo de vida de la aplicación. Sin embargo este tipo de planificadores no son adecuados para garantizar plazos de tiempo real, lo cual es importante en aplicaciones de control. En la serie de productos Zigbee de Chipcon y Texas Instruments se emplea un planificador de tareas basado en prioridades estáticas, y mecanismos de comunicación soportados en paso de mensajes.

Aunque la planificación por prioridades fijas es la técnica de planificación en línea más popular en sistemas de tiempo real, el uso del algoritmo EDF está tomando cada vez más atención en ambientes industriales, debido a sus beneficios en cuanto al manejo de los recursos del sistema. Actualmente EDF está disponible en lenguajes de tiempo real como RTSJ. También está disponible en sistemas operativos de tiempo real como SHaRK y ERIKA.

2.5 Escalado dinámico de voltaje

Además de garantizar el cumplimiento de los plazos de las tareas, la propuesta presentada en este trabajo busca disminuir el consumo de potencia, para lo cual se ha considerado el empleo de técnicas de escalado dinámico de voltaje (*Dinamic Voltage Scaling* - DVS). Estas técnicas intentan resolver el compromiso entre el cómputo requerido por la aplicación y el tiempo de vida de la misma, este último relacionado directamente con la duración de la energía almacenada en la batería.

Teniendo en cuenta que en la mayoría de los casos, el cómputo máximo necesario demandado por un conjunto de tareas es mucho mayor que el promedio que debe ser mantenido, y que para los procesadores diseñados con tecnología CMOS, que en la actualidad son la mayoría, el voltaje de funcionamiento depende de la frecuencia de conmutación, de modo que cuando se utiliza un nivel reducido de frecuencias el procesador puede funcionar con un voltaje de alimentación inferior; es posible emplear un nivel bajo de voltaje en los intervalos cuando el cómputo máximo no es necesario. De esta manera, como la energía disipada por ciclo en circuitos CMOS se escala de forma cuadrática con el voltaje de alimentación, las técnicas de DVS pueden proporcionar un gran ahorro de energía escalando la frecuencia y el voltaje (Pillai, 2001).

En la literatura aparecen diversos algoritmos para realizar DVS garantizando el cumplimiento de restricciones de tiempo real. En (Mejia-Alvarez, 2002) se presenta una metodología basada en heurísticas para realizar DVS, la cual requiere un bajo tiempo de cómputo. En (Saewong, 2003) se muestra un método para encontrar la frecuencia de funcionamiento óptima que permite alcanzar el mínimo consumo de potencia, sin embargo éste método es muy complejo y por tanto no es adecuado su uso en línea. En (Marinoni, 2007) se propone un método que combina DVS con una estrategia de planificación de tareas basada en un modelo elástico de las mismas; para la cual, en función de diversos objetivos de funcionamiento se altera el periodo de las tareas del sistema. En (Zhu, 2004) utilizan un *feedback control scheduling* para realizar el DVS del procesador y un planificador EDF para administrar el orden de ejecución de las tareas.

En (Pillai, 2001) se analizó el desempeño del algoritmo de escalado estático de voltaje, en el que se selecciona la frecuencia de operación más baja que permite al planificador de tareas alcanzar todos los plazos impuestos en el sistema. Empleando esta estrategia, la frecuencia de funcionamiento seleccionada es puesta de forma estática y no es alterada a menos que cambie el conjunto de tareas. Las ventajas que presenta este método de escalado es su fácil implementación, la carga que genera al sistema es muy baja y cumple en cada ciclo con los plazos de tiempo del conjunto de tareas. Sin embargo, al realizar los análisis siempre con el tiempo de cómputo del peor caso de ejecución de las tareas, es muy pesimista por cuanto no considera el cómputo requerido por las tareas en cada ciclo de ejecución y de esta forma no alcanza el mayor ahorro de energía del sistema. En este trabajo también se muestra que el planificador EDF permite alcanzar frecuencias de funcionamiento más bajas, cumpliendo con los plazos de tiempo del conjunto de tareas, que el alcanzado con RM.

2.6 Metodologías de diseño de redes inalámbricas de sensores y actuadores

En (Bakshi, 2004) abordan el problema del diseño de WSN desde una perspectiva de sistema distribuido y paralelo, y proponen una metodología que permite a los expertos de distintas disciplinas diseñar, analizar y sintetizar las aplicaciones de WSN sin necesidad de conocer detalles de la implementación. El núcleo de su metodología es una arquitectura virtual para una clase de WSN que oculta detalles del sistema para facilitar a los programadores la gestión del control de bajo nivel. El costo de la computación y la comunicación en términos de tiempo y energía se contemplan en funciones de costo, cuyo principal objetivo es encontrar la distribución de componentes de la aplicación que aumente el tiempo de vida de la aplicación, a veces incluso a costa de una mayor latencia de alguna ruta de ejecución. La métrica de rendimiento que se utiliza para la evaluación de un algoritmo depende de la aplicación.

En (Bonivento, 2005), (Bonivento, 2006a, b) se presenta un procedimiento de diseño de WSN para aplicaciones industriales, el cual comienza con una descripción de alto nivel del algoritmo de control y un conjunto de posibles plataformas de hardware para la síntesis del mismo, para posteriormente generar de manera automática una aplicación que satisface los requisitos del sistema y optimiza el consumo de potencia. Para gestionar la heterogeneidad y complejidad, se identifican tres capas de abstracción y se introducen las herramientas para la transición entre las diferentes capas y obtener la solución final.

Este enfoque de diseño a nivel de sistema, se caracteriza por una fase *top-down* donde los requisitos de la aplicación son detallados en las necesidades extremo-extremo de la red, una fase inferior donde se abstrae el rendimiento del hardware, y una fase intermedia donde se encuentran las dos fases anteriores, en la cual se utilizan los requisitos y el rendimiento para resolver un problema de optimización con restricciones, cuya solución determina los protocolos de acceso al medio y encaminamiento de la red.

Algunas restricciones del modelo de componentes a nivel de sistema limitan la solución propuesta, por ejemplo no se contempla la comunicación entre sensores, lo que es utilizado en aplicaciones de fusión sensorial. Además, en este procedimiento se separan el diseño de la red y el diseño del sistema embebido, lo que limita la optimización del sistema en parámetros como consumo de potencia, retrasos, jitter, entre otros.

El protocolo presentado para el acceso al medio implementa dos niveles. En el nivel superior, para la comunicación entre grupos de sensores, utiliza TDMA como estrategia para ahorrar energía y disminuir las colisiones de datos; mientras que en el nivel inferior emplea un acceso por contienda para intercambiar información al interior de cada grupo, por lo que no permite garantizar un tiempo máximo de envío de información. Además, para encontrar la mejor ruta el algoritmo de encaminamiento emplea una estrategia para hallar el camino más corto, lo cual no es siempre la mejor opción en términos de ahorro de energía y disminución de retrasos en la comunicación.

En (Prasad, 2007) se presenta un método de análisis y diseño de WSN soportado en la herramienta ANDES. Este consiste en un análisis previo a la implementación del sistema, el cual se soporta en técnicas teóricas de análisis para estimar los parámetros

más importantes del rendimiento del sistema, como el tiempo de vida, la cobertura y la fiabilidad, y realiza un análisis intuitivo del cumplimiento de restricciones de tiempo real en la red soportado en la capacidad de la red, la cual relaciona la distancia que deben recorrer los mensajes con su plazo relativo. Estos análisis se soportan en un conjunto de parámetros del sistema, como el número de nodos, el rango de detección de los nodos, y el ancho de banda disponible. Durante el proceso de diseño, estas técnicas de análisis se pueden aplicar de forma iterativa para ajustar parámetros del sistema en función del rendimiento deseado y el rendimiento estimado por el análisis. Sin embargo la herramienta no cubre todas las fases del ciclo de diseño de estas aplicaciones.

En (Shen, 2007) se presenta una metodología de diseño a nivel de sistema que permite encontrar configuraciones óptimas en una WSN para una aplicación específica, en donde se considera la optimización del consumo de potencia como criterio fundamental. Para la exploración eficiente del espacio de diseño proponen una metodología de diseño de WSN a nivel de sistema, a través de la cual se pretende alcanzar configuraciones optimizadas que minimicen el consumo de potencia. Este enfoque se soporta en el uso de componentes estándar para el procesamiento de datos, comunicación y control, y realiza una optimización de los parámetros de configuración en el contexto de la aplicación específica. El método de optimización utilizado se basa en una estrategia llamada *particle swarm optimization* (PSO), a partir del cual se obtiene una configuración eficiente de la WSN que optimiza el consumo de potencia.

(Streichert, 2007) propone una metodología para el diseño de WSN que se soporta en la generación de dos gráficos que permiten modelar requisitos en redes heterogéneas, a partir de las cuales se genera una solución a la especificación empleando algoritmos evolutivos multiobjetivo. El gráfico de la arquitectura contiene todos los recursos disponibles, mientras que el gráfico de los problemas contiene los recursos requeridos por el sistema final. Los mensajes entre las funciones están asignados a enlaces de comunicaciones. Para encontrar la solución al problema multiobjetivo se determina el conjunto de soluciones óptimas de Pareto. En esta propuesta no se consideran restricciones de tiempo real.

Existen varias propuestas dirigidas a un problema similar al abordado en (Streichert, 2007), el cual se conoce comúnmente como la exploración del espacio de diseño de sistemas embebidos. Por desgracia, ya que estos enfoques están destinados para los diseños de sistemas embebidos, no existen extensiones directas para explorar las alternativas de implementación en WSN. Algunas de estas propuestas son:

- SPADE (*System-level Performance Analysis and Design space Exploration*) es una herramienta para analizar arquitecturas para procesamiento digital de señales, la cual se incorporó en Artemis (*Architectures and Methods for Embedded Media Systems*) para explorar el espacio de diseño.
- MILAN (*Model-based Integrated simuLatioN*), es una herramienta que trabaja en diferentes niveles de abstracción. Gráficos jerárquicos de flujo de datos incluyen alternativas para especificar la aplicación, así como arquitecturas para definir y explorar en los diferentes niveles de detalle.

- EXPO es una metodología de exploración del espacio de diseño soportada en algoritmos evolutivos para aplicaciones de procesamiento de paquetes.
- CHARMED (Co-synthesis of HARdware-software Multimode EmbeddeD systems) es un *framework* para la exploración automática del espacio de diseño sistemas embebidos multimodo.
- Metrópolis, es un *framework* para explorar el espacio de diseño, que integra herramientas para la simulación, verificación y síntesis.

En la literatura aparecen diferentes herramientas de apoyo al desarrollo de aplicaciones de WSN (Campanoni, 2008), entre ellas: Ns-2, SensorSim, EMStar, OPNET, OMNET++, Avrora, TOSSIM, VisualSense/Ptolemy, SENSE, J-Sim. Estos ambientes permiten definir una red de nodos con la especificación de una capa de red a medida que considera la peculiaridad de los protocolos inalámbricos. La mayoría de ellas ofrecen algún tipo de simulación de eventos discretos, Ptolemy también permite mezclar distintos dominios de computación y aspectos como la movilidad de los nodos. Sin embargo, ninguno está tratando con una medida adecuada y formal la capacidad de la red para capturar los eventos que se supervisan. Con frecuencia su enfoque está relacionado con la optimización del costo, o la verificación a través de simulaciones de propiedades como el consumo de potencia, la robustez y el análisis de los modelos de computación (middleware).

En (Campanoni, 2008) se presenta un método que permite encontrar la ubicación y agrupación de nodos sensores con base en la minimización del costo de implementación. Inicialmente se verifica la existencia de una solución viable al problema de detección de eventos, con la precisión requerida por la aplicación. Luego, aprovechando las capacidades del método de optimización SWORDFISH (*Sensor networks Development Framework Integrating Simulation and Hardware optimization*), se refina la arquitectura de la WSN con base en las restricciones y objetivos de diseño.

En (Meshkova, 2008) se propone una metodología de diseño de WSN orientada al servicio, mas no se presentan herramientas para automatizar el procedimiento propuesto. Esta metodología se fundamenta en la interacción entre componentes para proporcionar un servicio con una determinada calidad. Las etapas de la metodología de diseño son: recopilación de los requisitos, análisis, diseño de la solución, desarrollo de la arquitectura de software, desarrollo del código, implementación y pruebas.

Los principales parámetros identificaron para la descripción de estas aplicaciones son el coste, el tiempo de vida, el retraso, la tolerancia a fallos y los servicios. Para la descripción del nivel de red se incluye información sobre la movilidad esperada de los nodos, la tasa de fallos, el ancho de banda mínimo requerido, el número de nodos en la red, el diámetro promedio de la red, la simetría de la red y la heterogeneidad. Los parámetros de los servicios incluyen una lista de las funcionalidades previstas y los servicios necesarios, para lo cual cada servicio se caracteriza por la calidad esperada del mismo. Respecto a la especificación del hardware y software se propone la especificación de la plataforma de hardware y del sensor, el sistema operativo requerido, la cantidad de memoria necesaria, y una lista de los módulos requeridos para

el funcionamiento del software.

2.7 Conclusiones

Como se ha mencionado en los apartados anteriores, factores como la movilidad de los nodos en algunas aplicaciones, la limitación de los recursos y la escalabilidad, tienen una incidencia importante en el desarrollo de estos sistemas. También se aprecia que su desempeño está fuertemente determinado por las restricciones impuestas por la aplicación, y que los trabajos realizados hasta el momento se han orientado principalmente al planteamiento de arquitecturas de nodos y protocolos de comunicaciones.

Con base en la revisión realizada sobre el estado actual de la tecnología y las aplicaciones, los principales desafíos que actualmente se presentan en esta área están relacionados principalmente con:

- El soporte de los niveles de calidad de servicio propios de estas aplicaciones.
- La seguridad, relacionada principalmente con la protección de la información.
- La proposición de estrategias de cooperación entre los niveles de la arquitectura del sistema, y el planteamiento de nuevas arquitecturas de nodos y protocolos de comunicaciones, que contribuyan a la optimización de los parámetros de mayor relevancia en el sistema.
- El planteamiento de metodologías y herramientas de diseño de WSN que permitan un análisis total de la aplicación, integren métodos de verificación del cumplimiento de plazos extremo-extremo, y realicen optimizaciones considerando no solamente el consumo de potencia, sino que permitan establecer un compromiso con otros índices de desempeño, como por ejemplo las latencias.

Capítulo 3

ARQUITECTURA DE LOS NODOS

En la revisión del estado actual de las temáticas de WSN se concluyó que estos desarrollos están determinados por el cumplimiento de fuertes restricciones impuestas por la aplicación, tales como restricciones de consumo de potencia, limitado poder de cómputo, cobertura de grandes áreas, restricciones de tiempo real, entre otras. Con base en ello, en este capítulo se propone una arquitectura para los nodos que conforman este tipo de aplicaciones.

3.1 Arquitectura propuesta para los nodos

La clasificación de las aplicaciones realizada en el capítulo 2 permite apreciar que en una misma aplicación pueden existir nodos con requisitos de cómputo y tamaño de memorias bastante disímiles al de otros, lo que llevó a plantear la posibilidad de contar con dos tipos de nodos, bajo desempeño y alto desempeño. La descripción de los tipos de nodos es la siguiente:

- Nodos de bajo desempeño. Pueden ser desarrollados con sistemas en chip que contengan los coprocesadores de comunicaciones, cifrado de datos y localización, tales como los actuales procesadores CC2430/31, CC2530/31 y ATmega128RF, estos facilitan la implementación de los protocolos IEEE 802.15.4 y Zigbee, entre otros. El CC2431 también tiene implementado un componente hardware que facilita la localización de nodos empleando algoritmos de triangulación.
- Nodos de alto desempeño. Para el desarrollo de estos nodos se emplea un procesador para ejecutar la aplicación y un coprocesador para implementar los protocolos de la red de comunicaciones. Para implementar la aplicación se sugiere utilizar procesadores MSP430, ARM7 o XScale, los cuales realizan un buen balance entre el cómputo realizado y la potencia consumida; adicionalmente los procesadores XScale facilitan la implementación de estrategias de ahorro de energía soportadas en DVS. Como coprocesador se sugieren el MRF24J40 y el ATmega128RF para facilitar la implementación de ranuras de tiempo garantizadas, o el CC2480 con el que se facilita el desarrollo de aplicaciones con el protocolo Zigbee.

Para los nodos de alto desempeño se propone una arquitectura lógica, la cual se presenta en la figura 3.1, (Martínez, 2010a). Los niveles que conforman la arquitectura de los nodos de alto desempeño han sido seleccionados considerando el análisis de los requisitos de las aplicaciones y el estado actual de las temáticas presentadas en el capítulo 2. La función de los diferentes niveles de la arquitectura del nodo es:

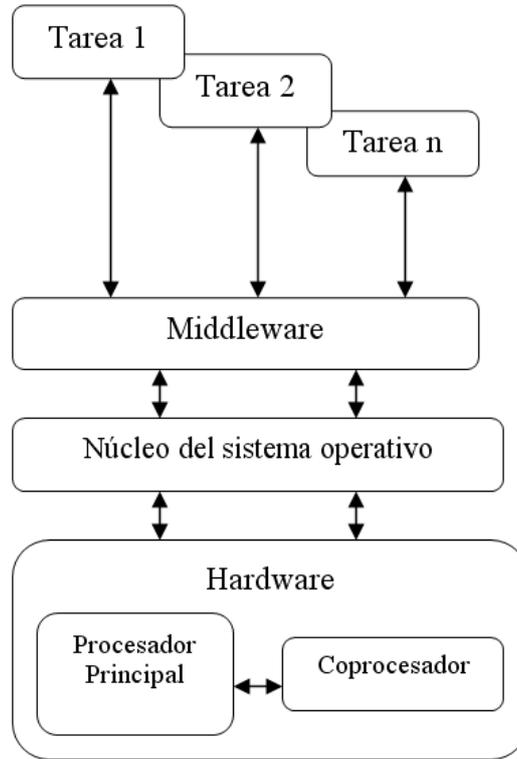


Figura 3.1. Arquitectura lógica de los nodos de alto desempeño

- **Tareas:** Realizan las actividades relacionadas con la aplicación.
- **Middleware:** Recibe las solicitudes de las tareas a partir de lo cual realiza la configuración del modo de funcionamiento para un determinado escenario, entendido este como un conjunto de tareas con requisitos de cómputo y restricciones de tiempo conocidos. Lo cual realiza enviando órdenes al sistema operativo para que, con base en el algoritmo de escalado estático de voltaje presentado en (Pillai, 2001), configure el procesador (frecuencia y voltaje de funcionamiento) y active las tareas que ha de ejecutar el nodo durante dicho modo de funcionamiento.
- **Núcleo del sistema operativo:** Es un núcleo de soporte mínimo, que realiza la ejecución de las tareas con base en una política de planificación EDF, el cual es adecuado para abordar sistemas de tiempo real.
- **Hardware.** Está compuesto de dos procesadores, el procesador principal y el coprocesador. El procesador principal ejecuta las tareas y permite el uso de estrategias de DVS para disminuir el consumo de potencia, para lo cual es posible utilizar procesadores con arquitecturas como la del XScale (Varma, 2005). El coprocesador implementa los niveles físico, de enlace de datos y de

encaminamiento de mensajes, de la red de comunicaciones. De tal forma que estas actividades no afecten el desempeño de las funciones de procesamiento, garantizando el nivel de calidad de la aplicación y la sincronización de los nodos de la red. En relación al acceso al medio se emplea un algoritmo TDMA, el cual es importante para cumplir con los plazos de tiempo en el sistema e implementar estrategias de ahorro de energía en la comunicación.

La arquitectura propuesta propicia el desarrollo de sistemas sensibles a los retrasos en la comunicación, la reconfiguración del sistema en función del cambio dinámico de los requisitos de la aplicación, y al desarrollo de soluciones de compromiso entre el consumo de potencia y la ejecución en tiempo real de los algoritmos de control en NCS, los cuales han sido identificados como retos actuales en el desarrollo de estas aplicaciones (Tariq, 2011).

3.2 Análisis de los requisitos de cómputo en un caso para la detección de arritmias

En este apartado se analiza a modo de ejemplo representativo el cómputo requerido para abordar un caso de estudio para la detección de arritmias, con el objetivo de establecer los procesadores para la arquitectura de los nodos en este tipo de aplicaciones.

Para la elección del procesador se analizó el cómputo requerido por un algoritmo de detección de arritmias a partir de la medida del electrocardiograma (ECG), en pacientes que se encuentran en movimiento durante una sesión de rehabilitación en un espacio cerrado, (Martínez, 2008). Para tal efecto se toman muestras de la señal ECG cada 3 ms.

Para la detección de arritmias se utilizó el Algoritmo de Pan y Tompkins, (Pan, 1985). La secuencia de etapas que se aplican a la señal ECG en este algoritmo es la siguiente. Inicialmente se aplica un filtro pasa banda, el cual reduce el ruido producido por componentes armónicos de la señal que se encuentran fuera de la banda de frecuencias del segmento QRS del ECG. Posteriormente se utiliza una función derivada (denominado *derivador*) que enfatiza las pendientes del segmento R del ECG. Luego se eleva al cuadrado la señal resultante del derivador para acentuar aún más las características de alta frecuencia del complejo QRS. Por último, se hace una estimación de la energía con una ventana móvil del tamaño del complejo QRS más largo. En la Figura 3.2 se muestra un diagrama de bloques de este algoritmo.



Figura 3.2. Diagrama de bloques del algoritmo de Pan y Tompkins

Los resultados del algoritmo de Pan y Tompkins son utilizados por un algoritmo de detección de picos, el cual identifica los instantes de tiempo en que se presentaron los segmentos de la onda ECG, y posteriormente se realiza un análisis del tiempo de separación entre dos segmentos R, la duración del complejo QRS y la altura de la onda R, con lo cual se detecta la presencia de arritmias (Tompkins, 1981).

Para seleccionar el procesador que mejor cumple con los requisitos de estas aplicaciones, se comparó el desempeño del algoritmo de detección de arritmias en cuatro tipos de procesadores utilizados actualmente para implementar nodos en redes de sensores: ARM7TDMI, MSP430, PIC18, MC9S08GB60.

Para el análisis se utilizó la misma velocidad de operación para cada procesador, 8MIPS. El tiempo de cómputo total de la implementación del algoritmo de Pan y Tompkins se presenta en la tabla 3.1, el cual se estimó considerando la suma de los valores de las funciones individuales en cada arquitectura. Los resultados muestran como la arquitectura ARM es la que requiere un menor porcentaje de utilización (U), mientras la arquitectura PIC es la que mayor U necesita.

Tabla 3.1. Tiempo de cómputo para desarrollar el algoritmo de Pan y Tompkins

Procesador	Derivación	Elevación al Cuadrado	Ventana Integradora	Tiempo Total de Computo	Periodo	U
LPC2124 – ARM	70.2 μ s	142 μ s	280.5 μ s	492.7 μ s	3000 μ s	16.4%
MSP430F1611	191.9 μ s	162.5 μ s	697.8 μ s	1052.2 μ s	3000 μ s	35%
PIC18F458	406.2 μ s	209 μ s	1083.7 μ s	1698.9 μ s	3000 μ s	56.6%
MC9S08GB60	497.2 μ s	332 μ s	707.35 μ s	1536.55 μ s	3000 μ s	51.3%

El otro aspecto considerado fue la potencia activa consumida (P_A) por cada arquitectura durante la ejecución del algoritmo de Pan y Tompkins. Para ello se relacionó la P_A consumida por cada arquitectura con el respectivo porcentaje de utilización de la CPU durante la ejecución del algoritmo.

En la tabla 3.2 se presenta el indicador $P_A * U$. Se puede observar como la arquitectura ARM7 posee un valor muy cercano al de la arquitectura MC9S08GB60. La arquitectura MSP430 presenta el menor valor, y por tanto mejor, de este indicador.

Tabla 3.2. Indicador $P_A * U$ en el desarrollo el algoritmo de Pan y Tompkins

	Porcentaje de Utilización (U)	Potencia Activa (P_A) [mVA]	$P_A * U$
LPC2124 – ARM	16.4%	180	29.52
MSP430F1611	35%	19.2	6.72
PIC18F458	56.6%	220	124.52
MC9S08GB60	51.3%	51.6	26.47

Por lo anterior se concluye que para esta aplicación el procesador que permite encontrar un mejor balance entre el cómputo requerido y potencia consumida es el MSP430, el cual ha de cumplir con la función de procesador principal en la arquitectura de nodo de alto desempeño.

3.3 Selección del protocolo de comunicaciones

El tipo de aplicaciones de WSN consideradas en este trabajo demanda un ancho de banda en la comunicación reducido, por lo que en esta propuesta este parámetro se acotó a 250kbps, lo cual se puede obtener con el protocolo IEEE 802.15.4.

Con base en el análisis de las aplicaciones que se realizó en el capítulo 2, es posible establecer dos tipos de requisitos de las aplicaciones en función de los retrasos en el envío de información a través de la red de comunicaciones, aplicaciones que demandan un tiempo de envío de información tan corto como sea posible, y aplicaciones que demandan un tiempo de envío de información acotado.

Para abordar aplicaciones con restricciones de tiempo real, donde se demanda un tiempo de envío de información acotado, en este trabajo se ha optado por la utilización de GTS para el acceso a la red y el empleo de rutas fijas para el encaminamiento de mensajes.

Actualmente, para el desarrollo de WSN en las que no hay requisitos de tiempo real estricto el protocolo que más se utiliza es Zigbee, este se soporta en el algoritmo AODV para realizar el encaminamiento de mensajes, el cual emplea una función de costo para elegir la mejor ruta o camino.

Para la selección de la mejor ruta entre un nodo fuente y otro destino de la información, se define un camino P de longitud L como un conjunto ordenado de dispositivos $[D_1, D_2, \dots, D_L]$, y un enlace como un sub-camino de dispositivos $[D_i, D_{i+1}]$, por lo que el costo de enviar información a través de un determinado camino es:

$$C\{P\} = \sum_{i=1}^{L-1} C\{[D_i, D_{i+1}]\}, \text{ donde } C\{[D_i, D_{i+1}]\} \text{ es el costo de un enlace.}$$

Uno de los parámetros más utilizados para calcular el costo de un enlace es el *Indicador de Calidad del Enlace* (LQI), el cual se relaciona con la probabilidad de tener éxito en el envío de información a través de una determinada ruta. Sin embargo es posible emplear más parámetros, en el IEEE 802.15.4 por ejemplo se plantea el uso del LQI y el ED (*Detección de Energía*).

Como se ha mencionado, en términos generales los parámetros de mayor importancia para el desarrollo de WSN son el consumo de potencia y la latencia en el envío de la información. Los principales factores que afectan la latencia son el número de saltos que existe en el camino entre fuente y destino de la información, y la saturación de los buffer de los enrutadores en cada salto. Los algoritmos que únicamente consideran el número de saltos generan un alto consumo de potencia en los nodos más cercanos a los nodos destinos de la información.

En relación al consumo de potencia en una WSN, lo que se pretende es encontrar soluciones que garanticen un consumo de potencia distribuido entre todos los nodos de la red. Esto se logra distribuyendo el cómputo en los nodos, el cual se relaciona con el

procesamiento de las señales requerido por la aplicación y la utilización del nodo para encaminar mensajes.

Con base en lo anterior, en este trabajo se ha propuesto la siguiente ecuación para calcular el costo de un camino en una WSN:

$$C\{P\} = \sum_{i=1}^{L-1} C\{[D_i, D_{i+1}]\}_{EN,LQI,SB} + K_{nh}nh \quad (1)$$

Donde:

- nh es el número de saltos entre la fuente y el destino de la información.
- $C\{[D_i, D_{i+1}]\}_{EN,LQI,SB} = (K_{EN} * EN) + (K_{LQI} * LQI) + [K_{SB} * (SB + \Delta SB)]$

EN es la energía en el nodo enrutador del que se desea calcular el costo, SB es la longitud del buffer del nodo enrutador, y ΔSB es la variación que ha tenido la longitud del buffer de dicho enrutador desde la última vez que se analizó un establecimiento de una ruta a través de él.

- K_{EN} , K_{LQI} , K_{SB} y K_{nh} representan el peso de cada parámetro en la función de costo, los cuales pueden variar de acuerdo a los requisitos específicos de la aplicación.

Utilizando la ecuación (1) se pueden encontrar rutas en las que se logre un balance entre la latencia media en el envío de los mensajes y el tiempo de vida de la red. Además, los cálculos que se deben realizar no significaran un incremento elevado en el cómputo y la capacidad de memoria de los nodos, como tampoco en la información que circula en las tramas que actualmente utilizan los protocolos para encontrar las rutas.

Para analizar el desempeño de una WSN que utiliza la ecuación (1) se simuló un caso de estudio en Truetime, (Cervin, 2003), figura 3.3. El planteamiento es el siguiente:

- Se dispone de una WSN compuesta de 7 nodos, sobre los cuales se implementan dos aplicaciones:
 - La aplicación 1 envía un dato cada 100 ms desde el nodo 1 hacia el nodo 5.
 - La aplicación 2 envía un dato cada 300 ms desde el nodo 2 hacia el nodo 7.
- El tiempo requerido para el envío de un mensaje y la recepción del mismo, en un único salto, es de 100 μ s.
- El valor inicial de la fuente de energía es 1 J, y se disminuye en 0.01 J por cada transmisión o recepción que se realiza.

- Los parámetros temporales del protocolo AODV son:
 - El intervalo durante el cual se considera activa una ruta es de 500 ms.
 - Se envían mensajes *Hello* cada 200 ms, con el propósito de que un nodo confirme su conectividad en la red a sus nodos vecinos.

Durante los análisis se comparó el desempeño de la aplicación considerando dos funciones de costo:

- $C\{P\}_1 = \sum_{i=1}^{L-1} nh$, en la que solo se considera el número de saltos.
- $C\{P\}_2 = \sum_{i=1}^{L-1} (10 * EN) - 10[K_{SB} * (SB + \Delta SB)] + 0.1nh$, en la que se utiliza la ecuación (1).

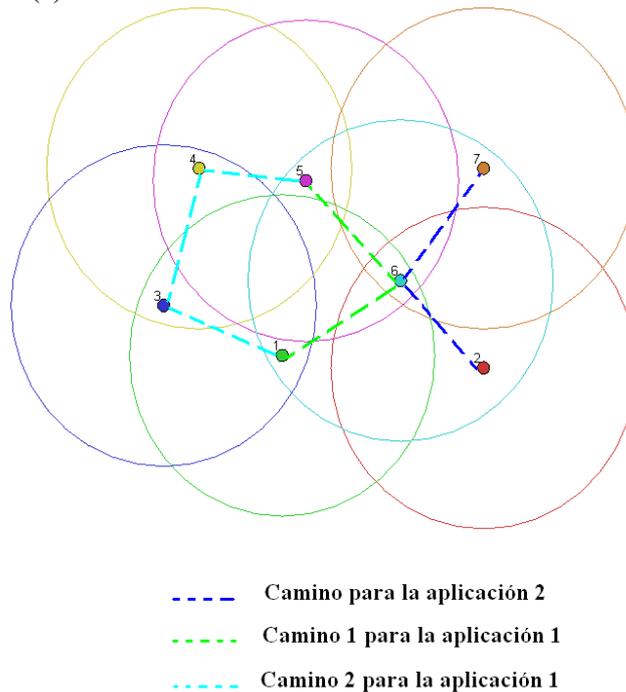


Figura 3.3. Caso de estudio propuesto para analizar el desempeño de una WSN que utiliza la ecuación (1) para el establecimiento de rutas

Los resultados de los primeros 3.5 segundos de simulación se presentan en la tabla 3.3. Es importante aclarar que además de los datos, los nodos también intercambian información de control del protocolo.

Tabla. 3.3. Resultados del análisis del desempeño de una WSN que utiliza la ecuación (1) para el establecimiento de rutas

Función	Camino, aplicac. 1	Camino, aplicación 2	Latencia media apl. 1	Energía final en los nodos [J]
$C\{P\}_1$	1 – 6 – 5	2 – 6 – 7	732 μ s	N1=0.58; N2=0.8; N3=0.92; N4=0.93; N5=0.58; N6=0.34 ; N7=0.82.
$C\{P\}_2$	- Inicial: 1–6–5 - Final: 1 – 3 – 4 – 5, a partir de 2.1 s	2 – 6 – 7	723 μ s	N1=0.46; N2=0.75; N3=0.71; N4=0.73; N5=0.53; N6=0.47 ; N7=0.8.

Se puede apreciar que con la función $C\{P\}_2$ se logra una mejor distribución del consumo de potencia entre los nodos de los posibles caminos para el envío de la información, logrando aumentar el tiempo de vida de la aplicación. Esto se observa en el valor final de la energía en el nodo 6, el cual establece la ruta mas corta para el envío de los mensajes de las dos aplicaciones.

Al mismo tiempo se logró una disminución en la latencia media en el envío de los datos de la aplicación 1, la cual fue afectada por las colisiones en los mensajes, el número de saltos en las rutas y la cantidad de datos en los buffer de los nodos.

Capítulo 4

VALIDACIÓN DEL DISEÑO

La naturaleza concurrente y no determinista de estos sistemas hace que su análisis y diseño sean complejos. Estos sistemas pueden evolucionar por diferentes estados y dada su magnitud, es muy probable que se presenten condiciones de operación no consideradas durante las fases de diseño, que conlleven a un desempeño no deseado del sistema. Esto ha generado que en algunos casos no se obtenga una buena correspondencia entre los resultados experimentales y los objetivos de los algoritmos de procesamiento de señales propuestos, lo cual es consecuencia del uso de modelos imprecisos para analizar y diseñar estos sistemas, métodos de validación poco elaborados y plataformas que no soportan los modelos empleados.

Por lo anterior, en este trabajo se han propuesto estrategias para analizar los requisitos temporales, y las propiedades de comportamiento y estructurales, de aplicaciones de NCS implementados sobre WSA, considerando la arquitectura de nodos propuesta. Para el análisis temporal del sistema se han desarrollado dos test de planificabilidad para WSA. Mientras que para validar las propiedades de comportamiento y estructurales, y poder realizar análisis dinámicos del sistema completo, se han introducido modelos de redes de Petri coloreadas (CPN). A continuación se presenta cada uno de los métodos de validación.

4.1 Notación y modelo del sistema

Como tipo de aplicaciones con retrasos acotados en este trabajo se han considerado NCS. En un esquema general, las tareas que realizan las funciones dentro del NCS, *Medición*, *Control* y *Actuación*, se ejecutan en dicha secuencia, en nodos independientes y en exclusión mutua (no se puede iniciar la ejecución de la siguiente función si la anterior no ha finalizado), se definen:

- T_{SF} , es el periodo con el que se repite el itinerario de ranuras en la red TDMA.
- D_{CGR} , es el plazo extremo-extremo, entre el inicio de la medición y la finalización de la actuación, impuesto por los objetivos de desempeño del sistema.
- T_S , es el periodo de muestreo empleado en la medición, el cual se define de acuerdo con la respuesta dinámica del sistema a controlar, y ha de cumplir con $D_{CGR} \leq T_S$.

- $\tau = \{Task_1, Task_2, \dots, Task_n\}$, es un sistema planificable bajo EDF, con $Task_i = (WCET_i, D_i, T_i)$; donde $WCET_i$, D_i y T_i , son los respectivos valores de tiempo de cómputo del peor caso, plazo y periodo de la tarea $Task_i$.
- $WCRT_i$, es el tiempo de respuesta del peor caso de la tarea $Task_i$.

En (Ripoll, 1996) se define:

- $H_r(t) = \sum_{i=1}^n WCET_i \left\lfloor \frac{t + T_i - D_i}{T_i} \right\rfloor$, es la cantidad de tiempo de cómputo que tiene que haber sido servida por el procesador hasta el tiempo t para cumplir con los plazos de tiempo del sistema.
- Intervalo crítico inicial (ICI), es el intervalo de tiempo entre cero y el primer instante en el que no hay activaciones pendientes $[0, R)$.

El test de planificabilidad para un conjunto de tareas ejecutadas en un mismo nodo, empleando una política de planificación EDF consiste en verificar que:

$$H_r(t) < t \quad \forall \quad t \leq R \quad (2)$$

4.2 Análisis de planificabilidad extremo-extremo para WSAAN con distancia entre nodos acotada a un único salto

Se supone que todos los nodos se encuentran conectados a la misma red y que cada mensaje se envía dentro de la ranura asignada a cada nodo en la red TDMA. Por lo que el retraso máximo introducido por la red es igual al periodo con el que se repite el itinerario de ranuras en la red TDMA.

En esta aproximación no se consideró el almacenamiento de mensajes en buffers, por cuanto la cantidad de datos enviados en las aplicaciones consideradas, es pequeño comparado con la cantidad de datos que es posible enviar en cada mensaje de los actuales protocolos estándar (máximo *payload* en la PDU del nivel físico de 127 bytes para Zigbee y WirelessHART).

Luego el análisis de planificabilidad extremo-extremo para el sistema de control se garantiza cuando D_M , D_C y D_A cumplen con las expresiones presentadas en la tabla 4.1, donde D_M , D_C y D_A son los plazos de las tareas de *Medición*, *Control* y *Actuación* respectivamente (Martínez, 2009b).

Para la implementación del sistema es posible asignar a D_M , D_C y D_A un valor entre $[D_j^{\min}, D_j^{\max}]$, en función de su impacto en parámetros como el consumo de potencia en cada nodo, los retrasos y jitter en el sistema de control, entre otros. Siendo D_j^{\min} y

$D_j^{m\acute{a}x}$ los valores mınimo y mımico de D_j que permiten alcanzar la planificabilidad local y extremo-extremo respectivamente. Donde D^{min} se obtiene a partir del algoritmo *Deadlinemin* presentado en (Balbastre, 2008).

Tabla 4.1. Parımetros para el analisis de planificabilidad extremo-extremo, en redes con distancia entre nodos acotada a un unico salto

Tarea <i>Medicion</i>	Tarea <i>Control</i>	Tarea <i>Actuacion</i>
Periodica, $T_M = T_S$	Esporadica, $T_{C_{Mn}} = T_S$	Esporadica, $T_{A_{Mn}} = T_S$
$WCET_M \leq WCRT_M \leq D_M$	$WCET_C \leq WCRT_C \leq D_C$	$WCET_A \leq WCRT_A \leq D_A$
$D_M^{m\acute{a}x} = D_{CGR}$ $-(WCRT'_C + WCRT'_A + T_{SF})$	$D_C^{m\acute{a}x} = D_{CGR}$ $-(WCRT'_M + WCRT'_A + T_{SF})$	$D_A^{m\acute{a}x} = D_{CGR}$ $-(WCRT'_M + WCRT'_C)$
$WCRT'_M = WCRT_M + T_{SF}$	$WCRT'_C = WCRT_C + T_{SF}$	

El algoritmo *Deadlinemin* parte de la suposicion de que el conjunto de tareas, τ , es planificable, y se soporta en el valor que toma la funcion $H_\tau(t)$ durante todo el ICI. Este algoritmo realiza iteraciones modificando el plazo de la tarea a la que se le desea encontrar su plazo mınimo, lo que no modifica el ICI, y mientras se cumpla con la ecuacion (2) el τ continua siendo planificable. El pseudocodigo del algoritmo *Deadlinemin* es el siguiente:

Entradas: τ , $Task_i$ a la que se le desea encontrar el plazo mınimo.

Calcular: R , $k = \lceil R/T_i \rceil$

$D_i^{min} = WCET_i$

$deadline = WCET_i$;

for $s = 0:(k-1)$

$t = \min[(sT_i) + D_i, R]$

while $[t > (sT_i + WCET_i)]$

if $[(t - H(t)) < WCET_i]$

$deadline = H_\tau(t) + WCET_i - (sT_i)$

end

$t = t - 1$;

end

$D_i^{min} = \max(D_i^{min}, deadline)$

end

retornar D_i^{min}

La utilización del algoritmo *Deadlinemin* es importante porque en el desarrollo de las estrategias de control en NCS se establecen únicamente los D_{CGR} , sin embargo para realizar los análisis de planificabilidad locales es necesario contar con plazos individuales de las tareas de *Medición*, *Control* y *Actuación*. El algoritmo *Deadlinemin* permite encontrar los límites inferiores a los plazos de estas tareas en cada nodo, mientras que los límites superiores se establecen en función del cumplimiento del D_{CGR} .

Dependiendo de las características físicas del sistema, las tareas de *Medición* y *Actuación* pueden estar en el mismo nodo; de igual forma, la tarea *Control* se podrá ejecutar en cualquier nodo con el propósito de equilibrar el porcentaje de utilización entre los mismos, lo cual incide sobre el tiempo de vida en aplicaciones donde los nodos dependen de baterías. Por lo anterior, se pueden presentar tres variantes a la arquitectura general del NCS, para las cuales los valores presentados en la tabla 4.1 se deben modificar de la siguiente manera:

- *Medición* y *Control* comparten el mismo nodo. Para este caso $WCRT'_M = WCRT_M$
- *Control* y *Actuación* comparten el mismo nodo. Para este caso $WCRT'_C = WCRT_C$
- *Medición* y *Actuación* comparten el mismo nodo. Para este caso no se presenta ningún cambio en la formulación inicial, lo cual es consecuencia de que la red es utilizada para el envío de información entre el sensor y el controlador, y entre el controlador y el actuador.

4.2.1 Caso de estudio: NCS con distancia entre nodos acotada a un único salto

Para observar los resultados que se logran empleando este test de planificabilidad, se analizó el desempeño de un NCS sobre una WSN.

Existen diversos autores que han analizado el desempeño y la estabilidad en NCS considerando protocolos de red con retrasos constantes y variables, de igual forma se han realizado propuestas para alterar los algoritmos de control con el fin de contrarrestar dichos efectos, en (Salt, 2008) se presenta una revisión de los trabajos realizados en este área. Si los retrasos de red son constantes, se pueden emplear técnicas de análisis estabilidad para sistemas de tiempo discreto con retardo (Astrom, 1997), y técnicas de diseño de controladores soportadas en el predictor de Smith o controladores clásicos. Si los retrasos de red son aleatorios, estos pueden ser modelados como una distribución probabilística constante o como una cadena de Markov, y se propone el uso algoritmos de control LQG, LQR, control óptimo, entre otros.

En (Walsh, 2002) se considera un modelo de NCS donde la planta y el controlador son continuos, se desprecian los retrasos de propagación y de procesamiento, y el medio de comunicación es libre de errores; con lo cual se propone un nuevo algoritmo dinámico para el acceso a la red y funciones para garantizar la estabilidad asintótica global de un NCS. En (Zhang, 2001) consideran la planta continua y el controlador discreto, modelan

un sistema de control en red con pérdida de datos, y determinan la estabilidad del NCS con altas frecuencias de pérdida de datos. En (Branicky, 2002) se considera una planta continua y un controlador discreto, analizan la relación entre el periodo de muestreo y el retraso de la red para establecer la estabilidad del NCS, y aplican un algoritmo de planificación RM para el acceso a la red. Otros trabajos sobre estabilidad de NCS se presentan en (Gregory, 1999), (Yang, 2006), (Hespanha, 2007), (Tabbara, 2007), (Hu, 2008), (Huang, 2008), (Xiong, 2009).

Sin embargo no existe una técnica de análisis de estabilidad genérica que pueda ser aplicada a cualquier NCS. Cada técnica está sujeta a la configuración de la red, el protocolo de comunicaciones empleado, las suposiciones realizadas para el sistema y el algoritmo de control utilizado.

Por otro lado, las metodologías de diseño de controladores empleadas se han enfocado de diferentes maneras, diseñando el algoritmo de control con métodos tradicionales y trasladando el problema de los retrasos al diseño del sistema de cómputo, considerando las características del sistema de cómputo y diseñando un algoritmo de control que compense su efecto, y empleando un enfoque en donde el sistema de cómputo y el algoritmo de control son co-diseñados.

En la actualidad existe un gran interés por el uso de algoritmos de control basados en eventos sobre NCS, los cuales contribuyen a disminuir la utilización de los procesadores y de la red. A pesar de su buen desempeño en sistemas con capacidades reducidas de computación y comunicación, la teoría de análisis y diseño de estos sistemas requiere un mayor desarrollo. En (Dormido, 2008) se presenta una revisión de los trabajos realizados respecto a sistemas de control basados en eventos.

En la mayoría de los trabajos realizados sobre NCS, se analiza el desempeño del sistema considerando únicamente la estabilidad del lazo cerrado, y aunque éste es un requisito básico con el que debe cumplir el sistema, en muchos casos prácticos los criterios de desempeño deseados en los sistemas de control se especifican en términos de la respuesta transitoria del sistema frente a una entrada específica, una de las entradas más empleadas es la escalón. Por tal razón en este trabajo se presenta un modelo lineal del sistema en lazo cerrado que permita analizar el efecto que genera el retraso sobre la respuesta transitoria del sistema.

La estructura del NCS considerado en este trabajo se presenta en figura 4.1. ZOH representa un retenedor de orden cero a la entrada de la planta. Los retrasos son generados por el envío de la información a través de la red entre Medidor – Controlador y Controlador – Actuador.

El NCS considerado presenta las siguientes suposiciones:

- Sistemas de control SISO (Single Input, Single Output).
- El sistema de control se encuentra sincronizado. En este caso la tarea que realiza la medición se activa periódicamente, y las tareas que realizan el cálculo de la acción de control y actúan sobre el sistema se activan por eventos.

- El retraso presente entre los instantes en que se inicia la medición y se finaliza la actuación es menor o igual a T_s .
- Las tramas son transmitidas y recibidas libres de error.

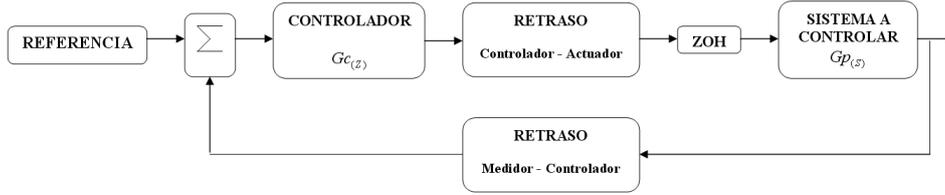


Figura 4.1. Representación general del NCS considerado

Para el análisis de cada lazo de control, considerando que el sistema de control se encuentra sincronizado, es posible llevar el retraso de realimentación al lazo directo y considerar un único retraso, figura 4.2.

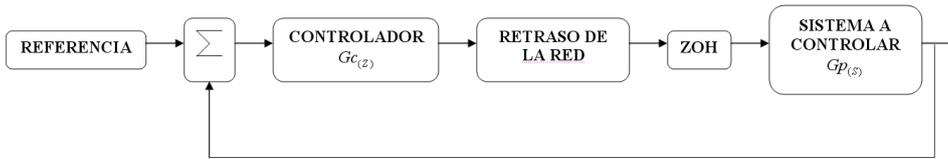


Figura 4.2. Representación del NCS como sistema sincronizado

El retraso se puede modelar como $e^{-\tau_r s}$, con $\tau_r = (1-m)T_s$ y $0 \leq m \leq 1$. Donde τ_r representa la latencia equivalente a la suma de los retrasos entre medidor – controlador y controlador – actuador.

Luego, empleando la transformada Z modificada es posible encontrar la función de transferencia en lazo cerrado del sistema entre dos periodos de muestreo consecutivos, de lo cual se obtiene:

$$H_{(z,m)} = \frac{G_{c(z)} G_{p'(z,m)}}{1 + G_{c(z)} G_{p'(z,m)}}$$

Donde:

- $G_{p'(z,m)} = (1 - z^{-1}) Z \left[\frac{G_{p(s)}}{S} \right]$
- $G_{p'(z,m)} = z^{-1} (1 - z^{-1}) \left[\sum \text{residuo de } \frac{G_{p(s)}}{S} \cdot \frac{e^{mT_s s} z}{z - e^{T_m s}} \text{ en polo de } G_{p(s)} \right]$

Para un sistema de primer orden se tiene $Gp_{(s)} = \frac{K}{\tau_p S + 1}$, luego

$$Gp'_{(z,m)} = \frac{K}{Z} \left[1 - \frac{e^{-mT_s/\tau_p} (z-1)}{z - e^{-T_s/\tau_p}} \right] \quad (3)$$

Para un sistema de segundo orden se tiene $Gp_{(s)} = \frac{w_n^2}{S^2 + 2\xi w_n S + w_n^2}$, suponiendo dos polos reales diferentes $Gp_{(s)} = \frac{w_n^2}{(S + p_1)(S + p_2)}$, luego

$$\bullet \quad Gp'_{(z,m)} = \frac{w_n^2}{Z} \left[\frac{1}{p_1 p_2} - \frac{e^{-mT_s p_1} (z-1)}{p_1 (p_2 - p_1) (z - e^{-T_s p_1})} - \frac{e^{-mT_s p_2} (z-1)}{p_2 (p_1 - p_2) (z - e^{-T_s p_2})} \right]$$

Para el caso de estudio se consideró un sistema lineal de primer orden con función de transferencia:

$$Gp_{(s)} = \frac{0.3}{0.86s + 1}$$

Utilizando (3) se obtiene,

$$Gp'_{(z,m)} = \frac{0.3}{Z} \left[1 - \frac{e^{-mT_s/0.86} (z-1)}{z - e^{-T_s/0.86}} \right]$$

Con el fin de mejorar la respuesta del sistema se utilizó un algoritmo PI de la forma,

$$U_{(z)} = \frac{k_p z + (k_i T_m - k_p)}{z - 1} \varepsilon_{(z)}$$

Los coeficientes del regulador se calcularon de tal forma que el sistema disminuyera su tiempo de establecimiento. Con lo cual se obtuvo $k_p = 30.2778$ y $k_i = 25.463$.

Para el caso de estudio considerado se suponen tres lazos de control como el anterior, y para cada uno se considera como comportamiento aceptable la respuesta transitoria para un periodo de muestreo de 110 ms y un retraso de 80 ms, figura 4.3, luego se establece $D_{CGR} = 80ms$ para los tres lazos de control.

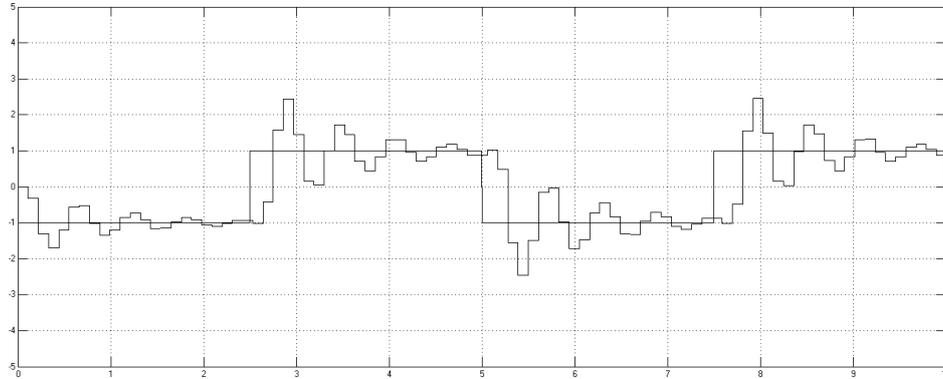


Figura 4.3. Respuesta del sistema discreto, $T_s = 110 \text{ ms}$ y $\tau_r = 80 \text{ ms}$. Referencia tren de pulsos unitario.

El planteamiento es el siguiente:

- Cada lazo de control requiere de tres tareas, *Medición*, *Control* y *Actuación*, las cuales se preasignarán cada una en un nodo diferente. Los $WCET$ de las tareas son $WCET_M = 1.5 \text{ ms}$, $WCET_C = 5 \text{ ms}$, $WCET_A = 0.3 \text{ ms}$.
- Con el propósito de aumentar la carga en los procesadores y en la red se han generado:
 - Dos tareas periódicas en cada nodo, denominadas L_1 y L_2, cuyos parámetros son: $WCET_i = 3 \text{ ms}$, $D_i = T_i = 10 \text{ ms}$.
 - Tres nodos generadores de tráfico, denominados GT_1, GT_2 y GT_3, los cuales envían una trama de 144 bits cada 35 ms .
- Los parámetros de la red de comunicaciones son:
 - Tramas de igual tamaño, 18 bytes = 144 bits.
 - Velocidad de comunicación = 250 kbps.
 - Duración de cada *slot* en la red TDMA de 1 ms .
 - El periodo con el que se repite el itinerario de los *slots* es de 35 ms .

El proceso para verificar el cumplimiento de los plazos extremo-extremo consiste en dos etapas:

- Etapa 1. Verificar la planificabilidad local en cada nodo empleando el test de planificabilidad para EDF, ecuación (2). Por cuanto solo se conoce el plazo extremo-extremo y se desconoce el plazo individual para las tareas *Medición*,

Control y *Actuación*, se establece como plazo de partida para la ejecución del algoritmo de análisis de planificabilidad local $D_M = D_C = D_A = 110ms$.

- Etapa 2. Calcular los D^{Min} , empleando el algoritmo *Deadlinemin*, y los $D^{Máx}$ que permitan el cumplimiento de la planificabilidad extremo-extremo, tabla 4.1.

Los datos presentados en la tabla 4.2 permiten observar el cumplimiento de la planificabilidad local y extremo-extremo en los tres lazos de control.

Tabla 4.2. Resultados del análisis de planificabilidad

Nodo	Planificabilidad local	D_{Min}	$D_{Máx}$
<i>Medición_i</i>	Si	1.5 ms	38.5 ms
<i>Control_i</i>	Si	11 ms	29 ms
<i>Actuación_i</i>	Si	0.3 ms	39.7 ms

Para observar el comportamiento dinámico del sistema se simuló el caso de estudio empleando la herramienta Truetime. En las figuras 4.4, 4.5 y 4.6, se presentan la planificación de mensajes en la red, la planificación de tareas en el nodo que ejecuta la tarea *Actuación* del lazo de control 1 y la respuesta de los tres lazos de control respectivamente.

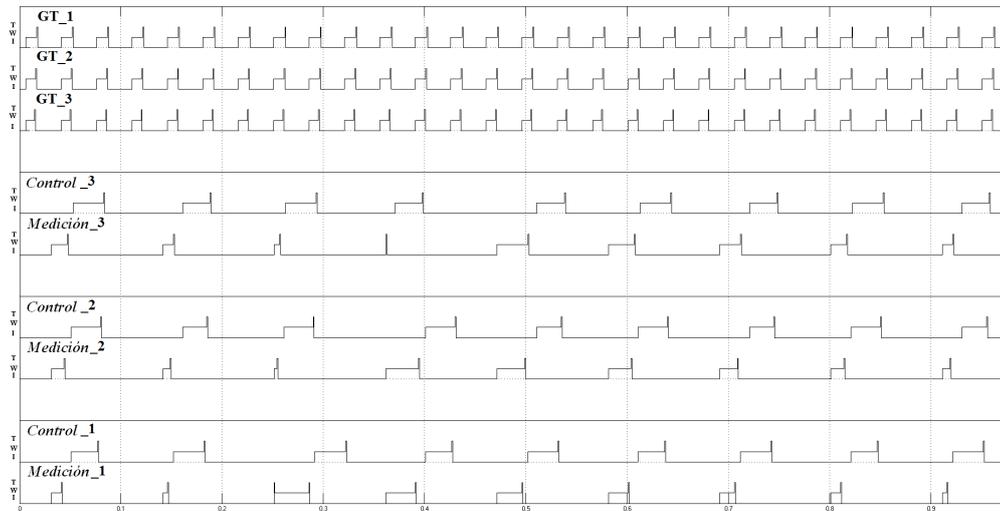


Figura 4.4. Planificación de los mensajes en la red TDMA. T, W e I indican los estados *Transmitting*, *Waiting* e *Idle* de cada uno de los transmisores de los nodos

Se puede apreciar que la transmisión de mensajes de cada uno de los nodos no es periódica, y depende del instante en que finalice la ejecución de la tarea que genera el dato a ser enviado y la sincronización de este instante con el *slot* destinado para el envío de su información en la red, sin embargo este valor está acotado. Este resultado es muy importante porque indica que en cada nodo los procesadores encargados de la

comunicación no necesitan estar sincronizados con aquellos que desarrollan la aplicación.

Además, a pesar de que el tiempo de transmisión en la red no sea constante sí se cumple con los plazos extremo-extremo, lo cual se puede apreciar en la figura 4.5, donde la tarea *Actuación* del lazo de control 1 finaliza antes de cumplirse el plazo de 80 ms impuesto por los objetivos de control.

En la figura 4.6 se observa que el desempeño de los tres lazos de control es muy similar, y en todos los casos se cumple con el comportamiento deseado.

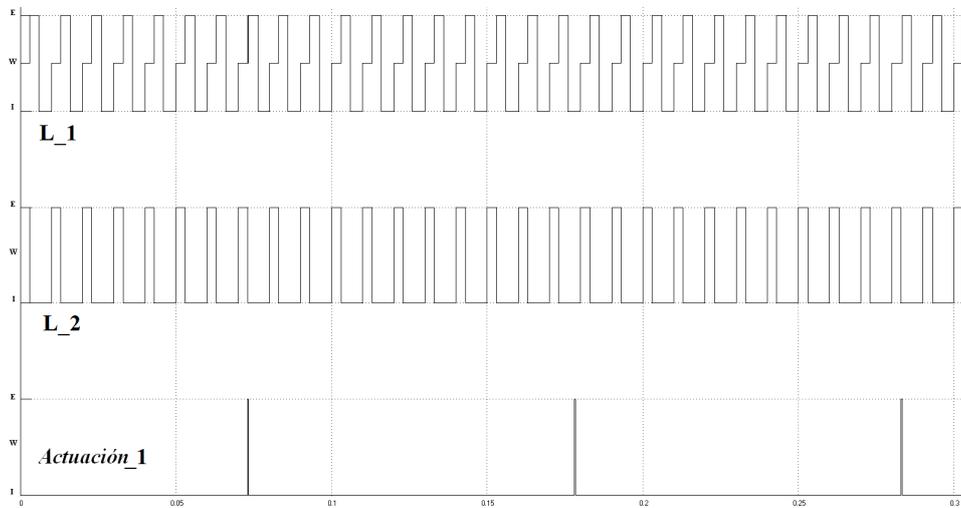


Figura 4.5. Planificación de tareas en el nodo que ejecuta la tarea *Actuación* del lazo de control 1. E, W e I indican los estados *Execution*, *Waiting* e *Idle* de cada tarea.

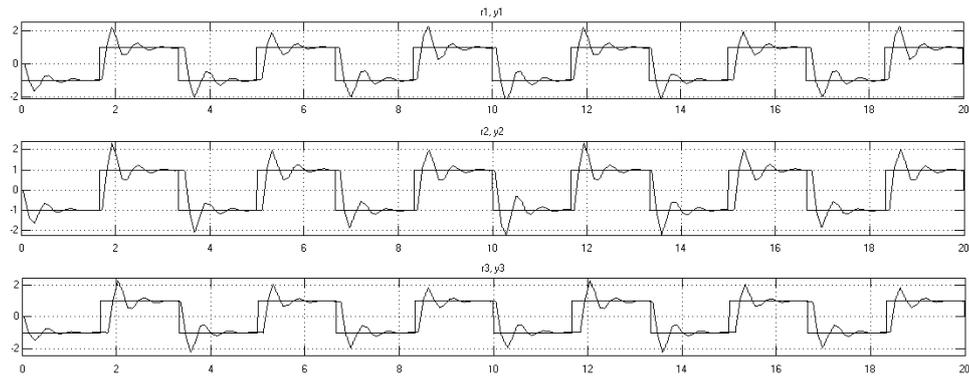


Figura 4.6. Respuesta del los tres lazos de control para una referencia tren de pulsos y una red TDMA.

Para comparar el desempeño de los lazos de control empleando una red CSMA-CA, se implementó la simulación del caso de estudio empleando una red Zigbee con los mismos parámetros de velocidad de comunicación y tamaños de tramas empleados para el análisis de la red TDMA. Los resultados obtenidos se presentan en las figuras 4.7 y 4.8.

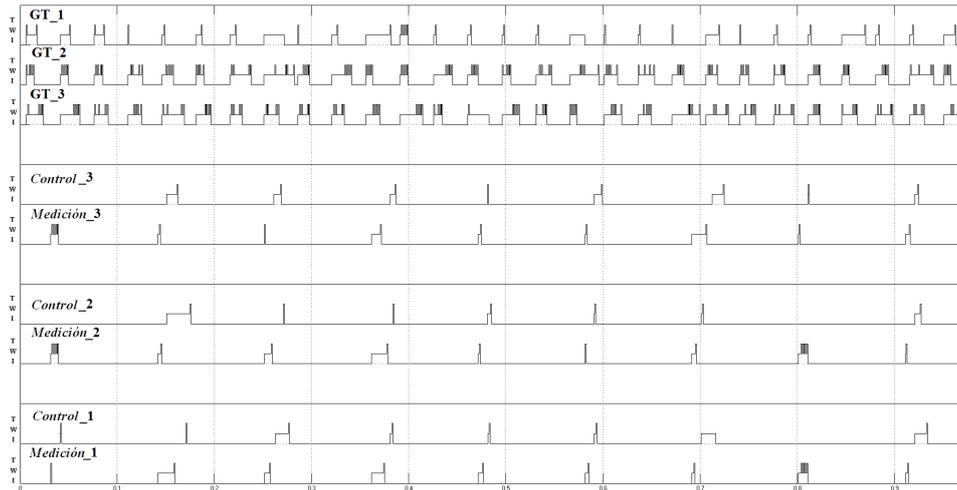


Figura 4.7. Planificación de los mensajes en la red Zigbee

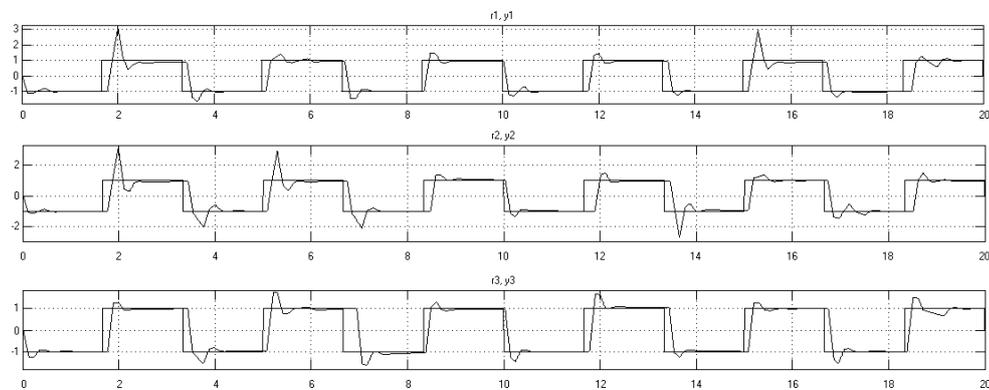


Figura 4.8. Respuesta de los tres lazos de control para una referencia tren de pulsos y una red Zigbee

En la figura 4.7 se puede apreciar que en algunos periodos la latencia generada por la red CSMA-CA es menor que la generada por la red TDMA, obteniendo una mejor respuesta del sistema de control en dichos instantes, como se aprecia en la figura 4.8. Sin embargo en otros periodos el retraso supera el plazo extremo-extremo propuesto de 80 ms, debido a la colisión de mensajes en la red, con lo que se degrada considerablemente la respuesta del sistema respecto del comportamiento deseado.

Los resultados obtenidos permiten concluir que:

- La utilización del test de planificabilidad propuesto permiten diseñar WSN que cumplan con las restricciones de tiempo impuestas por los algoritmos de control.
- Para garantizar el cumplimiento de los plazos extremo-extremo no es necesario que en cada nodo los procesadores encargados de la comunicación estén sincronizados con aquellos que desarrollan la aplicación.
- El uso de protocolos CSMA-CA, como el empleado en Zigbee para la comunicación en redes con arquitectura mesh, no garantiza una latencia máxima en el retraso presente en el sistema de control, lo cual puede degradar considerablemente el desempeño de estos sistemas.

4.3 Análisis de planificabilidad extremo-extremo para WSN multisalto

En este caso si se tuvo en cuenta el almacenamiento de mensajes en buffers, por cuanto los mensajes realizan varios saltos desde que son enviados por el emisor hasta alcanzar el destino final de la información, figura 4.9, luego en un mismo nodo intermedio pueden llegar a coincidir más de un mensaje en un determinado instante. Sin embargo, teniendo en cuenta la cantidad de datos enviados en estas aplicaciones y la cantidad de datos que es posible enviar en cada mensaje de los protocolos actuales, nuevamente se considera que toda la información que se desea enviar es transmitida en un mismo mensaje.

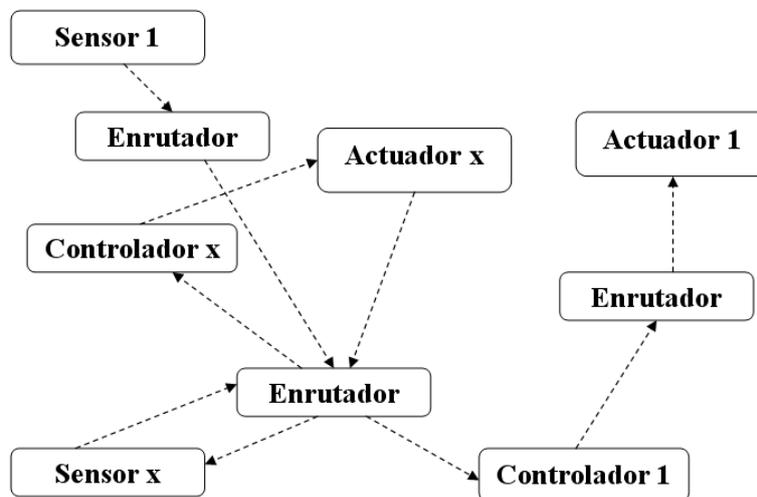


Figura 4.9. Envío de información en modo multisalto

Las consideraciones realizadas para calcular el tiempo máximo de permanencia de un mensaje en un nodo enrutador, y la recepción del mismo por el nodo siguiente en el flujo de información, son:

- Se debe cumplir con el plazo de todos los mensajes.
- Se considera el almacenamiento de mensajes en los buffers de los nodos.
- Cada nodo tiene asignada una ranura de tiempo para acceder al canal.
- Toda la información que se desea enviar es transmitida en un mismo mensaje, y no se presenta expropiación de la red una vez se ha iniciado el envío de un mensaje.
- El orden de envío de los mensajes almacenados en un nodo enrutador sigue una política basada en EDF. Además se supone que la elección del mensaje más prioritario se realiza antes de que se presente la ranura de tiempo asignada al nodo.
- El plazo asignado a cada salto del mensaje, para un determinado lazo de control, es:

$$D_{MLoop_i} = \frac{D_{CGRi} - (D_{M_i} + D_{C_i} + D_{A_i})}{N}$$

Donde N es el número de saltos que hay en el camino de la información entre medidor – controlador y controlador – actuador.

- Durante la ranura asignada al nodo i cada $T_{TDM_A_i}$, éste nodo solamente envía uno de los mensajes que tiene almacenados.
- El máximo tiempo de bloqueo que puede presentar el mensaje de mayor prioridad es $T_{TDM_A_i} + slot$, lo cual es consecuencia de que el procesador donde se ejecutan las tareas y el coprocesador encargado de la comunicación no se encuentran sincronizados.
- Para cada lazo de control j , los mensajes son generados por cada tarea de medición cada T_{S_j} .

Con base en el test de planificabilidad de tareas presentado en (Ripoll, 1996), y realizando una analogía entre mensajes y tareas, se definen las siguientes funciones:

- $G_{\tau_N}(t) = \left(\sum_{i=1}^m T_{TDM_A_k} \left\lceil \frac{t}{T_{S_i}} \right\rceil \right) + slot$, es la cantidad de tiempo requerido para el envío de m mensajes almacenados en el nodo enrutador k , medido desde el instante cero.

Desde las propiedades de las funciones *ceiling* y *floor* se puede asegurar que $G_{\tau_N}(t)$ es una función monótona no decreciente.

- Intervalo crítico inicial de la red, (ICI_N), es el intervalo de tiempo entre cero y el primer instante en el que no hay mensajes pendientes de ser enviados en un enrutador, es igual a $[0, R_N)$. Se calcula de forma recursiva así:
 - $K_{N_0} = 1; \quad K_{N_{i+1}} = G_{\tau_N}(K_{N_i})$
 - Si $K_{N_{i+1}} = K_{N_i} \Rightarrow R_N = K_{N_i}$
- $H_{\tau_N}(t) = \sum_{i=1}^m T_{TDM_k} \left\lfloor \frac{t + T_{S_i} - D_{MLoop}}{T_{S_i}} \right\rfloor + slot$, es la cantidad de tiempo que tiene que haber sido servida por la red al nodo enrutador hasta el tiempo t , para cumplir con los plazos de tiempo de los mensajes que este nodo debe enviar. Como $T_{S_i} \geq D_{MLoop}$, $H_{\tau_N}(t)$ es una función definida positiva.

Suponiendo que las solicitudes de envío se ordenan en orden ascendente en relación con el plazo, los valores de $H_{\tau_N}(t)$ pueden ser calculados como:

$$\begin{aligned}
 T_{TDM_k} + slot &= H_{\tau_N}(D_{MLoop_1}) \\
 2T_{TDM_k} + slot &= H_{\tau_N}(D_{MLoop_2}) \\
 3T_{TDM_k} + slot &= H_{\tau_N}(D_{MLoop_3}) \\
 &\vdots \\
 xT_{TDM_k} + slot &= H_{\tau_N}(D_{MLoop_x})
 \end{aligned}$$

Por tanto, si el tiempo demandado para enviar los mensajes almacenados en un enrutador durante el periodo de mayor saturación del nodo, es menor al plazo del que dispone para el envío en cada uno de los lazos de control, el sistema cumplirá con todas las restricciones temporales. Luego, el test de planificabilidad extremo-extremo consiste en verificar el cumplimiento de los plazos de las tareas en cada nodo empleando la ecuación (2), y los plazos de los mensajes en cada uno de los saltos que realizan, para cada lazo de control, así, (Martínez, 2009c):

$$H_{\tau_N}(t) < t \quad \forall \quad t \leq R_N \quad (4)$$

4.4 Validación de propiedades estructurales y de comportamiento en WSAN

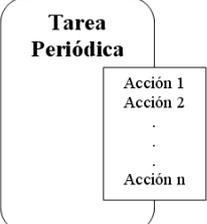
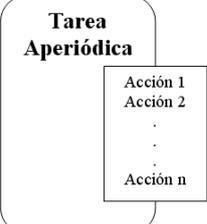
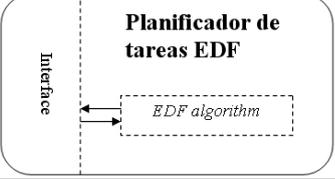
Para realizar la validación de las propiedades estructurales y de comportamiento de las WSAN, se desarrolla un modelo del sistema en CPN y posteriormente se verifican propiedades sobre el mismo.

4.4.1 Componentes

Con base en la arquitectura propuesta para los nodos, se propone un conjunto mínimo

de componentes para el diseño de estas aplicaciones, tabla 4.3:

Tabla. 4.3. Conjunto mínimo de componentes para el diseño de WSA

Conjunto mínimo de componentes para el diseño de WSA		Descripción
Actividades. Estos componentes representan el comportamiento de las diferentes actividades del sistema.		Actividades cíclicas con intervalos de tiempo constante.
		Actividades iniciadas por la ocurrencia de eventos externos.
Elementos de implementación. Son elementos necesarios para establecer la interacción entre las actividades		Representa el comportamiento de un planificador de tareas EDF, y permite analizar la incidencia de la expropiación en el tiempo de finalización de las actividades.
	<i>TDMA – 35 ranuras con 2 GTS</i>	Representa el modo de acceso al medio TDMA y permite analizar los retrasos sufridos en la transferencia de información a través de la red.

4.4.2 Validación de propiedades estructurales y de comportamiento del diseño

Un método para verificar el funcionamiento de un sistema consiste en desarrollar un modelo ejecutable del mismo. Por medio de la simulación del modelo es posible analizar cómo será el desempeño del sistema. Adicionalmente, el desarrollo de este tipo de modelos permite obtener una especificación completa que facilita una investigación sistemática de escenarios, con lo cual se puede disminuir significativamente el número de errores de diseño.

Las redes de Petri son una herramienta gráfica y matemática para la descripción y el estudio de sistemas que se caracterizan por ser concurrentes, asíncronos, distribuidos, paralelos, no deterministas y/o estocásticos. Por medio de la cual es posible describir y analizar secuencias, conflictos y concurrencia.

El concepto de red de Petri surge en 1962 en la Facultad de Matemáticas y Física en la Universidad Técnica de Darmstadt, en la disertación de Carl Adam Petri titulada “Comunicación con Automatas”. Desde entonces las redes de Petri se han estudiado de

diferentes maneras en sistemas concurrentes, varias extensiones se han definido que permiten tratar un gran número de aspectos entre los que se pueden citar tiempos, determinismo, datos, etc. Desarrollándose en forma paralela métodos de análisis de las extensiones propuestas. Entre las extensiones de redes de Petri se pueden citar las siguientes por su importancia y difusión: generalizadas, con arcos inhibidores, temporizadas, estocásticas, a predicados, coloreadas, ER, TER y TB.

Las redes de Petri permiten representar sistemas a cualquier escala de complejidad. Durante la etapa de diseño estos modelos pueden ser utilizados para realizar pruebas que en el prototipo final resultarían inadecuadas, bien sea por costes, tiempo, ubicación física, etc.

Otro aspecto importante de las redes de Petri se centra en la posibilidad de modelar en un mismo lenguaje los diferentes subsistemas de un gran sistema, facilitando la interacción de diferentes expertos en el desarrollo de aplicaciones complejas, como lo son las WSN. Esto es de gran importancia porque debido al estado actual de las herramientas de análisis y desarrollo, no siempre se cuenta con herramientas que integren todos los niveles de una arquitectura, como son CPU, sistema operativo, red de comunicaciones, middleware, coprocesadores, y las funcionalidades propias del contexto de la aplicación. Lo que conlleva a desarrollar análisis y desarrollos parciales de los subsistemas que posteriormente son integrados, dificultando el tener una visión general del sistema durante las fases de diseño, con lo cual se incrementa el riesgo de presentar fallos de funcionamiento.

Por otro lado, aunque algunos entornos de simulación permiten integrar los diferentes niveles de la arquitectura del sistema, no siempre los modelos de estos componentes se ajustan a las características y el comportamiento de los componentes de síntesis. Adicionalmente, dependiendo de la magnitud del sistema, la validación por medio de simulaciones o ejecución directa de las actividades no siempre considera todos los posibles estados del mismo, por tanto se pueden quedar condiciones sin verificar.

Las redes de Petri permiten verificar las siguientes propiedades:

- De comportamiento (aquéllas que dependen del marcado inicial): *Alcanzabilidad, Boundedness, Safeness y Liveness, Reversibilidad, Coverability, Persistencia, Fairness.*
- Estructurales (las cuales dependen únicamente de la topología de la red de Petri y son independientes del marcado inicial): *Liveness estructural, Controlabilidad, Boundedness estructural, Repetitividad y Consistencia.*

Entre las técnicas de análisis se encuentran:

- Método del árbol de cobertura (alcanzabilidad). Es aplicable a toda clase de redes, pero se limita a redes pequeñas debido a la explosión del espacio de estados.
- Matriz de incidencia. La cual es muy adecuada para analizar propiedades estructurales.

- Técnicas de reducción o descomposición. Debido a su generalidad, los modelos basados en redes de Petri se convierten en excesivamente grandes para el análisis incluso para sistemas sencillos. Por ello, a menudo es necesario añadir modificaciones o restricciones cuando se usan en una aplicación particular. Para facilitar el análisis de sistemas grandes se puede reducir el modelo del sistema a uno más simple preservando las propiedades a analizar. Se pueden usar técnicas para transformar un modelo abstracto en uno más refinado de una forma jerárquica.

Las redes de Petri coloreadas (*Coloured Petri Nets*, CP-nets o CPN) poseen marcas cuyo valor representa un dato perteneciente a un determinado tipo, (Jensen, 1994), (Jensen, 1997). Estos modelos formales pueden ser utilizados para verificar las propiedades del sistema, para lo cual se soportan en un conjunto de métodos de espacio de estado. Lo que permite verificar, entre otros aspectos, la ausencia de bloqueos y la posibilidad del sistema para evolucionar a un determinado estado. Estos análisis también pueden ser aplicados a CPN temporizadas, en las que además del dato se agrega un campo para representar una marca de tiempo, lo que posibilita la verificación de aspectos temporales del sistema (Jensen, 2007).

La principal desventaja de los métodos de análisis soportados en espacio de estado es el problema de la explosión de estados, para abordar este problema se han generado herramientas informáticas como CPN Tools, (Jensen, 2007), que incluyen técnicas de reducción que acotan el problema de la explosión de estados.

4.4.3 Representación de componentes en CPN

En las figuras 4.10 y 4.11 se presentan las representaciones en CPN de los componentes periódico y aperiódico, realizadas en la herramienta CPN Tools, (Martínez, 2011). Estos representan el comportamiento de las tareas periódicas y aperiódicas descritas en lenguajes C y ensamblador, utilizados actualmente para desarrollar aplicaciones de WSN.

En el objeto periódico, los lugares *In_P* y *End_P* representan los estados de inicio y fin de la actividad periódica respectivamente, mientras que la transición *Act* representa el desarrollo de las acciones asociadas a éste componente; si es necesario se pueden colocar más transiciones para representar el desarrollo de las acciones de forma independiente. La transición *Re_p* y el lugar *Re_p*, indican cuándo la tarea está lista para una nueva ejecución. Finalmente la transición *Period*, indica el cumplimiento del periodo de ejecución del componente.

La representación del componente aperiódico es muy similar a la del componente periódico, simplemente que la activación de este componente se realiza colocando una marca en el lugar *Event*, lo que genera el disparo de la transición *Event*.

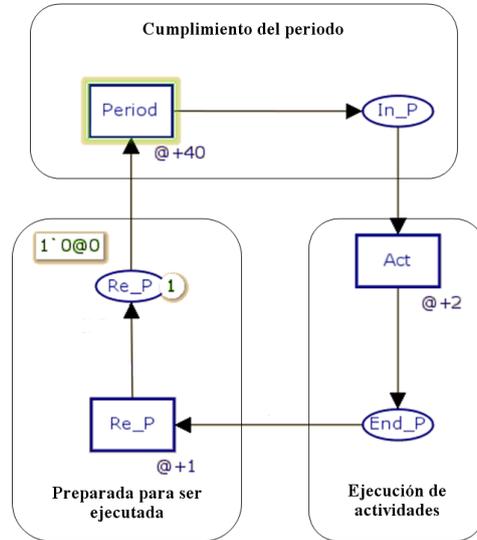


Figura 4.10. Representación en CPN de un componente periódico

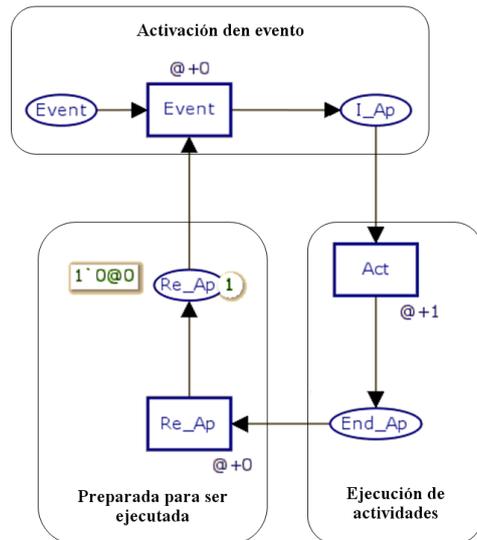


Figura 4.11. Representación en CPN de un componente aperiódico

En la figura 4.12 se presenta un esquema general de la interacción entre una tarea y el componente planificador de tareas, y en la figura 4.13 se presenta el modelo en CPN del planificador de tareas EDF, el cual utiliza una *Interface* para interactuar con las tareas, a través de la cual le indica a cada tarea cuándo ha ganado la contienda por el procesador (lugar *CPU*), o el momento en que deben finalizar su ejecución.

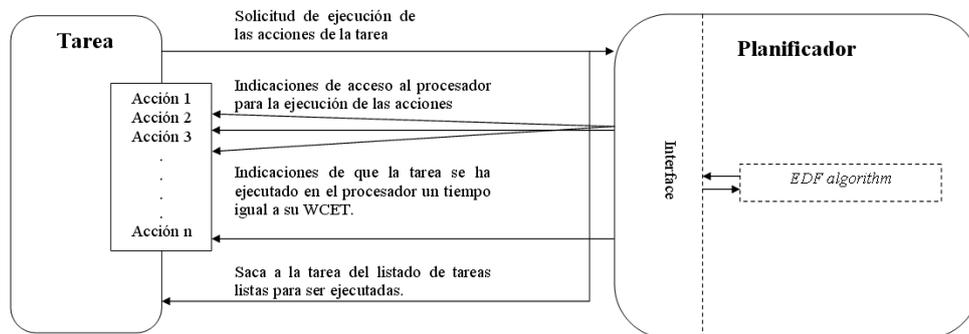


Figura 4.12. Esquema general de la interacción entre una tarea y el planificador de tareas

En este caso se presenta la ejecución de dos tareas periódicas. Las marcas en las tareas tienen 4 campos ($name, d, WCET, tt$), que representan en su orden el nombre de la tarea, el plazo de ejecución, el tiempo del peor caso de ejecución, y el tiempo que la tarea se ha ejecutado en el procesador.

Durante cada intervalo de reloj los plazos de las tareas que están listas para ser ejecutadas son comparados dentro de la zona *EDF algorithm*, incluida la tarea que pueda estar en ejecución en ese momento, y se envía a ejecución en el procesador la de menor plazo.

En la *Interface* se reciben las solicitudes de ejecución de las tareas y se implementa un método de detección de bloqueos en las tareas, que finaliza la ejecución de una tarea cuando su tiempo de permanencia en la CPU supere su *WCET*. La tabla 4.4 presenta la secuencia de marcas en el lugar CPU durante la simulación del modelo.

Tabla 4.4. Secuencia de marcas en el lugar CPU entre los instantes de tiempo 40 y 43

Instante de tiempo	Marca
40	"Task1",5,2,0
41	"Task1",4,2,1
42	"Task2",4,2,0
43	"Task1",3,2,1

En la figura 4.14 se presenta el esquema general del acceso de dos nodos a una red TDMA con 35 ranuras y dos GTS, ranuras 1 y 2, los cuales están asignados a los nodos 1 y 2 respectivamente, y en la figura 4.15 se muestra su modelo en CPN. Este modelo se asemeja al comportamiento de una supertrama del protocolo IEEE 802.15.4, aunque a diferencia de esta se realiza la asignación de las ranuras al inicio de la supertrama y no se contemplan los intervalos de acceso por contienda y periodo inactivo.

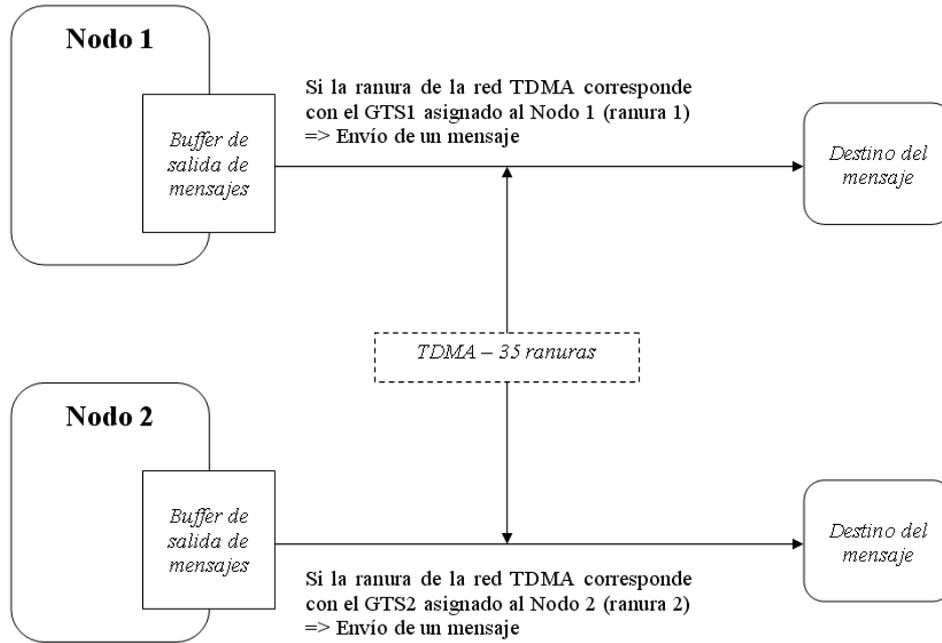


Figura 4.14. Esquema general de acceso de una red TDMA con 35 ranuras y dos GTS

Los lugares *Buff-M* y *BuffC*, representan los lugares de entrada a los buffer de los nodos de la red. Las transiciones *Buff_M* y *Buff_C* se activan en el instante de tiempo en que el nodo, 1 o 2, tiene algún dato para transmitir y el itinerario de ranuras en la red corresponde con su respectivo GTS. Los lugares *IN_Act* e *IC* representan los buffer de entrada de los nodos a los que va dirigida la información.

Actualmente existe un gran interés por el desarrollo de técnicas soportadas en modelos arquitecturales y estrategias de abstracción que permitan a los diseñadores del sistema modelar, evaluar y verificar estas características a gran escala, a partir de modelos de sistemas basados en componentes, (Tariq, 2011). El conjunto de componentes y modelos en CPN presentados en este apartado contribuye a disminuir la complejidad del diseño de WSN, y facilita su verificación y análisis.

Capítulo 5

METODOLOGÍA DE DISEÑO DE WSAN

La metodología de diseño propuesta se presenta en la figura 5.1, (Martínez, 2010b).

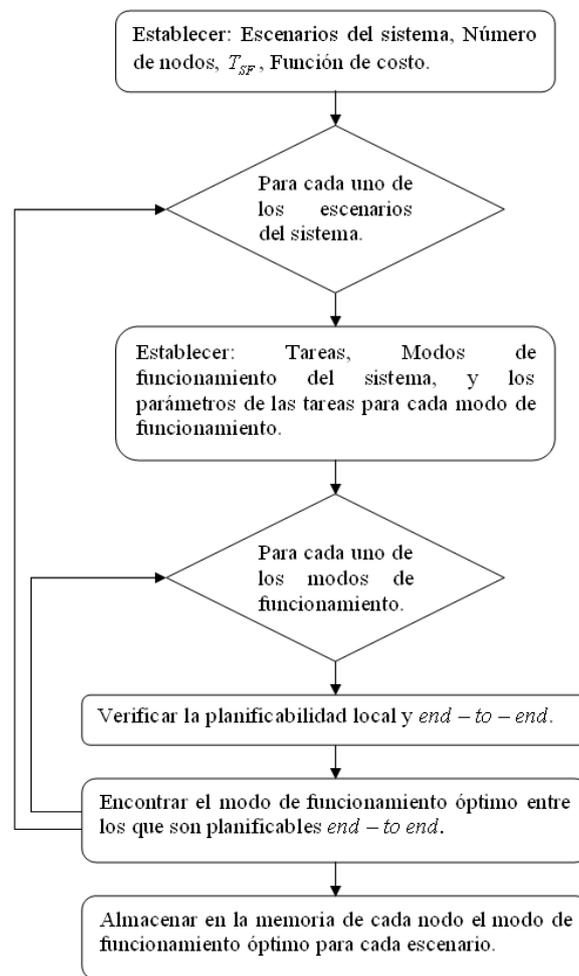


Figura 5.1. Metodología de diseño de WSAN

Como caso particular, se abordó el diseño de aplicaciones de NCS implementados sobre WSAN. Las consideraciones generales de la metodología son:

- El número y ubicación de los nodos de la red es conocido.
- Debido al elevado tiempo de cómputo requerido para evaluar la planificabilidad y la configuración del sistema, estos análisis se realizan fuera de línea. En este orden de ideas, todos los escenarios de la aplicación y la configuración óptima del sistema para cada uno (modos de funcionamiento) se suponen conocidos, y se almacenan en la memoria de cada nodo para ser utilizados durante el funcionamiento del sistema en el momento en que se genere cada uno de los escenarios. Los diferentes escenarios se generan a partir de cambios de modo de las tareas, creación y destrucción de tareas, y cambios en el número y ubicación de los nodos.
- Se desprecia la latencia entre los cambios de modo de funcionamiento.
- La elección del mejor modo de funcionamiento del sistema para cada uno de los escenarios se obtiene a partir de un proceso de optimización, para el cual se utilizan funciones de costo.
- Para la activación de tareas en cada escenario de la aplicación, se supone que existe una réplica de cada una de las tareas que no tienen una asignación preestablecida en cada nodo en los que pueden ser ejecutadas en función del escenario del sistema. Esta consideración se puede establecer debido a que para estas aplicaciones la cantidad de tareas en cada nodo y el tamaño de código de las mismas no es elevado y puede ser almacenado en los nodos actuales.
- Las tramas son transmitidas y recibidas libres de error.
- El tiempo requerido para los cambios de contexto de las tareas es considerado como parte del tiempo de cómputo de las mismas.
- El procesador principal y el coprocesador de cada nodo no se encuentran sincronizados entre sí.

A continuación se describen los pasos del procedimiento de diseño:

- Para establecer el número de nodos requeridos para implementar la solución, se considera principalmente el alcance de los sistemas de comunicaciones de los nodos y la distribución física de las variables del sistema, lo que determina el área de cobertura de la red.
- El valor de T_{SF} está relacionado con la máxima frecuencia de muestreo presente en la aplicación, el plazo de entrega del valor de esta variable y el número máximo de saltos que pueda tener el camino entre fuente y destino de la información. Por ejemplo, para un lazo de control, si las funciones de medición, control y actuación se encuentran en nodos independientes, los cuales a su vez se encuentran a un salto de distancia en la red, un criterio para seleccionar T_{SF} es:

$$T_{SF} \leq \frac{D_{CGR} - (WCET_M + WCET_C + WCET_A)}{2}.$$

- La función de costo, utilizada para medir la calidad del modo de funcionamiento para cada uno de los escenarios de la aplicación, depende de las características y restricciones de cada aplicación. En este trabajo se considera que tanto el consumo de potencia como los retrasos en el lazo de control, son importantes para definir la calidad de los diferentes modos de funcionamiento del sistema.
- Los periodos y plazos de las tareas están definidos por la aplicación. Para obtener los tiempos de cómputo para cada una de las frecuencias de funcionamiento de los procesadores de los nodos, se considera una arquitectura de procesador en la que al escalar la frecuencia de operación del procesador también se escalan los ciclos de lectura y escritura en memoria, y se suponen conocidos los WCET de las tareas para la frecuencia más baja; luego se calculan los WCET para las demás frecuencias empleando un factor de escalado de frecuencia, λ , por lo tanto:

$$f_i = \lambda_i * f_{baja}, \text{ luego } WCET_{f_i} = \frac{1}{\lambda_i} WCET_{f_{baja}}.$$

Esta suposición se puede establecer en procesadores XScale, para los cuales es posible obtener diferentes frecuencias de funcionamiento a partir de multiplicar por potencias de dos la frecuencia de reloj del sistema.

Las frecuencias seleccionadas para definir los modos de funcionamiento en cada nodo deben ser elegidas de tal forma que los WCET de las tareas que realizan la medición y el control sean del orden de magnitud o cercano al de T_{SF} , lo anterior debido a que la red actúa como filtro y por lo tanto el finalizar estas actividades muy rápido no implica que el retraso total se disminuya considerablemente, y se podría estar realizando un consumo de potencia innecesario. La disminución del WCET de la tarea de actuación si afecta de forma más directa el retraso total del sistema.

Adicionalmente, es importante considerar que algunos periféricos operan con señales de reloj externas a las del procesador y por lo tanto el escalado en la frecuencia del procesador no alterará su tiempo de respuesta.

- Con el propósito de disminuir al mínimo el consumo de potencia, mientras no hayan mas activaciones pendientes el procesador se llevará al estado de menor consumo que no implique el reinicio del mismo, y del cual salga al presentarse una interrupción.
- Para verificar el cumplimiento de los plazos temporales, en función de la arquitectura de la red, se utiliza uno de los dos métodos de análisis de planificabilidad presentados en el capítulo 4.

- Dependiendo de los parámetros definidos en la función de costo, durante la búsqueda del modo de funcionamiento óptimo puede ser necesario la utilización de herramientas de simulación, para la obtención de los valores que toma la función de costo para cada modo de funcionamiento.

5.1 Funciones de costo

Un proceso de optimización intenta encontrar el mejor entre un conjunto de elementos. Existen diferentes técnicas para la búsqueda de soluciones óptimas, las cuales se diferencian respecto al tipo de problema que pueden abordar como también a su coste computacional. El punto de partida de un proceso de optimización, y uno de sus principales problemas, es definir correctamente la función de costo.

En este trabajo la función de costo debe reflejar parámetros relacionados con el consumo de potencia y los retrasos en las WSAN, y el objetivo de la optimización es encontrar el modo de funcionamiento del sistema que minimiza el valor de la función de costo y que cumple con las restricciones de tiempo real de la aplicación. Como se puede apreciar, el espacio de los parámetros de la función de costo difiere del espacio de las restricciones del problema. Lo que llevó a encontrar el subconjunto de modos de funcionamiento del sistema que cumplen con las restricciones, y posteriormente elegir el mínimo entre ellos.

La generalidad del problema en cuanto al número y tipos de nodos, número de tareas y modos de funcionamiento del sistema, dificultan su representación y solución a través de métodos analíticos; la función toma valores discretos y dependiendo del tamaño de la aplicación se puede tener un número elevado de valores, por lo cual en algunos casos puede ser necesario el uso de métodos computacionales de búsqueda heurísticos, tales como el *Simulated Annealing*, Algoritmos genéticos, entre otros. El problema que presentan estos métodos es que no siempre convergen a la solución óptima, por lo que en los casos abordados en esta tesis se analizaron todas las posibles configuraciones del sistema y se eligió la que minimiza la función de costo, sin embargo esto puede no ser viable en aplicaciones donde la función de costo toma un número muy elevado de valores.

En términos generales, los objetivos planteados en la optimización están relacionados con la calidad del diseño del sistema de cómputo, representado en este caso a partir del consumo de potencia, y la calidad de la respuesta generada por los algoritmos de control, la cual se relaciona con los retrasos y jitter en el lazo de control, cuya importancia individual depende de las particularidades del contexto en el que se desempeña la aplicación.

Para analizar cada uno de los objetivos se han planteado dos funciones de costo, $F(Co)_{S_Cómputo}$ para evaluar el costo de un modo de funcionamiento considerando parámetros relacionados con el sistema de cómputo, y $F(Co)_{Control}$ para evaluar la calidad de parámetros que afectan las características dinámicas del sistema de control, cada una de las cuales tiene la siguiente forma:

$$F(C_o) = \sum_0^i K_i \frac{C_i(C_o)}{C_i}$$

K_i es el peso asignado al parámetro i ; $C_i(C_o)$ es la solución obtenida de un determinado modo de funcionamiento C_o ; C_i es la restricción de diseño aplicado al i -ésimo parámetro considerado en la función y es usada como parámetro de normalización.

Con relación al sistema de cómputo el objetivo es generar una solución óptima que aumente el tiempo de vida de la WSN, considerando una arquitectura para los nodos como la propuesta en el capítulo 3, lo cual está restringido principalmente por la disponibilidad de energía en cada uno de los nodos de la red. Esto se logra empleando la frecuencia de operación más baja para cada nodo, pero logrando un porcentaje de utilización similar entre los nodos. Las posibles acciones para incidir en el consumo de potencia son:

- Escalado de frecuencia y voltaje del procesador. Actúa sobre el consumo de potencia en un único nodo y escala los $WCET_i$ y los $WCRT_i$ en el respectivo nodo.
- Migración de tareas entre los nodos, lo cual se logra por la activación de la tarea respectiva en el nodo seleccionado, para lo cual se considera que en cada nodo existe una réplica de cada una de las tareas. Busca equilibrar el consumo de potencia entre los nodos de la WSN y en general afecta los $WCRT_i$ en varios nodos.

En el enfoque empleado en este trabajo, por cuanto se conoce el WCET de las tareas y las diferentes frecuencias de operación de los nodos, y puesto que el consumo de potencia está relacionado directamente con estos parámetros, se realiza un análisis indirecto del consumo de potencia de las diferentes configuraciones de operación del sistema, empleando los siguientes índices:

- fop_{M_j} , es la media entre las frecuencias de operación de los nodos que conforman la WSN para un modo de funcionamiento j . Se emplea para encontrar el modo de funcionamiento donde los nodos operen a la menor frecuencia posible.
- Δfop_{DEj} , es la desviación estándar de las frecuencias de operación de los nodos. Se utiliza para buscar un consumo de potencia equilibrado entre los nodos.
- ΔU_{DEj} , es la desviación estándar de los porcentajes de utilización de los nodos. Se utiliza para buscar un consumo de potencia equitativo entre los nodos. Donde el porcentaje de utilización de un nodo que ejecuta i tareas es:

$$U = \sum_{i=1}^n \frac{WCET_i}{T_i}$$

Respecto del desempeño del sistema de control se consideraron dos parámetros. Estos son:

- r_M : Valor medio de los retrasos τ_r , medidos desde el instante de activación de la tarea que realiza la medición de las variables para calcular la acción de control y el instante de finalización de la tarea de actuación.
- Δr_{DE} : Es la desviación estándar entre los retrasos τ_r . Idealmente debe tender a cero, de tal forma que el retraso pueda ser compensado más fácilmente en el algoritmo de control.

Junto con los parámetros a analizar se establecen los valores máximos de operación. Estos valores dependen de la arquitectura de síntesis de componentes que se ha seleccionado, las restricciones planteadas por la aplicación y los parámetros de cada modo de funcionamiento. Estos son:

- $fop_{M_j}^{\max}$, valor promedio máximo de los fop_{M_j} .
- Δfop_{DEj}^{\max} , máxima desviación estándar entre las frecuencias de operación de los nodos.
- ΔU_{DEj}^{\max} , máxima desviación estándar entre los porcentajes de utilización de los nodos.
- r_M^{\max} , valor máximo de r_M , este valor podría ser menor que aquel con el que se realizan los análisis de planificabilidad, de tal forma que se elijan soluciones cuyo retraso extremo-extremo afecte menos la respuesta del algoritmo de control.
- Δr_{DE}^{\max} . Máxima desviación estándar entre los retrasos presentes en el sistema.

La metodología propuesta permite encontrar el modo de funcionamiento óptimo en la aplicación, considerando parámetros del sistema de cómputo y el desempeño de sistemas de control en WSAN. Esta se fundamenta en dos funciones de costo relacionadas con el consumo de potencia y los retrasos en los lazos de control, cuyos parámetros dependen de las características y restricciones de cada aplicación, y emplea una combinación lineal de pesos para encontrar la distribución óptima de los componentes del sistema. De igual forma integra la arquitectura de genérica propuesta para los nodos y los métodos de verificación que permiten garantizar el cumplimiento de las restricciones de tiempo real presentes en este tipo de aplicaciones. Con lo que es posible aumentar el tiempo de vida de estas aplicaciones y cumplir con los objetivos de diseño establecidos en la respuesta de los sistemas de control.

Capítulo 6

PRUEBAS Y VALIDACIÓN

En este capítulo se presentan los diferentes pasos de la metodología y sus verificaciones. Para presentar el uso de la metodología de diseño propuesta se abordaron dos casos de estudio, en los que se implementan NCS.

Para simplificar el espacio de soluciones a los casos de estudio, se suponen nodos homogéneos con dos frecuencias de operación del procesador, baja y alta. Con el objetivo de enfatizar el efecto de los diferentes modos de configuración en el consumo de potencia, un factor de escalado de frecuencia, λ , de 10 fue considerado, por lo tanto:

$$f_{alta} = 10f_{baja}, \text{ luego } WCET_{alta} = \frac{1}{10}WCET_{baja}$$

En el cálculo de los WCET de las tareas de medición para la frecuencia alta, se añade un valor correspondiente al tiempo de cómputo de los ADC, el cual no se escala.

Los parámetros de configuración de la red para los casos de estudio analizados en este capítulo son:

- Tramas de igual tamaño, 144 bits.
- Velocidad de comunicación de 250 kbps.
- Duración de cada ranura en la red TDMA de $1ms$.

6.1 Caso de estudio: WSN con distancia entre nodos acotada a un único salto

En este caso se considera una red formada por tres nodos, sobre los cuales se implementan dos aplicaciones, un NCS y un sistema de detección de fallos en línea, (Martínez 2010a). La figura 6.1 presenta un diagrama de los componentes de cada una de las aplicaciones.

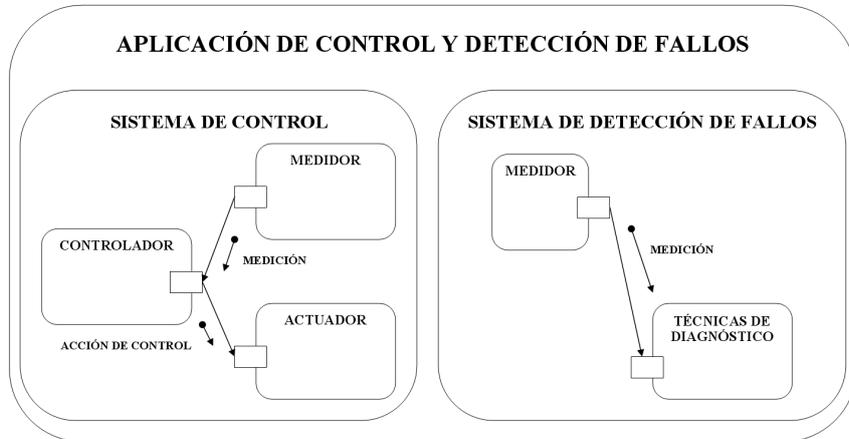


Figura 6.1. Diagrama de componentes del caso de estudio: WSN con distancia entre nodos acotada a un único salto

El periodo con el que se repite el itinerario de las ranuras en la red TDMA es de $35ms$.

En este caso se supone que los nodos se comunican a través de una WSN con distancia entre nodos acotada a un único salto, por lo que se emplea el test de planificabilidad extremo-extremo presentado en el apartado 4.2.

6.1.1 Consideraciones del sistema de detección de fallos

En el caso considerado, se supone que la degradación del sistema debida a algún fallo no se produce rápidamente, por lo que esta aplicación se realizará una vez por día, además no requiere de un plazo de respuesta muy exigente.

Esta aplicación está compuesta por dos tareas. La tarea que captura las muestras necesarias para ejecutar los algoritmos de detección de fallos se denomina *D_Medidor*, es periódica y se ejecuta hasta adquirir el número de muestras necesarias por los algoritmos. La otra tarea se ejecuta una sola vez y realiza los algoritmos de detección de fallos, se denomina *Técnicas_de_diagnóstico*, esta tarea no tiene restricciones de tiempo real por lo que el objetivo es finalizar su ejecución tan pronto como sea posible; para efectos de análisis se ha considerado un plazo de 5 s para esta tarea.

Se supone que de acuerdo a la configuración física del sistema, la tarea *D_Medidor* estará preasignada al nodo 1, mientras que la tarea *Técnicas_de_diagnóstico* puede ser ejecutada en cualquiera de los tres nodos.

Los tiempos de cómputo supuestos para *D_Medidor* y *Técnicas_de_diagnóstico* se presentan en la tabla 6.1.

Con base en lo anterior, el diagnóstico se realiza cada $24 h$. En este caso se supone que cuando se activa la tarea *D_Medidor* realiza 2048 medidas de las variables relacionadas con los fallos a detectar, con un periodo de $4 ms$, el cual se supone idóneo para realizar

el análisis por parte de los algoritmos de detección de fallos. Una vez esta tarea finaliza las mediciones, las almacena en la memoria e indica al middleware para que active la tarea *Técnicas_de_diagnóstico*, la cual procesa las muestras y genera un reporte del estado de la planta. Es decir que estos dos componentes se ejecutan en exclusión mutua.

Tabla 6.1. Tiempos de cómputo para el subsistema de diagnóstico

Tarea	WCET (ms), frecuencia baja	WCET (ms), frecuencia alta
<i>D_Medidor</i>	2	0.3
<i>Técnicas_de_diagnóstico</i>	1000	100

Los periodos y plazos de estas tareas son, $T_{D_MEDIDOR} = 4ms$ y $D_{D_MEDIDOR} = 4ms$; La tarea *Técnicas_de_diagnóstico* tiene un periodo de $24h = 86.400.000ms$ y un plazo $D_{DIAGNÓSTICO} = 5000ms$. Este último valor solamente se asigna para realizar los análisis debido a que esta aplicación no tiene restricciones estrictas de tiempo real, aunque se espera que una vez se active finalice en un periodo corto de tiempo.

6.1.2 Consideraciones del NCS

Para este caso de estudio se supone el mismo modelo de sistema y regulador, y respuesta deseada del sistema, presentados en el caso de estudio del apartado 4.2.1. Los parámetros temporales son $T_s = 110ms$ y $D_{CGR} = 80ms$.

El planteamiento de la aplicación de control para este caso de estudio es el siguiente:

- Se tiene un sistema de control para el cual se requieren tres tareas, *C_Medidor*, *C_Controlador* y *C_Actuador*, las cuales se preasignarán cada una en un nodo diferente, *C_Medidor* al nodo 1, *C_Controlador* al nodo 2 y *C_Actuador* al nodo 3.
- Los tiempos de cómputo para las tres tareas se presentan en la tabla 6.2.

Tabla 6.2 Tiempos de cómputo para el subsistema de control

Tarea	WCET, frecuencia baja	WCET, frecuencia alta
<i>C_Medidor</i>	2 ms	0.3 ms
<i>C_Controlador</i>	8 ms	0.8 ms
<i>C_Actuador</i>	1 ms	0.1 ms

6.1.3 Búsqueda del modo de funcionamiento óptimo

Existen tres posibles escenarios en este caso de estudio:

1. Cuando solo está en funcionamiento la aplicación de control.
2. Cuando funcionan la aplicación de control y la tarea *D_Medidor*.

3. Cuando funcionan la aplicación de control y la tarea *Técnicas_de_diagnóstico*.

Para la obtención de los parámetros temporales de cada modo de funcionamiento, requeridos en las funciones de costo, el caso fue simulado empleando la herramienta Truetime. Los pesos asignados a los coeficientes de la función de costo fueron:

- Sistema de cómputo:

$$K_{f_M} = 0.5, K_{\Delta f_{DE}} = 0.2, K_{\Delta U_{DE}} = 0.3.$$

Se da más importancia al valor de los $f_{op_{Mj}}$ (soluciones de frecuencias más bajas). El ΔU_{DEj} tiene mayor peso que $\Delta f_{op_{DEj}}$ por estar más relacionado con la migración de tareas entre los nodos.

- Sistema de control:

$$K_{r_M} = 0.6, K_{\Delta r_{DE}} = 0.4.$$

Se dio más importancia al valor del retraso extremo-extremo (r_M), porque este parámetro afecta más la respuesta dinámica del sistema.

Los modos de funcionamiento para los escenarios 1, 2 y 3, y el valor obtenido de las funciones de costo para cada uno se presentan en las tablas 6.3, 6.4 y 6.5 respectivamente.

Tabla 6.3. Valores de las funciones de costo para el escenario 1

	Nodo	Componentes	Frecuencia de operación	D_{Min}^{func}	D_{Max}^{func}	r_{M}^{func}	$F(Co)_{S_{Computo}}$	$F(Co)_{Control}$
Modo Func. 1	1	C_Medidor,	1	2	No es planificable <i>end - to - end</i>			
	2	C_Controlador	1	8				
	3	C_Actuador	1	1				
Modo Func. 2	1	C_Medidor,	10	0.3		52.1250000	0.6835823082	0.7909374999
	2	C_Controlador	1	8				
	3	C_Actuador	1	1.7				
Modo Func. 3	1	C_Medidor	1	2		60.6750000	0.4428184011	0.8550624999
	2	C_Controlador	10	0.8				
	3	C_Actuador	1	7				
Modo Func. 4	1	C_Medidor	10	0.3		51.9250000	0.5740133033	0.7894375000
	2	C_Controlador	10	0.8				
	3	C_Actuador	1	8.9				
Modo Func. 5	1	C_Medidor	1	2	No es planificable <i>end - to - end</i>			
	2	C_Controlador	1	8				
	3	C_Actuador	10	0.1				
Modo Func. 6	1	C_Medidor	10	0.3		51.2250000	0.8500000000	0.7841874999
	2	C_Controlador	1	8				
	3	C_Actuador	10	1.7				
Modo Func. 7	1	C_Medidor	1	2		59.7750000	0.6139969812	0.8483124999
	2	C_Controlador	10	0.8				
	3	C_Actuador	10	0.1				
Modo Func. 8	1	C_Medidor	10	0.3		51.0250000	0.5240133033	0.7826874999
	2	C_Controlador	10	0.8				
	3	C_Actuador	10	8.9				

Tabla 6.4. Valores de las funciones de costo para el escenario 2

	Nodo	Componentes	Frecuencia de operación	$D_{Min} /ms/$	$D_{Max} /ms/$	$r_M /ms/$	$F(Co)_S_{Computo}$	$F(Co)_{Control}$
Modo Func. 1	1	C_Medidor y D_Medidor	1	4	No es planificable end -to - end			
	2	C_Controlador	1	8				
	3	C_Actuador	1	1				
Modo Func. 2	1	C_Medidor y D_Medidor	10	0.6	1	52.1250000	0.4386745741	0.7909374999
	2	C_Controlador	1	8	8.4	60.6750000	0.6975970512	0.8550624999
	3	C_Actuador	1	1	1.4			
1	C_Medidor y D_Medidor	1	4	8.2				
Modo Func.3	1	C_Controlador	10	0.8	5	51.9250000	0.5905916227	0.7894375000
	2	C_Actuador	1	1	5.2			
	3	C_Medidor y D_Medidor	10	0.6	8.2			
Modo Func. 4	1	C_Controlador	10	0.8	8.4	No es planificable end -to - end		
	2	C_Actuador	1	1	8.6			
	3	C_Medidor y D_Medidor	1	4				
Modo Func. 5	1	C_Medidor y D_Medidor	1	4	No es planificable end -to - end			
	2	C_Controlador	1	8				
	3	C_Actuador	10	0.1				
Modo Func. 6	1	C_Medidor y D_Medidor	10	0.6	1.9	51.2250000	0.5934397554	0.7841874999
	2	C_Controlador	1	8	9.3			
	3	C_Actuador	10	0.1	1.4			
Modo Func. 7	1	C_Medidor y D_Medidor	1	4	9.1	59.7750000	0.8500000000	0.8483124999
	2	C_Controlador	10	0.8	5.9			
	3	C_Actuador	10	0.1	5.2			
Modo Func. 8	1	C_Medidor y D_Medidor	10	0.6	9.1	51.0250000	0.5430885242	0.7826874999
	2	C_Controlador	10	0.8	9.3			
	3	C_Actuador	10	0.1	8.6			

Tabla 6.5. Valores de las funciones de costo para el escenario 3

	Nodo	Componentes	Frecuencia de operación	$D_{Min} (ms)$	$D_{Máx} (ms)$	$r_M (ms)$	$F(Co)_S_{Cómputo}$	$F(Co)_{Control}$
Modo Func. 1	1	<i>C_Medidor,</i> <i>Técnicas de diagnóstico</i>	1	2	No es planificable <i>end-to-end</i>			
	2	<i>C_Controlador</i>	1	8				
	3	<i>C_Actuador</i>	1	1				
Modo Func. 2	1	<i>C_Medidor,</i> <i>Técnicas de diagnóstico</i>	10	0.3	1	52.1250000	0.6835817445	0.7909374999
	2	<i>C_Controlador</i>	1	8	8.7			
	3	<i>C_Actuador</i>	1	1	1.7			
Modo Func. 3	1	<i>C_Medidor,</i> <i>Técnicas de diagnóstico</i>	1	2	8.2	60.6750000	0.4428670973	0.8550624999
	2	<i>C_Controlador</i>	10	0.8	7			
	3	<i>C_Actuador</i>	1	1	7.2			
Modo Func. 4	1	<i>C_Medidor,</i> <i>Técnicas de diagnóstico</i>	10	0.3	8.2	51.9250000	0.5740087882	0.7894375000
	2	<i>C_Controlador</i>	10	0.8	8.7			
	3	<i>C_Actuador</i>	1	1	8.9			
Modo Func. 5	1	<i>C_Medidor,</i> <i>Técnicas de diagnóstico</i>	1	2	No es planificable <i>end-to-end</i>			
	2	<i>C_Controlador</i>	1	8				
	3	<i>C_Actuador</i>	10	0.1				
Modo Func. 6	1	<i>C_Medidor,</i> <i>Técnicas de diagnóstico</i>	10	0.3	1.9	51.2250000	0.8500000000	0.7841874999
	2	<i>C_Controlador</i>	1	8	9.6			
	3	<i>C_Actuador</i>	10	0.1	1.7			
Modo Func. 7	1	<i>C_Medidor,</i> <i>Técnicas de diagnóstico</i>	1	2	9.1	59.7750000	0.6140430781	0.8483124999
	2	<i>C_Controlador</i>	10	0.8	7.9			
	3	<i>C_Actuador</i>	10	0.1	7.2			
Modo Func. 8	1	<i>C_Medidor,</i> <i>Técnicas de diagnóstico</i>	10	0.3	9.1	51.0250000	0.5240123163	0.7826874999
	2	<i>C_Controlador</i>	10	0.8	9.6			
	3	<i>C_Actuador</i>	10	0.1	8.9			

Tabla 6.5. Valores de las funciones de costo para el escenario 3

	Nodo	Componentes	Frecuencia de operación	D_{Mts} / msf	D_{Mts} / msf	r_M / msf	$F(Co)_S_{Computo}$	$F(Co)_{Control}$
Modo Func. 9	1	C_Medidor	1	2	No es planificable <i>end - to - end</i>			
	2	C_Controlador, Técnicas de diagnóstico	1	8				
	3	C_Actuador	1	1				
Modo Fun. 10	1	C_Medidor	10	0.3	1	52.1250000	0.6835848327	0.7909374999
	2	C_Controlador, Técnicas de diagnóstico	1	8	8.7			
	3	C_Actuador	1	1	1.7			
Modo Fun. 11	1	C_Medidor	1	2	8.2	60.6750000	0.4428083399	0.8550624999
	2	C_Controlador, Técnicas de diagnóstico	10	0.8	7			
	3	C_Actuador	1	1	7.2			
Modo Fun. 12	1	C_Medidor	10	0.3	8.2	51.9250000	0.5740105625	0.7894375000
	2	C_Controlador, Técnicas de diagnóstico	10	0.8	8.7			
	3	C_Actuador	1	1	8.9			
Modo Fun. 13	1	C_Medidor	1	2	No es planificable <i>end - to - end</i>			
	2	C_Controlador, Técnicas de diagnóstico	1	8				
	3	C_Actuador	10	0.1				
Modo Fun. 14	1	C_Medidor	10	0.3	1.9	51.2250000	0.8500000000	0.7841874999
	2	C_Controlador, Técnicas de diagnóstico	1	8	9.6			
	3	C_Actuador	10	0.1	1.7			
Modo Fun. 15	1	C_Medidor	1	2	9.1	59.7750000	0.6139858071	0.8483124999
	2	C_Controlador, Técnicas de diagnóstico	10	0.8	7.9			
	3	C_Actuador	10	0.1	7.2			
Modo Fun. 16	1	C_Medidor	10	0.3	9.1	51.0250000	0.5240140891	0.7826874999
	2	C_Controlador, Técnicas de diagnóstico	10	0.8	9.6			
	3	C_Actuador	10	0.1	8.9			

Tabla 6.5. Valores de las funciones de costo para el escenario 3

	Nodo	Componentes	Frecuencia de operación	$D_{Min} [ms]$	$D_{Max} [ms]$	$r_M [ms]$	$F(Co)_S_{Computo}$	$F(Co)_{Control}$
Modo Fun. 17	1	C_Medidor	1	2	No es planificable <i>end - to - end</i>			
	2	C_Controlador	1	8				
	3	C_Actuador, Técnicas de diagnóstico	1	1				
Modo Fun. 18	1	C_Medidor	10	0.3	1	52.1250000	0.6835638039	0.7909374999
	2	C_Controlador	1	8	8.7			
	3	C_Actuador, Técnicas de diagnóstico	1	1	1.7			
Modo Fun. 19	1	C_Medidor	1	2	8.2	60.6750000	0.4428012028	0.8550624999
	2	C_Controlador	10	0.8	7			
	3	C_Actuador, Técnicas de diagnóstico	1	1	7.2			
Modo Fun. 20	1	C_Medidor	10	0.3	8.2	51.9250000	0.5740488069	0.7894375000
	2	C_Controlador	10	0.8	8.7			
	3	C_Actuador, Técnicas de diagnóstico	1	1	8.9			
Modo Fun. 21	1	C_Medidor	1	2	No es planificable <i>end - to - end</i>			
	2	C_Controlador	1	8				
	3	C_Actuador, Técnicas de diagnóstico	10	0.1				
Modo Fun. 22	1	C_Medidor	10	0.3	1.9	51.2250000	0.8500000000	0.7841874999
	2	C_Controlador	1	8	9.6			
	3	C_Actuador, Técnicas de diagnóstico	10	0.1	1.7			
Modo Fun. 23	1	C_Medidor	1	2	9.1	59.7750000	0.6139936994	0.8483124999
	2	C_Controlador	10	0.8	7.9			
	3	C_Actuador, Técnicas de diagnóstico	10	0.1	7.2			
Modo Fun. 24	1	C_Medidor	10	0.3	9.1	51.0250000	0.5240099793	0.7826874999
	2	C_Controlador	10	0.8	9.6			
	3	C_Actuador, Técnicas de diagnóstico	10	0.1	8.9			

Los resultados obtenidos del análisis de las tablas 6.3, 6.4 y 6.5 fueron:

- Para el escenario 1, el modo de funcionamiento 3 presenta el menor valor de $F(Co)_{S_C\acute{o}mputo}$, mientras que el modo de funcionamiento 8 presenta el menor valor de $F(Co)_{S_Control}$.
- Para el escenario 2, el modo de funcionamiento 2 presenta el menor valor de $F(Co)_{S_C\acute{o}mputo}$, mientras que el modo de funcionamiento 8 presenta el menor valor de $F(Co)_{S_Control}$.
- Para el escenario 3, el modo de funcionamiento 19 presenta el menor valor de $F(Co)_{S_C\acute{o}mputo}$, mientras que los modos de funcionamiento 8, 16 y 24 presentan el menor valor de $F(Co)_{S_Control}$.

Con base en lo anterior la optimización de ambos criterios es un problema multiobjetivo. Para analizar la forma de encontrar el modo de funcionamiento óptimo se observó el frente de Pareto, figura 6.2.

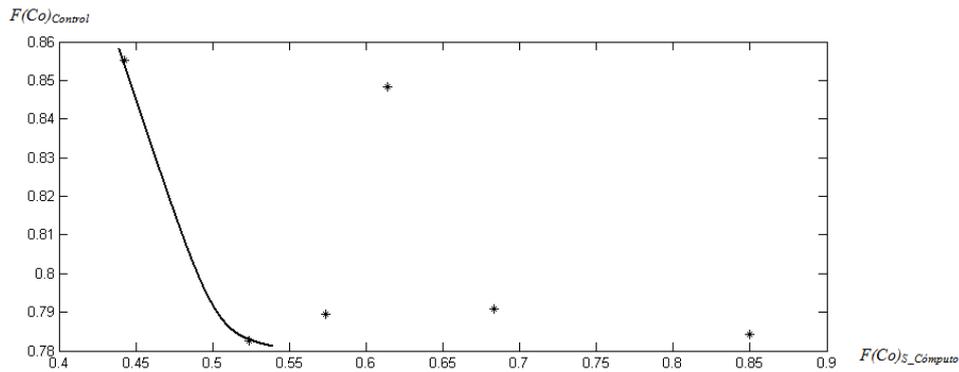


Figura 6.2. Frente de Pareto para el escenario 3.

Debido a que el frente de Pareto es convexo, es posible encontrar la solución óptima al problema multiobjetivo empleando una combinación lineal de pesos. Para lo cual se plantea:

$$F(C_o) = K_{S_C\acute{o}mputo} F(C_o)_{S_C\acute{o}mputo} + K_{Control} F(C_o)_{Control}$$

Donde $K_{C\acute{o}mputo}$ y $K_{Control}$ son coeficientes que indican el peso de cada tipo de parámetros en la calidad de la solución deseada, lo cual depende del tipo de aplicación.

Los valores de los coeficientes utilizados en el caso de estudio 2 fueron:

$$K_{C\acute{o}mputo} = 0.4, \quad K_{Control} = 0.6$$

Los resultados de la optimización son:

- Para el escenario 1, el modo de funcionamiento óptimo es el 8.
- Para el escenario 2, el modo de funcionamiento óptimo es el 2.
- Para el escenario 3, el modo de funcionamiento óptimo es el 24.

6.1.4 Resultados de simulación

En (Varma, 2005) se presenta el siguiente modelo de consumo de potencia para procesadores XScale:

$$P = P_{Estática} + P_{Dinámica} \Rightarrow P = VI_{Fugas} + \frac{1}{2} C_L V_{dd}^2 f$$

El modelo empleado en la simulación, suponiendo el mismo factor de escalado para la frecuencia y el voltaje, es:

- Durante el tiempo de ejecución de cada tarea: $P = 0.01 + 0.02 \left(\frac{f_{funcionamiento}}{f_{baja}} \right)^3$ W.

Donde $f_{funcionamiento}$ es la frecuencia con la que está funcionando el nodo en ese momento.

- Al finalizar el periodo ICI, el procesador se llevará a un estado de bajo consumo de potencia, durante el cual $P = 0.001$ W.

Los resultados de simulación coincidieron con los análisis realizados en el proceso de optimización. En las figuras 6.3 y 6.4 se presenta el consumo de potencia en los nodos, la planificación de tareas en el nodo 3 y la respuesta del sistema de control, para los modos de funcionamiento 16 y 19 del escenario 3.

De las figuras 6.3 y 6.4 se puede apreciar que:

- Las dos configuraciones cumplen con los plazos extremo-extremo propuestos.
- En ambos casos el sistema de control cumple con el desempeño deseado. Aunque se observa una mejor respuesta en el modo de funcionamiento 16 por presentar un retraso mas pequeño en el lazo de realimentación.
- El consumo de potencia en los nodos es mayor en el modo de funcionamiento 16 que en el 19, lo que disminuye el tiempo de vida de la aplicación, ocasionando la inestabilidad del sistema de control antes de 7s de iniciada la simulación, como consecuencia de la terminación de la energía en el nodo 2.

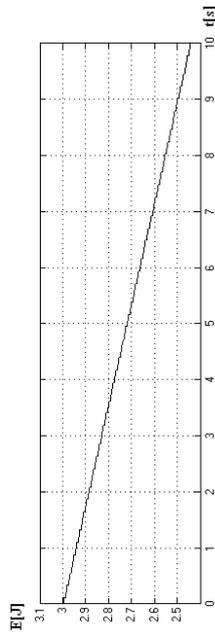


Figura 6.3a. Consumo de potencia en el nodo 1.

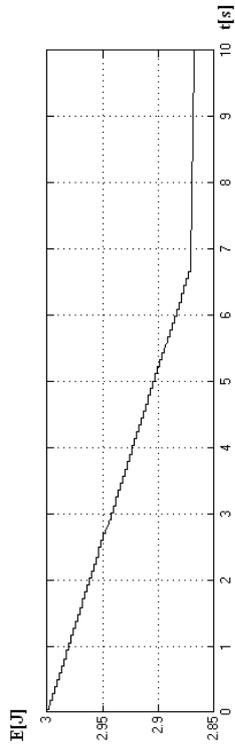


Figura 6.3c. Consumo de potencia en el nodo 3.

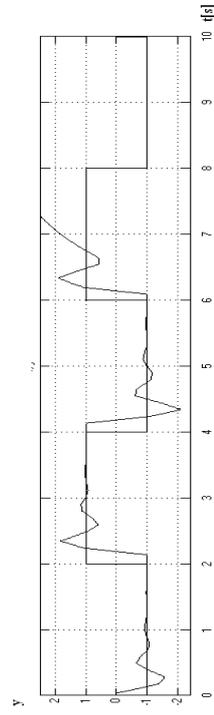


Figura 6.3e. Respuesta del sistema de control. El sistema de control se inestabiliza como consecuencia de la terminación de la energía en el nodo 2.

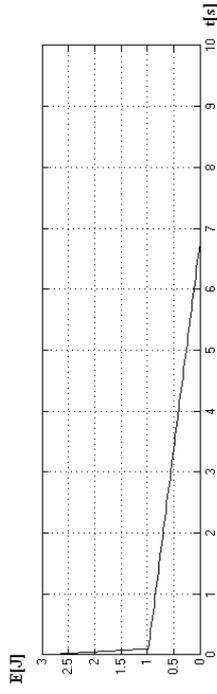


Figura 6.3b. Consumo de potencia en el nodo 2.

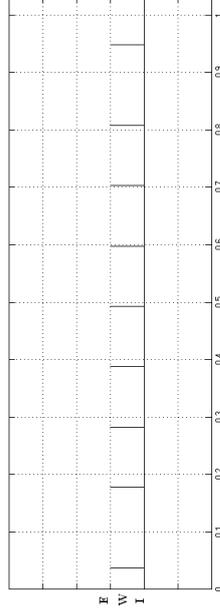


Figura 6.3d. Planificación de tareas en el nodo 3. E, W e I son los estados *Execution*, *Waiting* e *Idle* de la tarea *C_Actuador*.

Figura 6.3. Resultados de simulación del escenario 3, modo de funcionamiento 16.

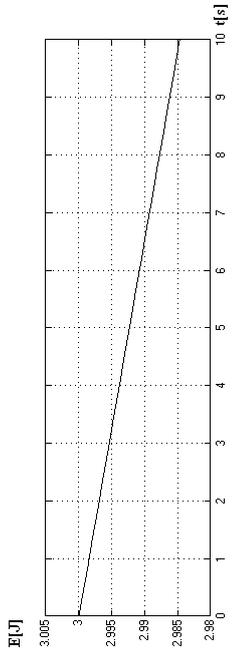


Figura 6.4a. Consumo de potencia en el nodo 1.

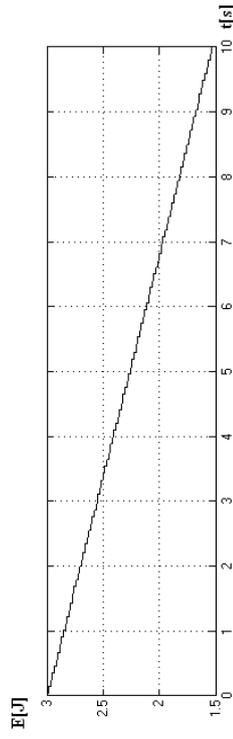


Figura 6.4b. Consumo de potencia en el nodo 2.

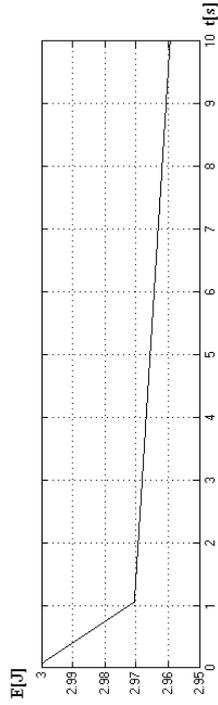


Figura 6.4c. Consumo de potencia en el nodo 3.

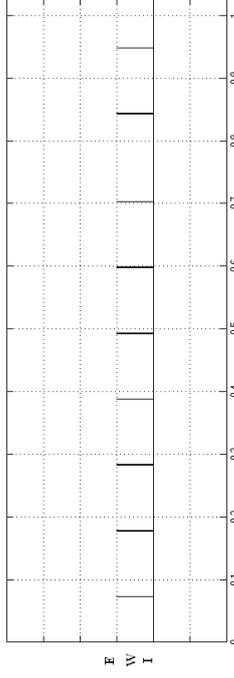


Figura 6.4d. Planificación de tareas en el nodo 3. E, W e I indican los estados *Execution*, *Waiting* e *Idle* de la tarea *C_Actuador*.

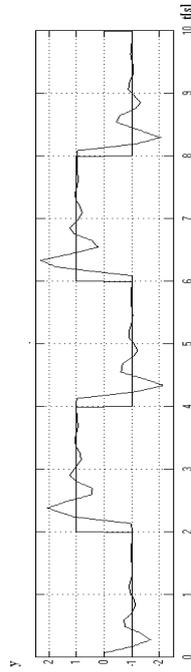


Figura 6.4. Resultados de simulación del escenario 3, modo de funcionamiento 19.

Figura 6.4e. Respuesta del sistema de control. Se aprecia un mayor tiempo de vida de la aplicación, debido a la disminución del consumo de potencia respecto del generado en el modo de funcionamiento 16.

Estos resultados resaltan la importancia de realizar un balance entre los parámetros más relevantes en el diseño de WSA, en este caso los retrasos en el lazo de realimentación y el consumo de potencia, pues se puede apreciar como a pesar de que en las dos configuraciones se cumple con los objetivos de control propuestos, el consumo de potencia es mucho mayor en el modo de funcionamiento 16, lo que generaría un incremento en los costes de implementación del sistema para lograr el mismo tiempo de vida de la aplicación que en el modo 19.

6.1.5 Modelado y validación empleando CPN

En este apartado se analiza el escenario 1 del caso de estudio 2, (Martínez 2011).

- Sistema a controlar $G_{p(z)} = \frac{B}{Z-A}$, con lo cual, $y(k) = Ay(k-1) + Bu(k-1)$.
- Regulador $U_{(z)} = \frac{Acz + Bc}{z-1} \varepsilon_{(z)}$, luego $u(k) = u(k-1) + Ace(k) + Bce(k-1)$

En la figura 6.5 se muestra la representación en CPN del sistema, para el cual se desarrolló un modelo jerárquico. No se ha considerado el uso de planificadores de tareas en los nodos debido a que en cada nodo se ejecutará un único componente. Es posible apreciar la representación en el mismo modelo de la dinámica del sistema y del algoritmo de control, de igual forma se representa el comportamiento de los coprocesadores de cada nodo y la red TDMA.

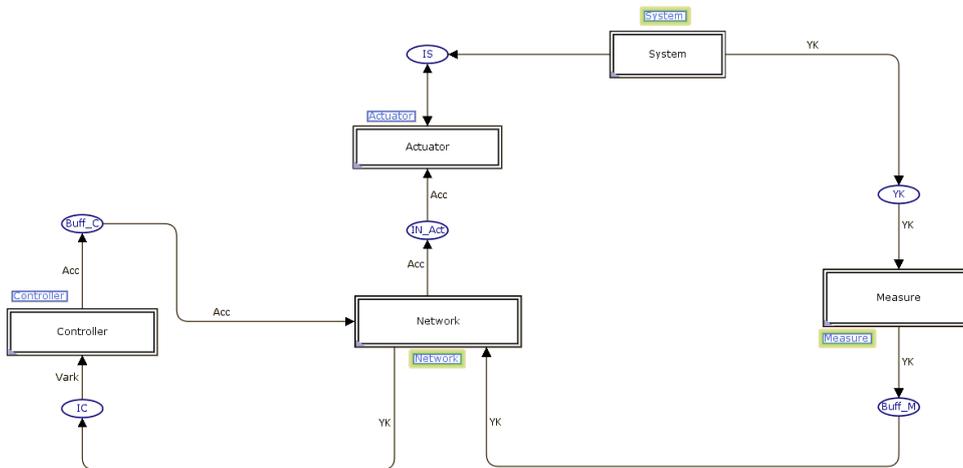


Figura 6.5. Representación en CPN del caso de estudio: WSA con distancia entre nodos acotada a un único salto

Los modelos del sistema a controlar y de la tarea de control, se presentan en las figuras 6.6 y 6.7.

En la figura 6.6, la transición *System* representa la dinámica del sistema, el cual es muestreado cada 110 ms, y el lugar *IS* representa un retenedor de orden cero de la señal de entrada aplicada al sistema.

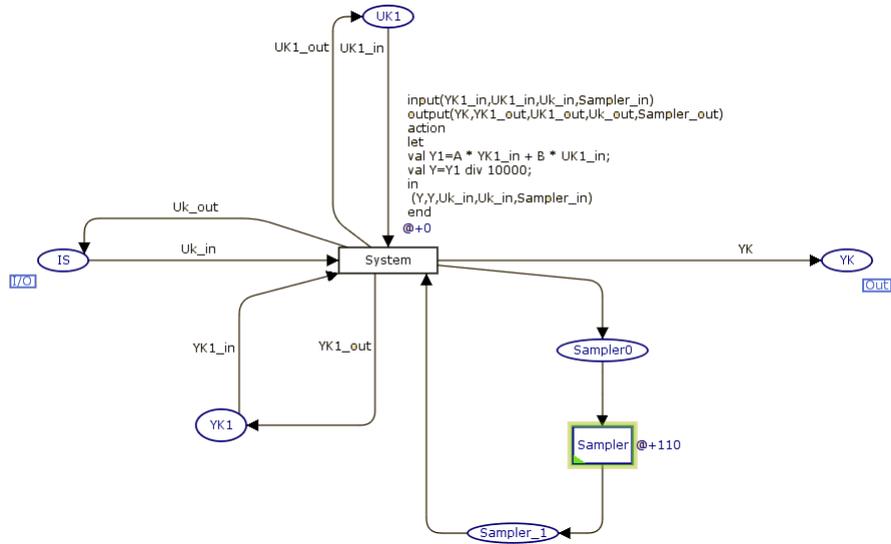


Figura 6.6. Modelo en CPN del sistema a controlar

La transición *Controller* en el modelo de la tarea de control representa la dinámica del regulador.

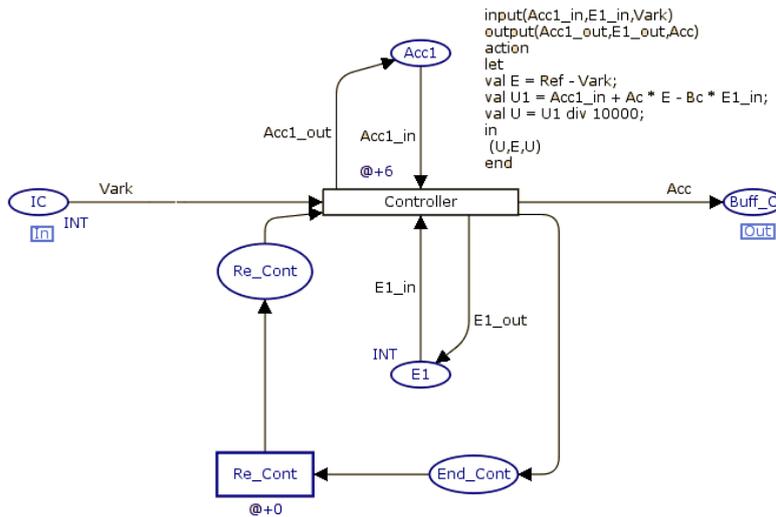


Figura 6.7. Modelo en CPN de la tarea de control

El actuador se modeló como un componente aperiódico, figura 6.8. El lugar *IS*, representa un retenedor de orden cero que se enlaza con el sistema.

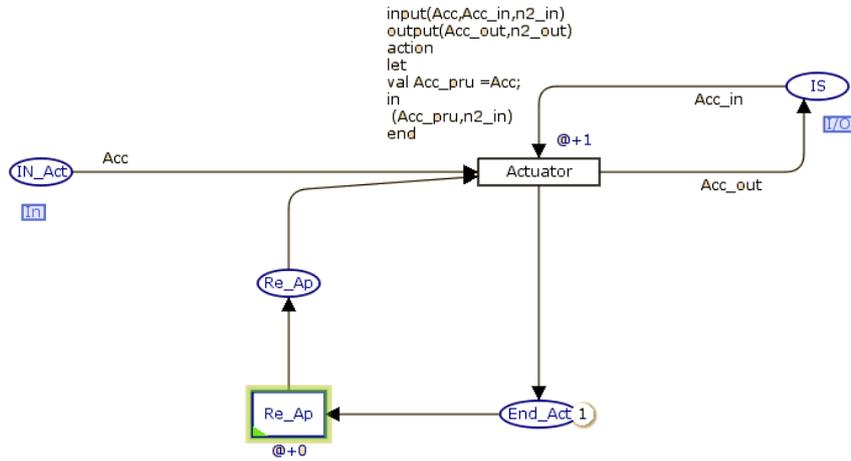


Figura 6.8. Modelo en CPN del actuador

En la figura 6.9, se presenta el modelo de la tarea de medición, la cual se modeló a partir de un componente periódico.

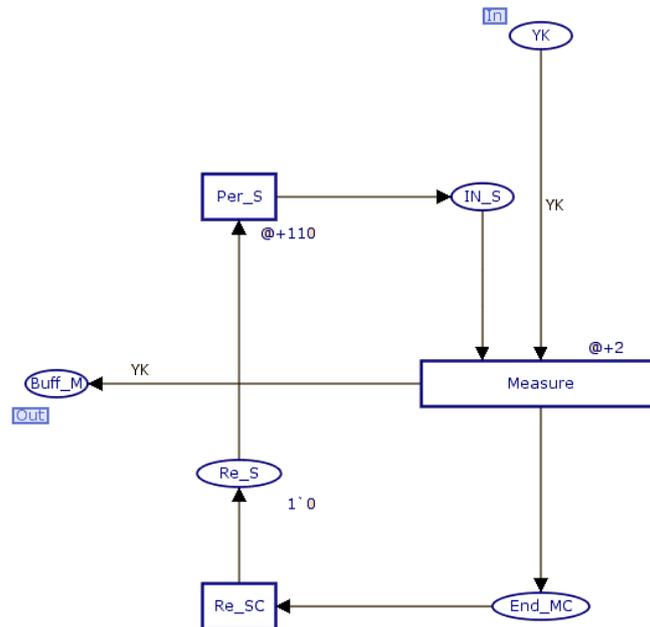


Figura 6.9. Modelo en CPN del medidor

Para representar la red de comunicaciones se utilizó el modelo presentado en la figura 4.15, con dos GTS, GTS1 para el envío de la medición y GTS2 para el envío de la acción de control.

Los resultados del análisis del espacio de estados de la red de comunicaciones y las tareas, sin el modelo del sistema a controlar, utilizando CPN Tools fueron:

- *State Space; Nodes: 300, Arcs: 904, Status: Full.* Lo que indica que se generó el espacio de estados completo, y que está compuesto de 300 nodos (estados) y 904 arcos (ocurrencia de eventos).
- *Best Integer Bounds: Upper = 1.* Condición que presentaron todos los lugares. Lo cual indica que todos los lugares de la red, incluidos los buffer de comunicaciones, no tendrán más de una marca en cualquier estado del sistema.
- *Home Markings: All.* Esto indica que cualquier marcado puede ser alcanzado desde cualquier otro, lo cual es de esperar dado el funcionamiento cíclico del sistema de control, y que además no hay bloqueos en el sistema.
- *Dead Markings: None.* Quiere decir que no hay marcados donde ninguna transición esté habilitada, lo cual concuerda con el resultado anterior.
- *Dead Transition Instances: None.* Esto indica que cada transición tiene la posibilidad de activarse por lo menos una vez.
- *Live Transition Instances: All.* Quiere decir que todas las transiciones son alcanzables desde cualquier marcado del sistema. Lo cual nuevamente coincide con el comportamiento cíclico del sistema.

6.2 Caso de estudio: WSAN con comunicación multisalto

En este caso se considera una red formada por cinco nodos, sobre los cuales se implementan dos NCS, figura 6.10, (Martínez, 2009c). Se puede apreciar los distintos caminos que se pueden utilizar para el envío de la información entre los diferentes nodos.

6.2.1 Consideraciones generales

Para este caso de estudio en cada lazo de control se supone el mismo modelo de sistema y regulador presentados en el apartado 4.2.1. En la WSAN existen diferentes caminos para interconectar los nodos, y para cada lazo de control siempre es necesario realizar entre dos y tres saltos en total para transmitir los mensajes entre las tareas *Medidor*, *Controlador* y *Actuador*, así que dependiendo de las rutas de encaminamiento utilizadas en cada lazo de control, pueden llegarse a almacenar mensajes en nodos enrutadores.

control, estos se presentan en la tabla 6.6.

Tabla 6.6. Tiempos de cómputo de las tareas para cada uno de los lazos de control

Tarea	WCET (ms), frecuencia baja	WCET (ms), frecuencia alta
Medidor x	2	0.3
Controlador x	8	0.8
Actuador x	1	0.1

6.2.2 Búsqueda del modo de funcionamiento óptimo

En este caso de estudio, para establecer la calidad de un modo de funcionamiento con relación al sistema de cómputo se han considerado los siguientes índices:

- foP_{M_j} , es la media entre las frecuencias de operación de los nodos que conforman la WSAW para una configuración j . Se emplea para encontrar el modo de funcionamiento donde los nodos operen a la menor frecuencia posible.
- ΔfoP_{DEj} , es la desviación estándar de las frecuencias de operación de los nodos. Se utiliza para buscar un consumo de potencia equilibrado entre los nodos.
- $\Delta U_p E_{DEj}$, es la desviación estándar del índice que relaciona el porcentaje de utilización del procesador principal y la energía de cada nodo. Se utiliza para buscar un tiempo de vida equitativo entre los nodos. Donde el índice que relaciona el porcentaje de utilización del procesador principal y la energía del nodo j en el que se ejecutan i tareas es:

$$U_p E = \left(\sum_{i=1}^n \frac{WCET_i}{T_i} \right) / E_j, \quad E_j \text{ es la energía en el nodo } j.$$

- $\Delta U_c E_{DEj}$, es la desviación estándar de los índices que relacionan el porcentaje de utilización del co-procesador y la energía de cada nodo. Se utiliza para buscar un tiempo de vida equitativo entre los nodos. Donde el índice que relaciona el porcentaje de utilización del coprocesador y la energía de un nodo j a través del cual circulan i mensajes es:

$$U_N E = \left(\sum_{i=1}^n \frac{slot}{T_{S_i}} \right) / E_j$$

Con base en la comparación de la potencia consumida en los diferentes modos de operación de los transceiver RFM TR1000, Chipcon CC1000 y Chipcon CC2420,

presentado en (Li, 2008), se considera el mismo consumo de potencia para la recepción y la transmisión del mensaje.

Respecto del desempeño del sistema de control se consideraron los mismos parámetros definidos en la función de costo del caso de estudio del apartado 6.1, r_M , y Δr_{DE} .

Para la obtención de los parámetros temporales de cada configuración de operación, requeridos en las funciones de costo, este caso de estudio fue simulado empleando la herramienta Truetime. Los pesos asignados a los parámetros fueron:

- Sistema de cómputo:

$$K_{fop_{Mj}} = 0.4, K_{\Delta fop_{DEj}} = 0.2, K_{\Delta U_{PEDEj}} = 0.2, K_{\Delta U_{CEDEj}} = 0.2.$$

Se da más importancia al valor de los fop_{Mj} (soluciones de frecuencias más bajas).

- Sistema de control:

$$K_{r_{M1,2}} = 0.35, K_{\Delta r_{DE1,2}} = 0.15.$$

Se dio más importancia al valor del retraso extremo-extremo (r_M) porque este parámetro afecta más la respuesta dinámica del sistema.

El número total de configuraciones fue 864, de las cuales 272 cumplen con los plazos extremo-extremo. La optimización de ambos criterios es un problema multiobjetivo, para encontrar la solución óptima considerando ambos criterios se empleó una combinación lineal de pesos, para lo cual se planteó la misma función del apartado 6.1.3:

$$F(C_o) = K_{S_Cómputo} F(C_o)_{S_Cómputo} + K_{Control} F(C_o)_{Control}$$

Los valores de los coeficientes utilizados fueron:

$$K_{Cómputo} = 0.4, K_{Control} = 0.6$$

La configuración óptima es:

- Todos los nodos, excepto el nodo 1, operan a alta frecuencia.
- *Controlador_1* en el nodo 3.
- *Controlador_2* en el nodo 4.
- Las rutas de los mensajes para cada lazo de control son:
 - Lazo 1: 1 – 3 – 4. Lazo 2: 2 – 4 – 5.

6.2.3 Resultados de simulación

Se empleó el mismo modelo de consumo de potencia del procesador presentado en el apartado 6.1.4. Los resultados de simulación coincidieron con los análisis realizados. En la figura 6.12 se compara el consumo de potencia en los nodos que ejecutan las tareas *Controlador*, cuando se utiliza la configuración óptima y cuando se utiliza la siguiente configuración sub-óptima:

- Nodos 2 y 3 operan a alta frecuencia y los demás a baja frecuencia.
- *Controlador_1* en nodo 3 y *Controlador_2* en el nodo 2.
- Las rutas de los mensajes para cada lazo de control son:
 - Lazo 1: 1 – 3 – 4. Lazo 2: 2 – 4 – 5.

Se observa como el tiempo de vida de la aplicación es mayor al utilizar la configuración óptima. Además, los retrasos presentes en cada lazo de control fueron $\tau_{r_1} = 11.1ms$ y $\tau_{r_2} = 4.1ms$ para la configuración óptima, y $\tau_{r_1} = 12ms$ y $\tau_{r_2} = 5ms$ para la sub-óptima.

Como una estrategia adicional para disminuir el consumo de potencia en el sistema, se analizó el desempeño de los lazos de control considerando el lazo 2 basado en tiempo, y el lazo 1 basado en eventos siguiendo la propuesta presentada en (Årzén, 1999).

Para el lazo de control basado en tiempo se considera $T_s = 40ms$. Mientras que en el lazo 2 los eventos son:

$$|e_k - e_s| \geq 0.1 \quad \text{o} \quad |t_k - t_s| \geq 120ms$$

- e_k y e_s representan el error actual y el error cuando se generó el último evento respectivamente.
- t_k y t_s representan el instante de muestreo actual y el instante de muestreo en que se presentó el último evento respectivamente. La máxima diferencia entre estos valores depende del periodo de muestreo máximo con el que se ha de muestrear el sistema, lo cual está relacionado con el ancho de banda del mismo.
- El periodo mínimo con el que se muestrea la señal es $40ms$.

De esta forma el lazo de control por eventos realiza acciones cada $40ms$ durante los transitorios, y retrasa el periodo hasta un máximo de $120ms$ en los periodos estacionarios, logrando una disminución significativa en la utilización de los procesadores y de la red. En la figura 6.13 se puede apreciar que existe un aumento del intervalo de tiempo de generación de eventos en el lazo de control 2 durante los estados estacionarios del respectivo sistema.

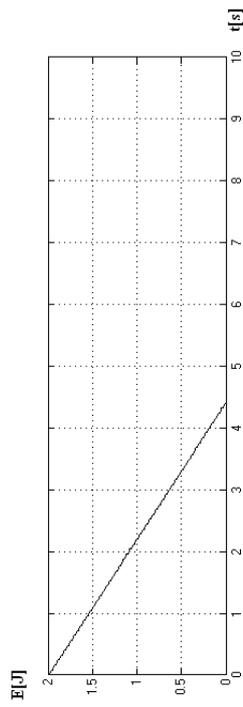


Figura 6.12a. Energía en el nodo 4. Configuración óptima.

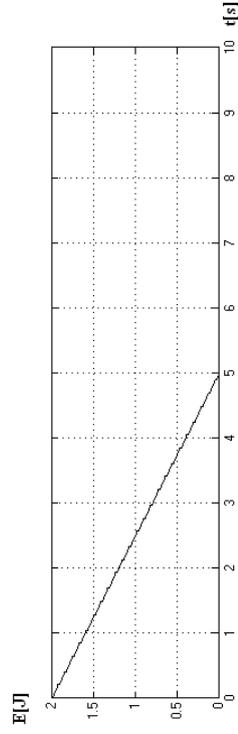


Figura 6.12b. Energía en el nodo 3. Configuración óptima.

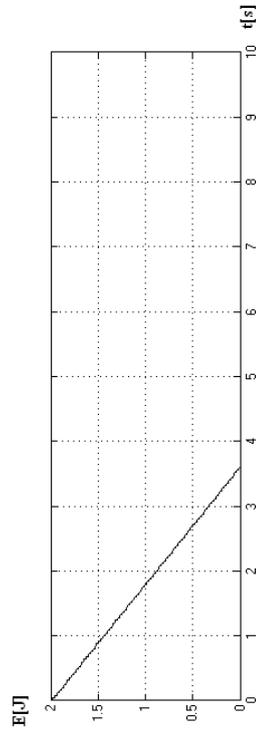


Figura 6.12c. Energía en el nodo 2. Configuración sub-óptima.

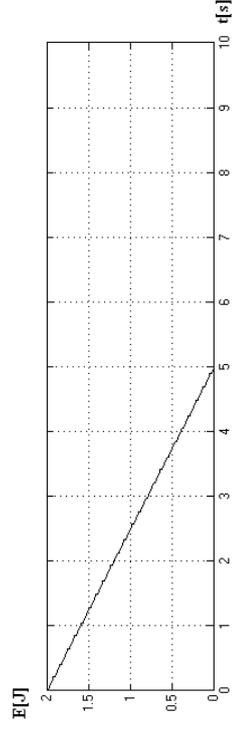


Figura 6.12d. Energía en el Nodo 4. Configuración sub-óptima.

Figura 6.12. Comparación del consumo de potencia en los nodos que ejecutan las tareas *Controlador*, para la configuración óptima y una sub-óptima

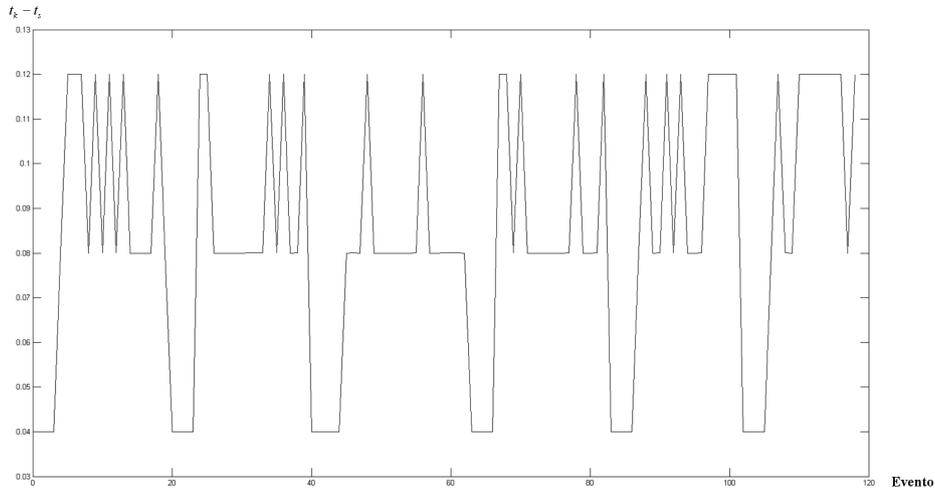


Figura 6.13. Periodos entre la generación de eventos en el lazo de control 2.

Las tareas *Controlador* son las que requieren un mayor cómputo y por lo tanto demandan un mayor consumo de potencia. En la figura 6.14 se observa que la energía en el nodo 4, el cual ejecuta las tareas *Controlador_2* y *Actuador_1*, tiene una mayor duración que la energía en el nodo 3, el cual ejecuta la tarea *Controlador_1*. Con lo que se aprecia el aumento en el tiempo de vida del lazo de control 2, como consecuencia de la disminución significativa en la utilización de los procesadores y de la red de la estrategia de control por eventos.

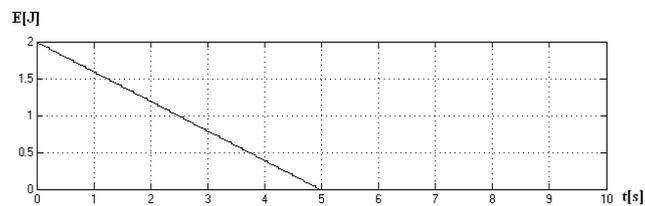


Figura 6.14a. Energía en el nodo 3.

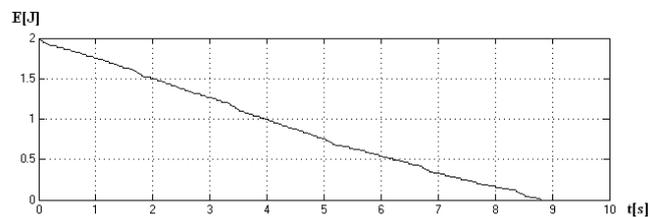


Figura 6.14b. Energía en el nodo 4.

Figura 6.14. Consumo de potencia en los nodos que ejecutan las tareas *Controlador*

Las respuestas de los lazos de control se presentan en la figura 6.15, se observa que ambos cumplen con las especificaciones propuestas. Existe un leve sobre paso en la respuesta del lazo de control 2 con relación al lazo de control 1, esto como consecuencia de su implementación por eventos.

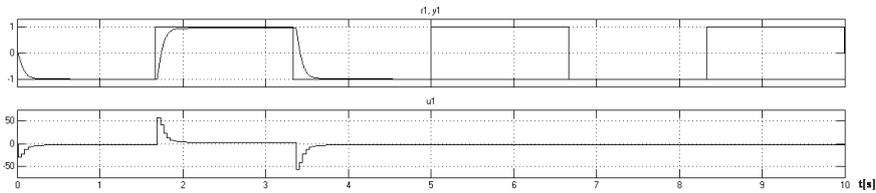


Figura 6.15a. Respuesta del lazo de control 1.

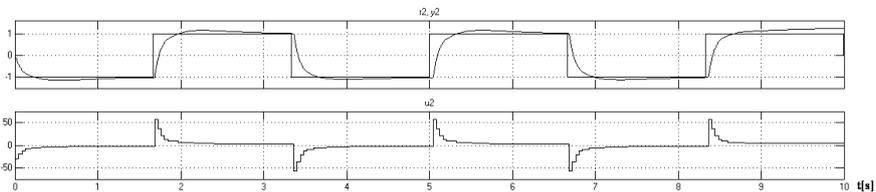


Figura 6.15b. Respuesta del lazo de control 2.

Figura 6.15. Resultados de simulación. Lazo de control 1 periódico y lazo de control 2 por eventos.

6.3 Conclusiones

Los resultados obtenidos corroboran la validez de la metodología de diseño propuesta, y resaltan la importancia de realizar un balance entre los retrasos en el lazo de realimentación y el consumo de potencia en el diseño de WSAN. La metodología propuesta también permite garantizar el cumplimiento de las restricciones de tiempo real impuestas por los algoritmos de control, y de esta manera contribuye a disminuir la brecha entre los resultados reales, y los análisis teóricos y de simulación.

Los modelos de componentes en CPN permiten comprobar las propiedades de comportamiento y estructurales del sistema, y realizar análisis dinámicos del sistema completo considerando el desempeño concurrente de todas sus funciones.

Capítulo 7

VERIFICACIÓN EXPERIMENTAL

Con el propósito de analizar el desempeño de WSN teniendo en cuenta las tecnologías seleccionadas y las consideraciones de diseño realizadas en esta tesis, se implementó una aplicación en la que se regula la variable de salida de un sistema de orden 2, para lo cual se emplearon dos nodos, uno de los nodos realizó las funciones de *Medición y Actuación*, mientras que en el otro se implementó la función del *Controlador*. La arquitectura del sistema se presenta en la figura 7.1 y se describe a continuación:

- Para comunicar los nodos se utilizó el modo síncrono del protocolo IEEE 802.15.4. Esto se realizó a través de dos sistemas de desarrollo SmartRF[®]04DK, cada uno de los cuales tiene incorporado un procesador CC2430. El periodo de la supertrama fue de 15.36 ms. Durante cada intervalo se transmite una trama de datos y una de confirmación.
- El nodo que implementa la función del *Controlador* también cumple con la función de coordinador de red IEEE 802.15.4, e implementa un acceso sincronizado a través de *beacons*. Estas dos funcionalidades se implementaron sobre el mismo sistema de desarrollo SmartRF[®]04DK.
- El nodo en el que se implementaron las tareas de *Medición y Actuación* actúa como dispositivo final en la red IEEE 802.15.4. Este nodo se implementó por medio de un sistema de desarrollo SmartRF[®]04DK y un sistema de desarrollo Colibri XScale[®] PXA320. En el procesador XScale se ejecutan las tareas *Medición y Actuación*. La primera realiza la medición y envía la orden al CC2430 para que transmita este dato al nodo que implementa la función del *Controlador*, y la tarea *Actuación* recibe la acción de control y actúa sobre el sistema.

Las tareas *Medición y Actuación* se implementaron sobre un kernel Linux. La tarea *Medición* se ejecuta de forma periódica, mientras que la tarea *Actuación* se ejecuta cuando recibe la indicación de que ha llegado la trama que contiene la acción de control. De igual forma la tarea *Controlador* se ejecuta cuando se detecta el evento de llegada de la medida de la variable controlada.

El esquema de ejecución de las tareas se presenta en la figura 7.2.

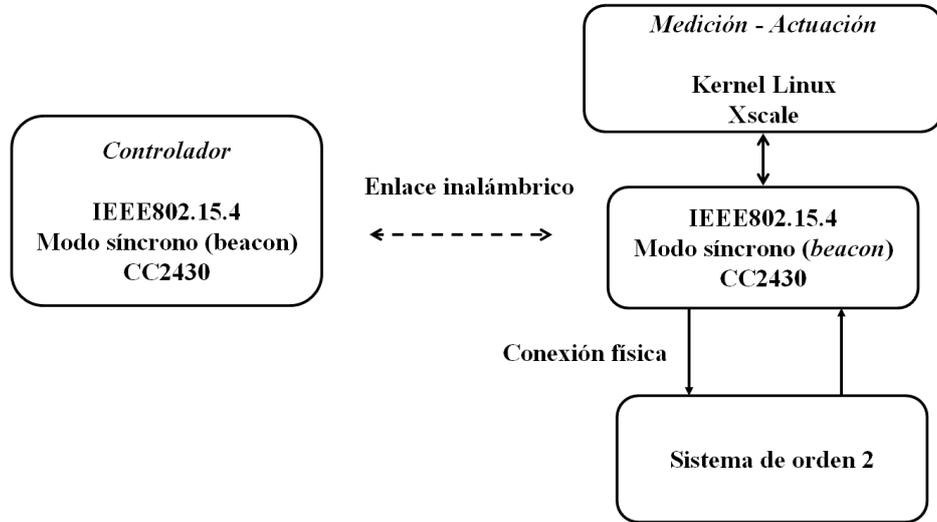


Figura 7.1. Arquitectura del NCS implementado

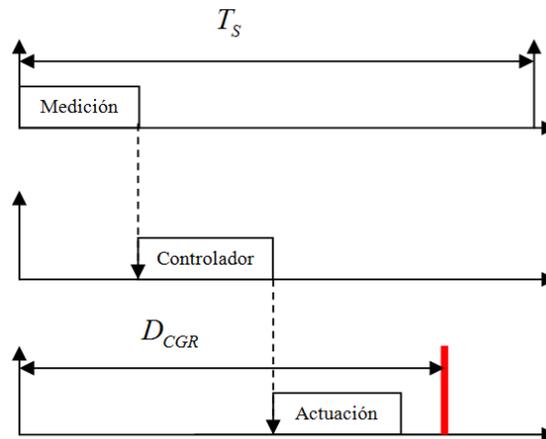


Figura 7.2. Esquema de ejecución de las tareas del NCS implementado

Como se puede apreciar, los nodos se comunican a través de una WSN con distancia entre nodos acotada a un único salto, por lo que es posible emplear el test de planificabilidad extremo-extremo presentado en el apartado 4.2. Sin embargo, debido a que las tareas se ejecutan en exclusión, y a que el retraso total en el lazo de control es inferior a un periodo de muestreo, las tareas *Medición* y *Actuación* nunca están en contienda por el acceso al procesador. De igual forma, al estar los nodos sincronizados tampoco estarán en contienda por el acceso a la red.

El establecimiento de la red entre el coordinador y el dispositivo final se presenta a continuación.

Inicialmente el coordinador envía tramas *beacons* periódicamente. Los campos PC y AP de la trama indican que esta fue transmitida por un coordinador (PC = 1) y que el coordinador está aceptando la vinculación de nuevos nodos (AP = 1).

T	dT	Len	Type	Encr	Pnd	Ack	Seq	SrcPAN	SrcAdd	BO	SO	CAP	BLE	PC	AP	GTSC	GTSP	SPI	LP	RSSI	LQI	FCS
20:15:23.0.250		16	Bcn	No	No	No	F7	11CC	AABB	04	02	0F	No	Yes	Yes	00	No	00	00	-44	98%	OK
20:15:23.0.250		16	Bcn	No	No	No	F8	11CC	AABB	04	02	0F	No	Yes	Yes	00	No	00	00	-44	97%	OK
20:15:24.0.250		16	Bcn	No	No	No	F9	11CC	AABB	04	02	0F	No	Yes	Yes	00	No	00	00	-42	98%	OK
20:15:24.0.235		16	Bcn	No	No	No	FA	11CC	AABB	04	02	0F	No	Yes	Yes	00	No	00	00	-41	98%	OK

Posteriormente el dispositivo envía una solicitud de vinculación al coordinador.

T	dT	Len	Type	Encr	Pnd	Ack	Seq	DstPAN	DstAdd	SrcPAN	SrcAdd	Association	AP	Type	Pover	Idle	Rx	Sec	Shrt	RSSI	LQI	FCS
20:15:24.0.031		21	Cmd	No	No	Yes	2D	11CC	AABB	FFFF	0703F0E0D0C0B0A0	Request	No	RFD	Battery	On	No	Yes	-46	97%	OK	

El coordinador responde enviando una trama de confirmación de la recepción de la solicitud de vinculación.

T	dT	Len	Type	Encr	Pnd	Ack	Seq	RSSI	LQI	FCS
20:15:24.0.031		5	Ack	No	No	No	2D	-42	98%	OK

El coordinador continúa enviando tramas *beacons* periódicamente

T	dT	Len	Type	Encr	Pnd	Ack	Seq	SrcPAN	SrcAdd	BO	SO	CAP	BLE	PC	AP	GTSC	GTSP	SPI	LP	PAI	RSSI	LQI	FCS
20:15:24.0.203		24	Bcn	No	No	No	FB	11CC	AABB	04	02	0F	No	Yes	Yes	00	No	00	01	0703F0E0D0C0B0A0	-43	98%	OK

Luego el dispositivo envía esta trama para solicitar una dirección dentro de la red.

T	dT	Len	Type	Encr	Pnd	Ack	Seq	DstPAN	DstAdd	SrcPAN	SrcAdd	Data	RSSI	LQI	FCS
20:15:24.0.032		18	Cmd	No	No	Yes	2E	11CC	AABB	Same	0703F0E0D0C0B0A0	Request	-45	98%	OK

El coordinador confirma la recepción de la nueva solicitud.

T	dT	Len	Type	Encr	Pnd	Ack	Seq	RSSI	LQI	FCS
20:15:24	0.031	5	Ack	No	Yes	No	2E	-43	97%	OK

Posteriormente el coordinador envía la dirección que le ha asignado al dispositivo.

T	dT	Len	Type	Encr	Pnd	Ack	Seq	DstAdd	SrcPAN	SrcAdd	Association Response	Addr	Status	RSSI	LQI	FCS
20:15:24	0.015	27	Cnd	No	No	Yes	04	11CC 0703F0E0D0C0B0A0	Same	8877665544332211	0001	0001	OK	-43	96%	OK

Finalmente el dispositivo envía la confirmación de la recepción de la dirección asignada.

T	dT	Len	Type	Encr	Pnd	Ack	Seq	RSSI	LQI	FCS
20:15:24	0.016	5	Ack	No	No	No	04	-44	97%	OK

En el NCS considerado se regula la variable de salida en un sistema de orden 2 implementado con amplificadores operacionales, el modelo obtenido del sistema fue:

$$G_{p(s)} = \frac{14.5}{s^2 + 5.128s + 21.08}$$

La respuesta simulada del sistema de orden 2 a una entrada escalón unitaria se presenta en la figura 7.3.

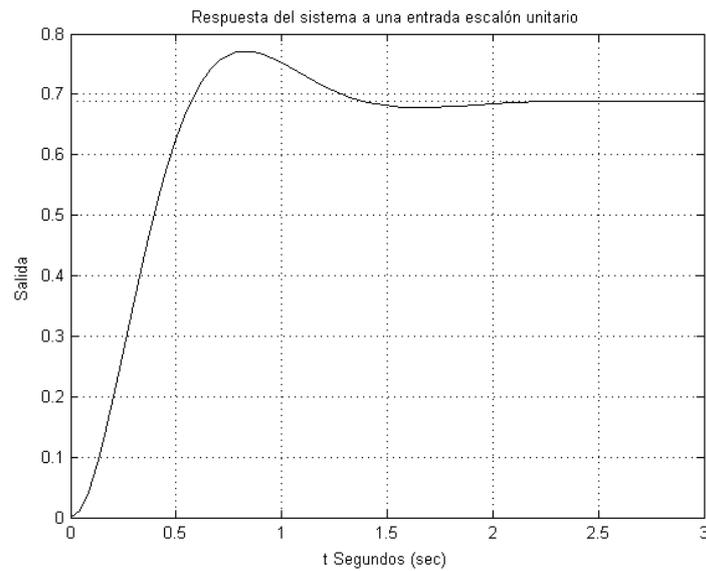


Figura 7.3. Respuesta simulada del sistema de orden 2 a una entrada escalón unitario

Los parámetros de la respuesta del sistema a una entrada escalón son:

- *RiseTime*: 0.3847
- *SettlingTime*: 1.2750
- *SettlingMin*: 0.6250
- *SettlingMax*: 0.7711
- *Overshoot*: 12.0602
- *Undershoot*: 0
- *Peak*: 0.7711
- *PeakTime*: 0.8265

Para regular la variable de salida del sistema se ajustó un regulador PID cuya estructura y parámetros son:

$$U_{(z)} = \frac{q_0 z^2 + q_1 z + q_2}{z^2 - z} \varepsilon_{(z)}$$

Donde:

$$q_0 = K_p + \frac{K_d}{T_s}, \quad q_1 = -K_p - \frac{2K_d}{T_s} + K_i T_s, \quad q_2 = \frac{K_d}{T_s}, \quad K_p = 2, \quad K_i = 4, \quad K_d = 1$$

Como se presentó en el apartado 4.2.1, los retrasos en este tipo de configuraciones no son fijos, debido a que los procesadores que ejecutan las tareas y que se encargan de la comunicación no se encuentran sincronizados, sin embargo este retraso está acotado, y en este caso el retraso máximo en el lazo de control es de dos veces el periodo de la supertrama.

Los resultados experimentales y de simulación en Truetime del NCS, con un periodo de muestreo de 90 ms, y con retrasos máximos en el lazo de control de 30 ms y 80 ms se presentan en la figura 7.4. Se aprecia el efecto en la dinámica del NCS cuando se realiza la implementación considerando un retraso de 80 ms en el lazo de control.

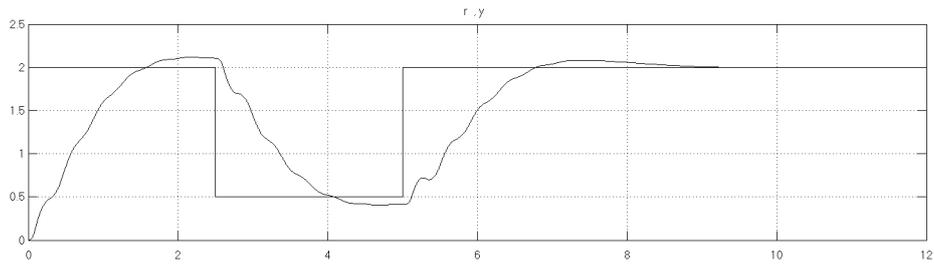


Figura 7.4a. Resultados de simulación, periodo 90 ms y retraso 30 ms

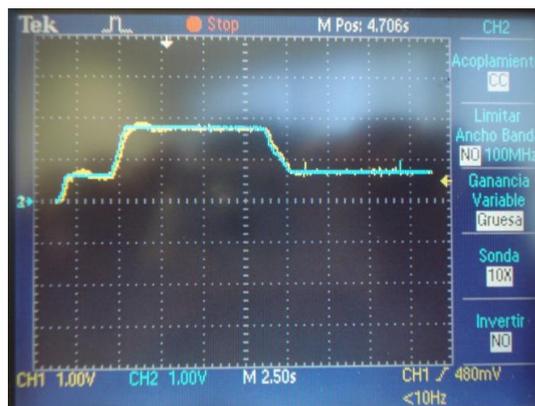


Figura 7.4b. Resultados experimentales, periodo 90 ms y retraso 30 ms

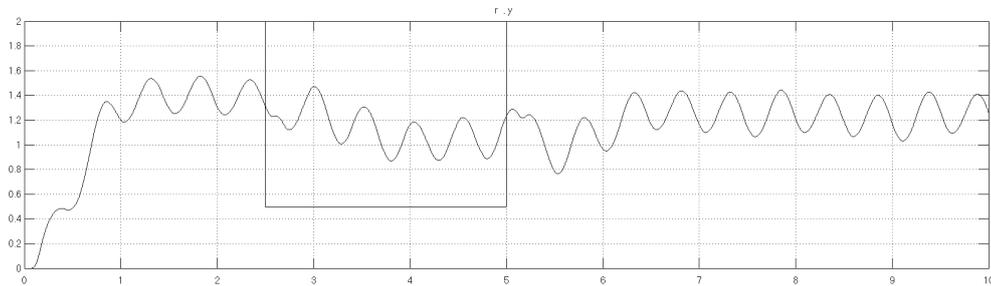


Figura 7.4c. Resultados de simulación, periodo 90 ms y retraso 80 ms

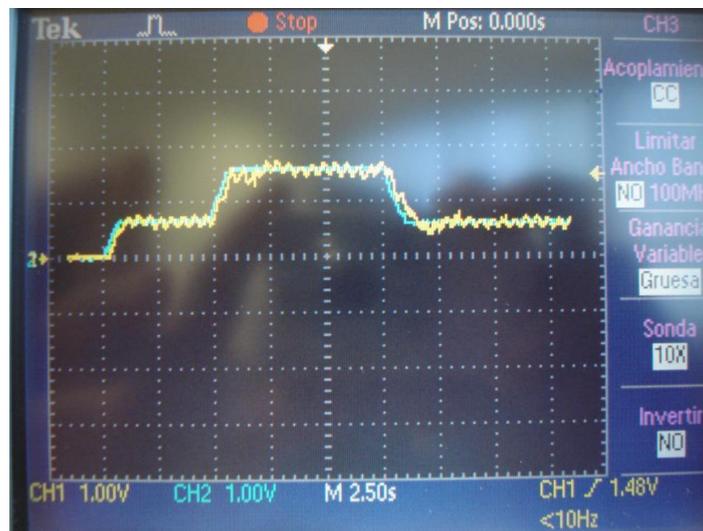


Figura 7.4d. Resultados experimentales, periodo 90 ms y retraso 80 ms

Figura 7.4. Resultados experimentales y de simulación del NCS

En la última prueba realizada sobre este caso se observó el desempeño del algoritmo PID basado en eventos siguiendo la propuesta presentada en (Årzén, 1999), figura 7.5. Los parámetros de configuración del controlador fueron:

- Periodo de muestreo 90 ms.
- Eventos: $|e_k - e_s| \geq 0.02v$ o $|t_k - t_s| \geq 180ms$
- Retraso máximo en la red de comunicaciones de 30 ms.

El tren de pulsos que aparece en la parte superior de la imagen presentada en la figura 7.5 representan los instantes en que se ejecutan las funciones de *Medición*, *Controlador*

y *Actuación*. Cada pulso se pone en nivel alto en el inicio de la *Medición* y se regresa al nivel bajo al completarse la *Actuación*.

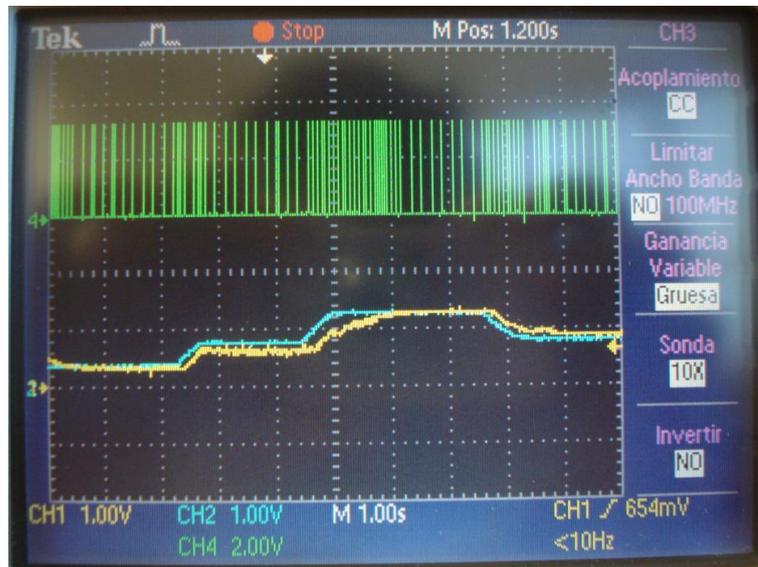


Figura 7.5. Resultados experimentales, PID por eventos

Se puede apreciar una disminución del uso de los recursos (la red y los procesadores) durante los estados estacionarios del sistema, y una disminución en la calidad del control con relación a la implementación periódica presentada en la figura 7.4b.

Los resultados experimentales permiten observar como incluso retrasos pequeños, del orden de decenas de milisegundos, los cuales se pueden presentar utilizando la tecnología actual, pueden llegar a degradar considerablemente la respuesta de NCS implementados sobre WSN en los que las constantes de tiempo sean de orden similar a estos retrasos, lo cual es común en un amplio espectro de sistemas dinámicos reales. De igual forma se aprecia que al utilizar las tecnologías seleccionadas y las consideraciones de diseño realizadas en esta tesis, es posible implementar WSN cuyos resultados experimentales concuerdan con los objetivos de diseño propuestos.

Capítulo 8

CONCLUSIONES Y LÍNEAS FUTURAS DE INVESTIGACIÓN

8.1 CONCLUSIONES

Las problemáticas presentes actualmente en el desarrollo de aplicaciones de monitorización, diagnóstico y control para los sectores médico, medioambiental, agrícola e industrial, entre otros, han motivado el desarrollo de las WSN.

El desarrollo de estas aplicaciones integra disciplinas que han tenido desarrollos aislados, por lo que cada una emplea métodos de diseño y análisis con formalismos de modelado y herramientas distintas, en donde cada representación pone de manifiesto ciertas características sin contemplar los demás subsistemas. Es decir, que los distintos componentes del sistema se diseñan y analizan en herramientas diferentes, limitando el análisis de la interacción entre los mismos y el uso de enfoques metodológicos de diseño concurrente que permitan optimizar algunos de los parámetros críticos en estas aplicaciones. Lo que plantea desafíos en el desarrollo de *frameworks*, algoritmos, metodologías de diseño sistemático y herramientas para satisfacer los requisitos de estas aplicaciones.

El desarrollo de estos sistemas está fuertemente determinado por las restricciones impuestas por la aplicación. Los trabajos realizados hasta el momento se han orientado principalmente al planteamiento de arquitecturas de nodos y protocolos de comunicaciones. Sin embargo aún continúan abiertos diversos problemas relacionados principalmente con el soporte de los niveles de calidad de servicio de estas aplicaciones, la protección de la información y la optimización de los parámetros de mayor relevancia en las aplicaciones.

La principal contribución de esta tesis consiste en el planteamiento de una metodología de diseño para aplicaciones de redes inalámbricas de sensores y actuadores con retrasos acotados. Como caso particular se analizó la implementación de NCS. La metodología se soporta en una arquitectura propuesta para los nodos en la que se consideran los requisitos de las aplicaciones, permite verificar el cumplimiento de la planificabilidad extremo-extremo, y encuentra el modo de funcionamiento óptimo para cada uno de los escenarios de la aplicación, que realiza un balance entre el mínimo consumo de potencia y los retrasos mínimos en la generación de acciones en los sistemas; para lo cual se soporta en la coordinación de las actividades de los diferentes niveles de la arquitectura de los nodos y emplea funciones de costo cuyos parámetros dependen de las características y restricciones de cada aplicación. Adicionalmente se presentan modelos de componentes en CPN que permiten la verificación de propiedades de comportamiento y estructurales del sistema, y realizar análisis dinámicos del sistema completo.

Para obtener la anterior contribución se ha realizado un análisis y clasificación de estas aplicaciones para los sectores médico, medioambiental, agrícola e industrial. Igualmente se analizaron las arquitecturas de los nodos utilizados en algunas aplicaciones, y las metodologías y tecnologías utilizadas para realizar este tipo de desarrollos. También se evaluó el desempeño de aplicaciones de NCS implementadas sobre WSAN, tanto en simulación como experimentalmente, y se analizaron métodos de análisis de planificabilidad para sistemas distribuidos.

Los resultados obtenidos en los casos de estudio abordados permiten concluir que:

- La arquitectura propuesta para los nodos considera las restricciones propias de los sectores de aplicación considerados, lo que permite encontrar soluciones óptimas a los desafíos que se presentan en el diseño de estas aplicaciones, y facilita su desarrollo y validación. Además posibilita la coordinación de las actividades de los niveles de la arquitectura del sistema, para elegir entre diferentes modos de operación en función de los índices de calidad de la aplicación.
- La integración de los componentes de la arquitectura de nodos propuesta, junto con los métodos de análisis de planificabilidad extremo-extremo propuestos, permiten diseñar WSAN que cumplen con las restricciones de tiempo impuestas por los algoritmos de control.
- Para garantizar el cumplimiento de los plazos de tiempo extremo-extremo utilizando la arquitectura de nodos y los test de planificabilidad propuestos, no es necesario que los procesadores encargados de la comunicación en los nodos estén sincronizados con aquellos que desarrollan la aplicación.
- En el análisis de los NCS se encontró una gran similitud entre los resultados utilizando métodos formales, simulaciones y la implementación física del sistema. De lo cual es posible concluir que entornos de simulación como Truetime permiten un análisis confiable y rápido de este tipo de aplicaciones, lo que facilita el proceso de diseño de las mismas.
- El uso de protocolos CSMA-CA, como el empleado en Zigbee para la comunicación en redes con arquitectura *mesh*, no garantiza un retraso máximo en la comunicación, lo cual puede degradar considerablemente el desempeño de los NCS.
- Los resultados obtenidos de la optimización en los casos de estudio corroboran la validez y la efectividad de las funciones de costo y de la metodología de diseño propuestas, y permiten apreciar la importancia de obtener configuraciones de funcionamiento que realicen un balance entre la potencia consumida en los nodos y los retrasos presentes en aplicaciones de control sobre WSAN, manteniendo los retrasos acotados; con lo cual se logra cumplir con las especificaciones planteadas por los objetivos de control y se logra incrementar el tiempo de vida de las aplicaciones.

- Los modelos de los componentes desarrollados en CPN permiten verificar propiedades de comportamiento y estructurales WSAN, cuya arquitectura de nodos corresponda con la propuesta en este trabajo. Adicionalmente permiten el análisis dinámico del sistema completo.
- Los resultados experimentales muestran como retrasos en los lazos de realimentación del orden de decenas de milisegundos, los cuales se pueden presentar utilizando la tecnología actual, pueden llegar a degradar considerablemente la respuesta de NCS implementados sobre WSAN en los que las constantes de tiempo sean de orden similar a estos retrasos. También se aprecia que utilizando las tecnologías seleccionadas y las consideraciones de diseño realizadas en esta tesis, es posible implementar WSAN cuyos resultados experimentales concuerdan con los objetivos de diseño propuestos.

Los resultados de la revisión bibliográfica, las simulaciones y los experimentos, obtenidos durante el desarrollo de esta tesis han sido presentados en (Martínez, 2008), (Martínez, 2009a, b, c), (Martínez, 2010a, b, c), (Martínez, 2011).

8.2 LÍNEAS FUTURAS DE INVESTIGACIÓN

Ante las perspectivas de las aplicaciones, los resultados obtenidos y considerando los desafíos de investigación presentados en el capítulo 2, se pueden destacar cuatro líneas futuras de investigación:

- La consideración de todas las posibles configuraciones de operación en los diferentes escenarios durante el proceso de optimización, permite encontrar la mejor configuración pero requiere un elevado tiempo de análisis; por tanto es importante desarrollar una técnica soportada en la función de costo propuesta, y en herramientas computacionales, que automatice el proceso de análisis, e integre heurísticas que faciliten el proceso de diseño y optimización en grandes aplicaciones.
- Desarrollar un middleware que se soporte en la arquitectura de nodo propuesta y gestione parámetros de calidad de servicio en las aplicaciones, y desarrollar su modelo en CPN con lo cual se facilite el diseño y análisis de las aplicaciones.
- Desarrollar métodos de análisis y compensación de los efectos producidos sobre algoritmos de procesamiento de señales y control, implementados sobre la arquitectura propuesta en este trabajo.
- Desarrollar nuevas arquitecturas de nodos que permitan incluir en el diseño estrategias adicionales de desarrollo de sistemas embebidos, como el codiseño hardware-software, que posibiliten la distribución de componentes al interior del nodo y contribuyan a encontrar soluciones donde haya un mejor balance del uso de los recursos, como la memoria de los procesadores, y de los requisitos de las aplicaciones, como el consumo de potencia y los retrasos en la generación de acciones.

BIBLIOGRAFÍA

- Alemdar H., Ersoy C.: “*Wireless sensor networks for healthcare: A survey*”. Computer Networks 54 (2010), pp. 2688–2710. 2010 Elsevier.
- Árzen K. J.: “*A simple event-based PID controller*”, Proceedings of 14th IFAC World Congress. vol. 18. Beijing, China. 423-428. 1999.
- Astrom K. and Wittenmark B.: “*Computer Control Systems. Theory and design*”. Prentice Hall, 1997.
- Audsley N., Burns A., Richardson M., Tindell K., and Wellings A.J.: “*Applying New Scheduling Theory to Static Priority Pre-emptive Scheduling*”. Software Engineering Journal, September 1993.
- Bakshi, A. ; Prasanna, V.K.: “*Algorithm Design and Synthesis for Wireless Sensor Networks*”. Proceedings of the 2004 International Conference on Parallel Processing (ICPP'04), IEEE Computer Society. ISSN : 0190-3918. ISBN: 0-7695-2197-5.
- Balbastre P., Ripoll I., and Crespo A.: “*Minimum Deadline Calculation for Periodic Real-Time Tasks in Dynamic Priority Systems*”. IEEE Transactions on computers, vol. 57, No. 1, January 2008.
- Banâtre M., Marrón P. J., Ollero A., Wolisz A.: “*Cooperating Embedded Systems and Wireless Sensor Networks*”. Wiley, 2008.
- Bonivento A., Sangiovanni-Vincentelli A., Graziosi F., Santucci F.: “*SERAN: A Semi Random Protocol Solution for Clustered Wireless Sensor Networks*”, Proc. of MASS 2005. 2005.
- Bonivento A., Carloni L. and Sangiovanni-Vincentelli A.: “*Platform-Based Design of Wireless Sensor Networks for Industrial Applications*”. Proceedings of the conference on Design, automation and test in Europe, pp. 1103-1107. 2006a.
- Bonivento A., Carloni L. and Sangiovanni-Vincentelli A.: “*Platform based design for wireless sensor networks*”. Mobile Networks and Applications - Springer, pp.469–485. 2006b.
- Branicky M., Phillips M., and Zhang W.: “*Scheduling and feedback codesign for networked control systems*”. In Proceedings of IEEE Conference on Decision and Control, Las Vegas, USA. 2002.
- Broustis I., Jakllari G., Repantis T., Molle. M.: “*A Comprehensive Comparison of Routing Protocols for Large-Scale Wireless MANETs*”. Sensor and Ad Hoc Communications and Networks, SECON '06. 2006 3rd Annual IEEE Communications Society, 2006.

- Campanoni S., Fornaciari W.: "*Multi-Level Design and Optimization of Wireless Sensor Networks*". Networked Sensing Systems, 2008. INSS 2008. 5th International Conference on, 17-19 June 2008, pp. ISBN: 978-4-907764-31-9. IEEE 2008.
- Cena I G., Bertolotti A C., C. Zunino V.: "*Industrial Applications of IEEE 802.11e WLANs*". IEEE International Workshop on Factory Communication Systems, 21-23 May 2008.
- Cervin A., Henriksson D., Lincoln B., Eker J., Årzén K.: "*How Does Control Timing Affect Performance?, Analysis and Simulation of Timing Using Jitterbug and TrueTime*". IEEE Control Systems Magazine, June 2003.
- Dormido S., Sánchez J., Kofman E.: "*Muestreo, control y comunicación basados en eventos*". Revista Iberoamericana de Automática e Informática Industrial. Vol. 5. No. 1, pp. 5-26, 2008.
- Gregory C W., Octavian B., Linda B.: "*Asymptotic Behavior of Networked Control Systems*". Submitted to Control Applications Conference, International Conference on Control Applications. Hawaii, USA. August 22-21, 1999.
- Hespanha J. P., Xu Y.: "*A Survey of Recent Results in Networked Control Systems*". Proceedings of the IEEE, Vol. 95, No. 1, January 2007.
- Hristu-Varsakelis D., Levine W. S., (Eds.): "*Handbook of Networked and Embedded Control Systems*". Pag: 677-720. Birkhäuser 2005.
- Hu S., Yan W.: "*Stability of Networked Control Systems Under a Multiple-Packet Transmission Policy*". IEEE Transactions on Automatic Control, Vol. 53, pp. 1706-1711. August 2008.
- Huang D., Nguang S. K.: "*State Feedback Control of Uncertain Networked Control Systems With Random Time Delays*". IEEE Transactions on Automatic Control, Vol. 53, pp. 829-834. April 2008.
- IEEE Standard for Information technology— Telecommunications and information exchange between systems— 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). IEEE Std 802.15.4™-2006 (Revision of IEEE Std 802.15.4-2003).
- Jensen K.,: "*An Introduction to the Theoretical Aspects of Coloured Petri Nets*". In: J.W. de Bakker, W.-P. de Roever, G. Rozenberg (eds.): A Decade of Concurrency, Lecture Notes in Computer Science Vol. 803, p. 230–272. Springer-Verlag, 1994.
- Jensen K.,: "*A Brief Introduction to Coloured Petri Nets*". TACAS '97: Proceedings of the Third International Workshop on Tools and Algorithms for Construction and Analysis of Systems. p. 203-208, Springer-Verlag, 1997.
- Jensen K., Kristensen L. M. and Wells L.: "*Coloured Petri Nets and CPN Tools for modelling and validation of concurrent systems*". Int. J. Softw. Tools Technol. Transf. volume = 9, number = 3, p. 213-254, Springer-Verlag, 2007.

-
- Ko J., Lu Ch., Srivastava M. B., Stankovic J. A., Terzis A., and Welsh M.: “*Wireless Sensor Networks for Healthcare*”. Proceedings of the IEEE | Vol. 98, No. 11, November 2010, pp. 1947 – 1960.
- Koubâa A., Cunha A., Alves M., Tovar E.: “*TDBS: a time division beacon scheduling mechanism for ZigBee cluster-tree wireless sensor networks*”, Real-Time Systems, Springer Netherlands, vol. 40, num. 3, 2008.
- Langendoen K., Baggio A., Visser O.: “*Murphy Loves Potatoes: experiences from a pilot sensor network deployment in precision agriculture*”. Parallel and Distributed Processing Symposium, 2006. IPDPS 2006. 20th International. 2006. ISBN: 1-4244-0054-6.
- Lee S., Park J. H., Ha K. N., Lee K. C.: “*Wireless Networked Control System Using NDIS-based Four-Layer Architecture for IEEE 802.11b*”. IEEE International Workshop on Factory Communication Systems, 21-23 May 2008.
- Lester H. J.: “*System architecture for wireless sensor networks*”. Doctoral thesis. University of California, Berkeley. 2003.
- Li Y., Thai M. T., Wu W.: “*Wireless Sensor Networks and Applications*”. Springer. 2008.
- López J.A., Soto F., Suardíaz J., Sánchez P., Iborra A., Vera J.A.: “*Wireless Sensor Networks for precision horticulture in Southern Spain*”. Computers and Electronics in Agriculture 68 (2009) 25–35. 2009 Elsevier.
- Mangharam R., Rowe A., Rajkumar R.: “*FireFly: a cross-layer platform for real-time embedded wireless networks*”. Real-Time Systems. Springer, 2007.
- Marinoni M., Buttazzo G.: “*Elastic DVS Management in Processors With Discrete Voltage/Frequency Modes*”. IEEE Transactions on industrial informatics, vol. 3, No. 1, 2007.
- Martínez D., Blanes F., Simo J. and Crespo A.: “*Wireless Sensor and Actuator Networks: Characterization and Cases Study for Confined Spaces Healthcare*”. Computer Science and Information Technology, 2008. IMCSIT 2008. International Multiconference on. IFAC Workshop on Real-Time Programming. Pag: 687 – 693. ISBN: 978-83-60810-14-9.
- Martínez D., Blanes F., Simo J. and Crespo A.: “*Wireless Sensor and Actuator Networks: Characterization and Cases Study for Confined Spaces Healthcare and Control Applications*”. Scalable Computing: Practice and Experience, Scientific International Journal for Parallel and Distributed Computing, Pages 291–305, Volume 10, no. 3. <http://www.scpe.org>. ISSN: 1895-1767. September, 2009a.
- Martínez D., Balbastre P., Blanes F., Simo J. and Crespo A.: “*Schedulability Analysis and Performance Evaluation of WSA*”. Preprints of the 30th IFAC Workshop on Real-Time Programming and 4th International Workshop on Real-Time Software, pp. 109 – 116. ISSN 1896-7094. Mrągowo - Poland, October 12-14, 2009b.
- Martínez D., Balbastre P., Blanes F., Simo J. and Crespo A.: “*Design of control applications on WSA with mesh architecture*”. Ibero-American Conference on Trends in

- Engineering: Wireless Sensors Networks Session. ISBN: 978-0-9842912-2-9. Albuquerque, NM-USA. October 27-28, 2009c.
- Martínez D., Balbastre P., Blanes F., Simo J. and Crespo A.: "*Procedimiento de diseño para minimizar el consumo de potencia y los retrasos en WSN*". Revista Iberoamericana de Automática e Informática Industrial – RIAI Vol. 7, Núm. 3, Julio 2010. 2010a. .
- Martínez D., Balbastre P., Blanes F., Simo J. and Crespo A.: "*Design procedure to minimize power consumption and delays in WSN*". ETFA10 - IEEE International Conference on Emerging Technology and Factory Automation, 2010b. ISSN: 1946-0740, 2010b.
- Martínez D., Blanes F., Simo J., Crespo A. y González A.: "*Diseño y validación de redes inalámbricas de sensores y actuadores utilizando redes de Petri coloreadas*". CONGRESO ESPAÑOL DE INFORMÁTICA (CEDI), III Simposio de Sistemas de Tiempo Real. IBERGARCETA PUBLICACIONES, S.L., Madrid, 2010. ISBN: 978-84-92812-64-6. 2010c.
- Martínez D., González A., Blanes F., Aquino R., Simo J., Crespo A.: "*Formal Specification and Design Techniques for Wireless Sensor and Actuator Networks*". Sensors 2011, 11(1). ISSN 1424-8220. January 2011.
- Mejia-Alvarez P., Levner E., and Mosse D.: "*Power-Optimized Scheduling Server for Real-Time Task*". Proc. IEEE Real-Time and Embedded Technology and Applications Symp. (RTAS '02), p. 239, 2002.
- Meshkova E., Riihijarvi J., Oldewurtel F., Jardak Ch. and Mahonen P.: "*Service-Oriented Design Methodology for Wireless Sensor Networks: A View through Case Studies*". IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing. 2008.
- Moraes R., Portugal P., Vasques F., Fonseca J. A.: "*Limitations of the IEEE 802.11e EDCA Protocol when Supporting Real-Time Communication*". IEEE International Workshop on Factory Communication Systems, 21-23 May 2008.
- Palencia J.C., Gutierrez J. J., Gonzalez M.: "*On the schedulability analysis for distributed hard real-time systems*". Real-Time Systems, 1997. Proceedings., Ninth Euromicro Workshop on 11-13 June 1997 Page(s):136 143.
- Pan J., Tompkins Willis J.: "*A Real-Time QRS Detection Algorithm*". IEEE Trans. Biomed. Eng., vol. BME-32, Pages: 230-236, 1985
- Pantazis, N.A.; Vergados, D.D.: "*A survey on power control issues in wireless sensor networks*". IEEE Communications Surveys & Tutorials, 4th Quarter 2007. vol 9, No. 4.
- Perkins C., Royer E., Das S., Marina M.: "*Performance Comparison of Two On-Demand Routing Protocols for Ad Hoc Networks*". IEEE INFOCOM, 2000.
- Pillai P. and Shin K.G.: "*Real-Time Dynamic Voltage Scaling for Low-Power Embedded Operating Systems*". Proc. ACM Symp. Operating Systems Principles, pp. 89-102, 2001.

-
- Prasad V., Yan T., Jayachandran P., Li Z., Son S., Stankovic J., Hansson J. and Abdelzaher T.: “ANDES: an ANalysis-based DEsign tool for wireless Sensor networks”. 28th IEEE International Real-Time Systems Symposium, IEEE Computer Society, 2007.
- Ripoll I., Crespo A., and Mok A.: “Improvement in feasibility testing for real-time tasks”. Journal of Real-Time Systems, 11:19–40, 1996.
- Römer K. and Mattern F.: “The Design Space of Wireless Sensor Networks”. IEEE Wireless Communications, 2004. Vol. 11. Pag.: 54 – 61. ISSN : 1536-1284.
- Saewong S. and Rajkumar R.: “Practical Voltage-Scaling for Fixed-Priority RT-Systems”. Proceedings of the 9th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS’03) 2003 IEEE.
- Salt J., Casanova V., Cuenca A., Pizá R.: “Sistemas de control basados en red modelado y diseño de estructuras de control”. Revista Iberoamericana de Automática e Informática Industrial. Vol. 5. No. 3, pp. 5-20, 2008.
- Shen Ch., Badr C., Kordari K., Bhattacharyya S.S., Blankenship G.L., Goldsman N.: “A Rapid Prototyping Methodology for Application-Specific Sensor Networks”. Computer Architecture for Machine Perception and Sensing, 2006. CAMP 2006. International Workshop on Digital Object Identifier: 10.1109/CAMP.2007.4350368 Publication Year: 2007, pp: 130 – 135.
- Spuri M.: “Holistic Analysis for Deadline Scheduled Real-Time Distributed Systems”. Tech. Rep. RR-2873, INRIA, France, April 1996.
- Streichert Th., Haubelt Ch., Teich J.: “Multi-Objective Topology Optimization for Networked Embedded Systems”. Embedded Computer Systems: Architectures, Modeling and Simulation, 2006. IC-SAMOS 2006. International Conference on. ISBN: 1-4244-0155-0. 2007.
- Swami A., Zhao Q., Hong Y. Tong L.: “Wireless Sensor Networks, Signal Processing and Communications Perspectives”. Wiley, 2007.
- Tabbara, M., Nestic, D.; Teel, A.R.: “Stability of Wireless and Wireline Networked Control Systems”. Automatic Control. IEEE Transactions on Automatic Control, Vol. 52, pp. 1615-1630. September 2007.
- Tariq S. and Anuradha A.: “The Impact of Control Technology. Overview, Success Stories, and Research Challenges”. IEEE Control Systems Society, February 2011.
- Tindell K. and Clark J.: “Holistic Schedulability Analysis for Distributed Hard Real-Time Systems, Microprocessors and Microprogramming 40”, 1994.
- Tindell K., Burns A., and Wellings A.J.: “Analysis of Hard Real-Time Communications”, The Journal of Real-Time Systems 9, 1995.
- TinyOS: <http://www.tinyos.net/>

- Tompkins W. J., Webster J. G.: "*Desing of microcomputer-based medical instrumentation*". New Jersey: Prentice-Hall, 1981. Pages: 396-397.
- Varma A., Debes E., Kozintsev I. and Jacob B.: "*Instruction-Level Power Dissipation in the Intel XScale Embedded Microprocessor*". Proceedings of the SPIE, 17th Annual Symposium on Electronic Imaging Science & Technology, vol. 5683, pp. 1-8. 2005
- Walsh G.C., Ye H., and Bushnell L.: "*Stability analysis of networked control systems*". IEEE Transactions on Cotrol Systems Technology, Vol. 10, no.5, pp.438-446, 2002.
- WirelessHART:
http://www.hartcomm2.org/hart_protocol/wireless_hart/wirelesshart_datasheet.pdf
- Xiong J., Lam, J.: "*Stabilization of Networked Control Systems With a Logic ZOH*". IEEE Transactions on Automatic Control, Vol. 54, pp. 358-363. February 2009.
- Yang T.C.: "*Networked control system: a brief survey*". IEE Proc.-Control Theory Appl., Vol. 153, No. 4, July 2006.
- Yick J., Mukherjee B., Ghosal D.: "*Wireless sensor network survey*". Computer Networks 52 (2008) 2292–2330. 2008 Elsevier.
- Zhang W., Branicky M., and Phillips S.: "*Stability of networked control systems*". IEEE Control Systems Magazine. Pp. 84–99. 2001.
- Zhu Y. and Mueller F.: "*Feedback EDF Scheduling Exploiting Dynamic Voltage Scaling*". Proceedings of the 10th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS'04), 2004 IEEE.
- Zigbee Specification. <http://www.Zigbee.org>

