RIELAC, Vol. 41 2/2020 p. 39-50 Mayo-Agosto ISSN: 1815-5928

RIELAC
Revista de Ingeniería Electrónica, Automática y Comunicaciones

Redes de Sensores Inalámbricos Definidas por Software: revisión del estado del arte

Lidice Romero Amondaray, Fernando José Artigas Fuentes, Caridad Anias Calderón

RESUMEN / *ABSTRACT*

En las redes de sensores inalámbricos definidas por software (SDWSN) se separa el plano de control del de datos al mover la lógica de control de los nodos sensores a un controlador central. Estas redes son un nuevo paradigma que promete simplificar la gestión de las redes de sensores inalámbricos (WSN) tradicionales. En este artículo se revisa el estado del arte de las arquitecturas SDWSN, haciendo especial hincapié en el desarrollo de los planos de datos y control y en las funcionalidades que brindan esas arquitecturas. El desarrollo de esta tecnología está aún en fase inicial y aunque trae beneficios para solucionar las problemáticas de las WSN, se deben superar los desafíos inherentes a las características propias de estas redes.

Palabras claves: WSN, SDN, SDWSN

Software defined Wireless Sensor Networks (SDWSN) decouples the control plane from the data plane, thus moving the control logic from the sensor node to a central controller. SDWSN is a new paradigm that promise to simplify the network management of traditional Wireless Sensor Networks (WSN). This paper presents a reviews of SDWSN architectures, with special emphasis on the development of the data and control plans and the functionalities provided by these architectures. The development of this technology is still in the initial phase and although it brings benefits to solve the problems of the WSN, the inherent challenges of the WSN characteristics must be overcome.

Key words: WSN, SDN, SDWSN

Software Defined Wireless Sensor Network

1. -INTRODUCCIÓN

Las redes de sensores inalámbricos (WSN, del inglés *Wireless Sensor Network*) son cada vez más populares, especialmente en aplicaciones que requieren un monitoreo continuo de diversas variables del entorno, en las que el uso de sensores tradicionales es inviable. La popularidad se debe principalmente a los bajos costos de instalación y mantenimiento, la alta flexibilidad y escalabilidad en el despliegue que brinda la tecnología. Estas redes están formadas por dispositivos equipados con sensores que colaboran para realizar una tarea común [1]. Estos dispositivos, conocidos como nodos sensores o motes, son unidades que captan información (con sensores tradicionales, ej. mecánicos, térmicos, biológicos, químicos, ópticos y magnéticos entre otros), la procesan y luego la envían a un nodo sumidero que transmite la información a una pasarela (*gateway*) conectada a Internet.

Las características básicas de los nodos sensores son: tamaño reducido, bajo consumo de energía, unidad de procesamiento con capacidad limitada, memoria con la capacidad de almacenar poca información y capacidad de comunicarse con otros nodos a corta distancia. Estas características hacen que los nodos sensores tengan un tiempo de vida mayor con respecto a dispositivos tradicionales [2].

Las redes de sensores inalámbricos se caracterizan por ser redes desatendidas y sin infraestructura física preestablecida. Generalmente operan en entornos muy dinámicos y críticos: los nodos se distribuyen en un área geográfica determinada, donde el acceso físico no siempre es posible, o se suelen implementar en gran escala. Por lo tanto, el mantenimiento de la red para la reconfiguración, la recuperación de fallas o los problemas técnicos se vuelven poco prácticos [3].

Recibido: 5/9/2019 Aceptado: 1/6/2020

En los próximos años se espera un crecimiento en el uso de esta tecnología por su importante papel en la Internet de las cosas (IoT, del inglés *Internet of Things*), paradigma emergente que pretende la interconexión objeto a objeto con el fin de cooperar y lograr objetivos comunes [3]. Las WSN son esenciales para monitorear varios objetos en aplicaciones tales como ciudades inteligentes, atención médica inteligente, redes hidráulicas inteligentes, redes eléctricas inteligentes, agricultura inteligente y sistemas de transporte inteligentes, entre otros [1].

Sin embargo, las limitaciones de los nodos en cuanto a capacidad computacional, energía, almacenamiento y ancho de banda impiden que estén listas para asumir las demandas de IoT, haciendo necesario la búsqueda de nuevas soluciones a estos retos [3]. Además, está la necesidad de que estos nodos se mantengan funcionando en entornos adversos.

Muchas de estas problemáticas tienen su origen en la propia arquitectura WSN ya que los nodos constituyen sistemas autónomos equipados con todas las funcionalidades, desde el nivel físico hasta el de aplicación. Además del reenvío de tráfico, los nodos realizan funciones de control, lo que impacta negativamente en la utilización de la red.

En este contexto, las Redes Definidas por Software (SDN, del inglés *Software Defined Networking*) [4,5] se presentan como una excelente opción ya que, al separar los planos de control y datos, permiten descargar parte de las funcionalidades del nodo sensor al controlador, convirtiendo al primero en un nodo no inteligente y dejando al segundo el procesamiento de los datos, el enrutamiento y otras funcionalidades que consumen gran parte de la energía de la batería. Los nodos sensores habilitados para SDN cambian dinámicamente sus funcionalidades y propiedades en función de las solicitudes de detección.

Aprovechando la inteligencia centralizada del controlador es posible, en las WSN, desplegar nuevas aplicaciones y servicios en tiempo real, flexibilizar la gestión y utilizar de manera más eficiente los recursos de la red [6]. Este último aspecto es especialmente importante para ofrecer soluciones verdaderamente IoT pues permitiría que varias WSN desplegadas en una misma área puedan compartir sus recursos [7].

La aplicación del modelo SDN a las WSN ha dado lugar a una nueva terminología: Red de Sensores Inalámbricos Definidas por Software (SDWSN, del inglés *Software-Defined Wireless Sensor Network*), un nuevo paradigma para las redes inalámbricas de área personal que promete simplificar significativamente la gestión de la red.

En este artículo se realiza un estudio del estado del arte de las arquitecturas de red de sensores inalámbricos definidas por software. En el mismo se describen los retos de las WSN que se pueden resolver si se les aplica el nuevo paradigma SDN, se revisan las principales arquitecturas SDWSN encontradas en la revisión bibliográfica realizada.

2.- PRINCIPALES RETOS DE LAS WSN QUE SE PUEDEN RESOLVER CON SDN

Las redes definidas por software se presentan como una solución a los problemas de las WSN [3,6,8]. A continuación, se comentan los más frecuentes y críticos y cómo este nuevo paradigma puede contribuir a su solución:

- **Energía**: la alimentación mediante baterías es común en las WSN ya que se ubican en lugares donde no se dispone de una fuente de alimentación. En estas redes el consumo energético es un factor muy importante que se relaciona directamente con el tiempo de vida de la red. El paradigma de SDN es útil porque, en virtud del desacoplamiento, los nodos de reenvío se liberan de muchas de las funcionalidades computacionales de uso intensivo de energía [9,10]. Empleando SDN la mayoría de las funciones que consumen energía residirán en el controlador, que usualmente están conectados a una fuente de energía. Esto ahorra considerable energía en los nodos sensores y prolonga la vida útil de la WSN. En teoría debe ser así, aunque esta afirmación aún no se ha cuantificado y permanece abierta a futuras investigaciones [3].
- **Gestión de red**: la reconfiguración y el mantenimiento de los nodos sensores tienden a ser procesos complicados y tediosos, pues la administración no es flexible; también influyen los entornos donde se implementan. Estos desafíos se alivian con el enfoque SDN al eliminar la lógica de control de los nodos sensores, dejándolos como meros elementos de reenvío que se controlan y manipulan desde el controlador, lo que permite la programación de la infraestructura física [6].
- **Escalabilidad**: muy importante especialmente para el paradigma de IoT [11-13]. Las WSN se vuelven ineficientes a medida que crecen. El modelo de abstracción de SDN puede mantener la topología y eficiencia de la red intacta debido a que se le entrega la escalabilidad al controlador [14]. Aunque el modelo SDN tradicionalmente se basa en un controlador central, se han realizado esfuerzos para crear un plano de control distribuido [15,16].
- **Enrutamiento, movilidad y localización**: en las WSN los dispositivos son fijos o móviles. En dependencia de la naturaleza del despliegue, la movilidad y la localización de los nodos móviles son aspectos críticos en el enrutamiento

en las redes de sensores. Normalmente, los protocolos de enrutamiento tradicionales actualizan periódicamente la tabla de enrutamiento (proactivo) o, de lo contrario, generarán una ruta a pedido (reactivo) en un evento de cambio. Este proceso consume mucha energía y no es adecuado en las WSN. Con SDN la movilidad se maneja desde el controlador central, es decir, las decisiones y políticas de enrutamiento se administran en el controlador [17,18]. Los algoritmos de localización se implementan en este o en el plano de aplicación en lugar de los nodos sensores con recursos limitados [19]. Esto ayuda al descubrimiento de la topología de la red y, posteriormente, a mejorar la toma de decisiones.

- **Interoperabilidad**: las redes de sensores son dependientes de la aplicación y del proveedor, lo que conduce a la subutilización de recursos. Este problema se puede resolver utilizando el enfoque SDN que elimina la dependencia de los proveedores. Con SDN, los elementos de infraestructura se controlan desde un punto central, ejecutando un protocolo común, aunque estos sean de diferentes fabricantes.
- **Comunicación**: la comunicación física es ampliamente manejada por el dispositivo, pero aspectos tales como el acceso al medio y la programación de los ciclos de trabajo (del inglés *duty cycling*) [20] pudieran ser decisiones tomadas con mayor eficiencia por el controlador. SDN también mejora el control de infraestructuras de redes heterogéneas. Así la comunicación entre las SDWSN y otras redes se puede manejar adecuadamente desde el controlador.
- **Seguridad**: la centralización de la administración de la seguridad simplifica la implementación y configuración de los mecanismos de seguridad [21]. Esta perspectiva global también permite el monitoreo proactivo y la evaluación del estado de la red, lo que propicia una respuesta rápida ante un ataque [3]. Como los nodos sensores se convierten en elementos que solo entienden comandos o mensajes del controlador, es más difícil que se use un nodo como elemento malicioso si se garantiza la seguridad desde los controladores. Además, SDN permite una configuración automática, eliminando los errores que se producen al reconfigurar manualmente los nodos de la red.

3.- ARQUITECTURAS WSN BASADAS EN SDN

En la Figura 1 se muestra la arquitectura general de las WSN basadas en SDN. Este nuevo paradigma desacopla la inteligencia de la red del plano de control, de los procesos de reenvío de paquetes en el plano de datos [3,4]. Esta separación permite centralizar la inteligencia de la red en el controlador, que posee una vista global de toda la red. Bajo esta filosofía, las redes de sensores inalámbricos definidos por software reorganizan las funcionalidades en tres capas: infraestructura, control y aplicación.

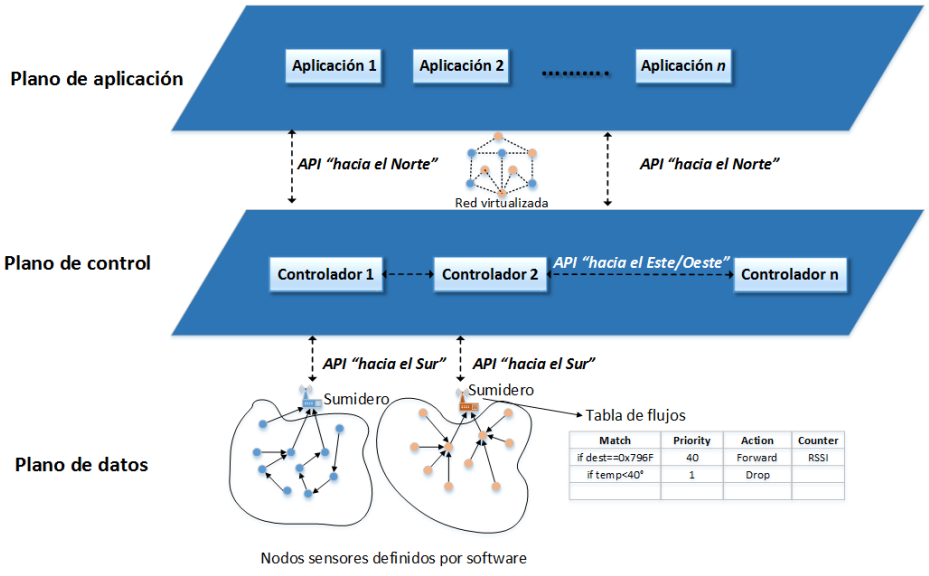

Figura 1

Arquitectura de Red de Sensores Inalámbricos Definidos de Software

En la capa de infraestructura, los nodos sensores definidos por software realizan tareas de seguimiento y control de objetos, o de monitoreo. Las aplicaciones de monitoreo incluyen la monitorización de ambientes interiores o exteriores, de pacientes hospitalarios, procesos industriales e infraestructuras y militares, entre otras [22-29]. Las de seguimiento y control incluyen el seguimiento de objetos, animales, seres humanos y vehículos. Además de esto, recogen información sobre el estado de la red (topología, estadísticas de tráfico, estado de los nodos) que se envía al controlador para la gestión de los nodos sensores que conforman la red

La información que se obtiene en las redes de sensores inalámbricos definidos por software se transmite a través de flujos organizados en una tabla que contiene las reglas que indican cómo reenviar los paquetes. Los nodos sensores mantiene una tabla de flujos (que se modifica dinámicamente por el controlador) que consta de reglas de flujo (entradas) que determinan como se manejan los paquetes en la red [9,23,24]. Las entradas de flujo consisten principalmente en campos de coincidencia, contadores e instrucciones/acciones. La entrada del campo de coincidencia se utiliza para hacer coincidir los paquetes entrantes. La capa de control puede estar formada por uno o más controladores.

En el nivel más alto de la arquitectura radican las aplicaciones de gestión de la topología de la red, enrutamiento, seguridad, entre otras y las de monitoreo, seguimiento o control de parámetros del mundo real. Como el plano de control esconde la complejidad de la red física a la capa de aplicación, esta tiene una visión virtualizada de la red lo que permite obtener varias redes lógicas con una misma infraestructura.

Aplicar los conceptos SDN a las WSN pueden resultar un reto si se tienen en cuenta las limitaciones de energía, memoria y recursos computacionales de los nodos. Una de las soluciones que se debe implementar es el ciclo de trabajo [10], esta técnica permite ahorrar energía al suspender la radio de los nodos durante períodos de tiempo. Otra, también para ahorrar energía, es la agregación de datos en algunos nodos de la red para eliminar redundancia. Ambos mecanismos, aunque permiten ahorrar energía, pueden provocar demoras en la red [30,31].

Las WSN también se pueden beneficiar al usar SDN para distribuir la carga de tráfico entre los nodos de la red, centralizando el enrutamiento en un elemento de control. No obstante, para implementar estos mecanismos se requiere que la interfaz “hacia al sur” tenga una amplia gama de definiciones de flujos y acciones. El principal problema está en que el protocolo oficial para la conexión remota entre el controlador SDN y los dispositivos SDN [23], *OpenFlow* [25,26], utiliza las direcciones Ethernet, IP (IPv4 e IPv6), los puertos TCP/UDP o las etiqueta MPLS [32] para la creación de los flujos. En las WSN es más importante la adquisición de los datos que conocer quién los envía, además de que emplean un mecanismo de direccionamiento diferente, mediante atributos, para especificar las propiedades de los datos (ej. los nodos que detectan una temperatura>30° [33].

A continuación, se describen las principales contribuciones a la arquitectura de la red de sensores definida por software.

3.1.- NODO SENSOR DEFINIDO POR SOFTWARE

El nodo sensor definido por software es un dispositivo compuesto por una serie de elementos básicos como: los sensores, un módulo de procesamiento (microcontrolador), uno de transmisión/recepción (transceptor) y una fuente de alimentación, normalmente una batería (ver Figura 2). Dependiendo de la aplicación, la complejidad del nodo sensor puede incrementarse pudiendo existir elementos adicionales como: un generador de energía para evitar el reemplazo periódico de la batería, un sistema de localización o un sistema de posicionamiento si el nodo es móvil.

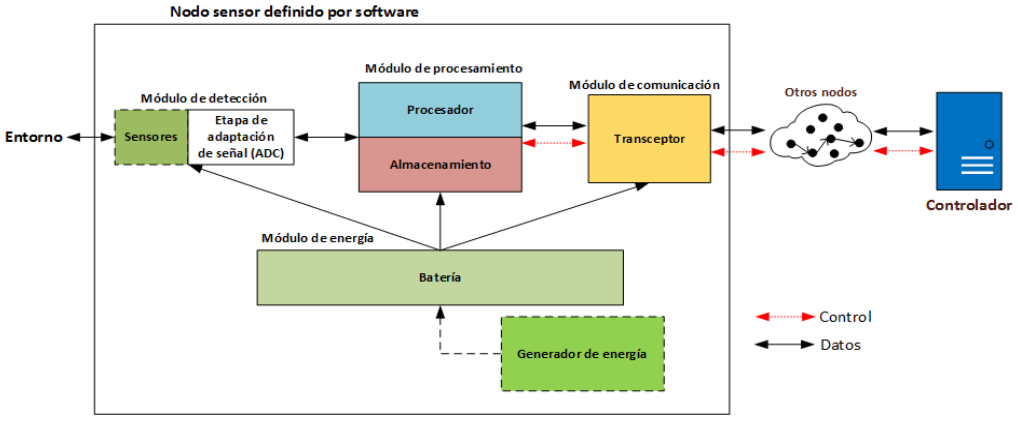

Figura 2

Componentes de un nodo sensor definido por software

En el transceptor se implementan las funciones de las capas física (PHY) y de control de acceso al medio (MAC) para redes de área personal de baja tasa de transmisión (LR-WPAN, del inglés *Low-Rate Wireless Personal Area Network*) [34]. En el microcontrolador se implementan funciones como el reenvío de datos, que se comunica con el controlador para mantener actualizadas las tablas de flujo, la agregación de datos y recolección de información para que los controladores tengan una vista de la topología de la red. La comunicación entre el nodo y el controlador sigue los principios del protocolo *OpenFlow*. Ejemplos de arquitecturas donde se aplica este enfoque son SDWN (*Software Defined Wireless Networks*) de Costanzo *et al.* [35], y SDN-WISE (*Software Defined Networking solution for WIreless SEnsor networks*) de Galluccio *et al.* [19,36].

En la arquitecta SDWSN [14], de Jacobsson y Orfanidis, la capa física se reconfigura para modificar, de manera flexible, la red y sus parámetros. La capa MAC maneja la identificación de los nodos, uno de los mayores retos de las WSN. Los nodos sensores están equipados con máquinas virtuales que ayudan a instalar nuevos protocolos o actualizaciones de código cuando sea necesario. Las funcionalidades que desempeña el nodo sensor de esta arquitectura consumen muchos recursos tiendo en cuenta la escasa capacidad de cómputo y limitación de energía de los nodos sensores.

La arquitectura SD-WSN de Luo *et al.* [33] se centra en la adaptación del protocolo *OpenFlow* a las particularidades de las WSN. Proponen el protocolo *Sensor OpenFlow* (SOF) que soporta agregación de datos en la red *(in-network packet processing*) y varios tipos de direcciones para WSN. En este protocolo se redefinen las tablas de flujo *OpenFlow* para atender los esquemas de direccionamiento especiales de las WSN.

En SD-WSN los autores definen dos tipos de direcciones para el *Sensor OpenFlow*: Class-1 y Class-2. Aprovechan que, en *OpenFlow*, el campo que se utiliza para identificar los flujos (OXM, del inglés *OpenFlow Extensible Match*) permite el desarrollo de expresiones de identificación más flexibles y la incorporación de nuevos campos de identificación [32] con una estructura tipo-longitud-valor (TLV, del inglés *Type, Length, Value*). Las direcciones Class-1 son direcciones de 16 bits como las de ZigBee (por ejemplo, 0x796F) y las Class-2 son pares concatenados valor-atributo como “30<temperature<60” y “Zone-ID=7 AND x-coordinate>150”. También dejan abierta la posibilidad de usar adaptaciones del protocolo IP a las WSN como uIP, uIPv6 [38,39] basadas en el sistema operativo específico para redes de sensores Contiki [40], y Blip, implementación de IP de Berkley para redes de baja potencia basada en TinyOS, otro sistema operativo para redes de sensores [41,42].

SD-WSN, si bien establece las ideas preliminares para la creación de WSN versátiles, flexibles y fáciles de gestionar, se centra principalmente en el nodo sensor definido por software y la interfaz “hacia el sur”, o sea en la interacción entre el plano de control y el de datos. Los autores no especifican cómo gestionar el canal radio, el consumo de energía en los nodos, la seguridad y la topología de la red, entre otros importantes aspectos que atañen a este tipo de redes. Además, el análisis del formato TLV, usado para redefinir la identificación de los nodos sensores en el protocolo *Sensor OpenFlow*, introduce campos de encabezado que son irrelevantes para los enrutadores WSN que poseen *buffers* limitados.

La arquitectura SDWSN cognitiva [43], de Huang *et al.*, para mejorar la eficiencia energética y la adaptabilidad de las WSN en aplicaciones de monitoreo ambiental, teniendo en cuenta sus limitaciones en términos de energía, recursos radio y capacidad computacional, además de la redundancia y naturaleza distribuida de los flujos de datos que generan las transmisiones periódicas de las aplicaciones de monitoreo. Los autores diseñan un mecanismo basado en aprendizaje por refuerzo (RL, del inglés *Reinforcement Learning*) para filtrar la redundancia en los datos, realizar el enrutamiento con balanceo de carga para la distribución de los flujos de datos, lo que mejora la eficiencia energética y la adaptabilidad de las WSN a los cambios ambientales. Esta arquitectura usa el protocolo *Sensor OpenFlow* para la comunicación entre el plano de datos y el de control. Las reglas de coincidencias en la tabla de flujo están diseñadas para frenar la sobrecarga de señalización de control y equilibrar la distribución de flujos de datos para lograr la fusión dentro de la red en el plano de datos con la calidad de servicio (QoS) garantizada.

En la arquitectura TinySDN [44,45], de Oliveira *et al.*, el nodo sensor funciona como un conmutador SDN y como un dispositivo SDN final. Este nodo se divide en tres partes: *ActiveMessageC*, *TinySdnP* y *TinyOS Application*. *ActiveMessageC* es un componente de TinyOS que se encarga del acceso al medio y las comunicaciones inalámbricas. *TinySdnP* es el componente principal de la arquitectura, comprueba si los paquetes que se reciben cumplen con una entrada de la tabla de flujos; si se cumple una condición, se ejecuta la acción asociada y si no se solicita al controlador cómo tratar el paquete. *TinyOS Application* es equivalente al dispositivo final y genera los datos en dependencia de la aplicación para la que fue diseñada la WSN.

3.2.- CONTROLADOR

El controlador constituye un aspecto clave de las redes de sensores definidas por software. La función principal de este componente de la arquitectura SDWSN es la generación de las reglas para las tablas de flujo en los nodos. Para esto, mantiene una vista actualizada del estado y topología de la red. El controlador intercambia mensajes de supervisión con los nodos que le permiten adquirir información sobre el nivel de las baterías, vecinos y estado de los enlaces, entre otros.

En los trabajos consultados, la implementación del controlador sigue varias variantes: en el sumidero o en un nivel más alto; incluso algunos autores proponen descargar algunas funcionalidades de control en el nodo sensor. El control por lo tanto puede ser centralizado, distribuido e híbrido.

La centralización de la inteligencia de la red en un sólo controlador constituye la implementación básica de las SDN en las WSN. Aunque este enfoque trae beneficios, porque con una visión de la red global el controlador puede tomar mejores decisiones que los nodos, también presenta varios desafíos. El principal es que toda la red está en riesgo si el controlador central se ve comprometido. La falla del controlador compromete la disponibilidad de la red, por lo que no es una configuración confiable. Otro desafío importante es que el rendimiento y la eficiencia se afectan a medida que la red crece como resultado de la sobrecarga. Ejemplos de arquitecturas con el control centralizado son SD-WSN [33], SDWN [35], Flow-Sensor [46], SDWSN [47], SDSN [48], SensorSDN [49] y *Soft-WSN* [13].

En la arquitectura *Flow-Sensor* [46], de Mahmud y Rahmani, se separa el control de la transmisión de los paquetes de datos. El plano de control usa el protocolo *OpenFlow*, como fue concebido para las redes cableadas, por un canal seguro, mientras que los datos se transmiten en otro canal usando TCP/IP. El problema con esta arquitectura es que TCP/IP, ampliamente usado en las redes cableadas e inalámbricas, es considerado muy complejo para implementarlo en dispositivos de escasos recursos como los nodos sensores [33,50]. Lo mismo pasa con *OpenFlow* concebido para dispositivos de red cableada por lo que muchas de sus funcionalidades son innecesarias en los nodos sensores [51]. Además, el control en las WSN se realiza en banda por las mismas limitaciones de recursos.

La arquitectura SDWN [35] de Costanzo *et al.* está compuesta por un nodo genérico y un sumidero (*sink*). El sumidero está divido en dos partes: sumidero y controlador embebido en un SO Linux, ambas partes conectadas por USB, una conexión serie u otra interfaz de comunicación. El SO embebido está compuesto de un módulo de adaptación, un virtualizador y un controlador. El módulo de adaptación se encarga de dar formato a los mensajes para que puedan ser procesados por los nodos de la red. El virtualizador se ocupa de construir, a partir de la información que recibe de cada nodo genérico, una representación detallada y consistente del estado de la red que incluye la topología, nivel de energía de cada nodo y calidad del enlace entre nodos, entre otros. Este componente del SO embebido permite la coexistencia entre varias redes lógicas desplegadas sobre los dispositivos físicos de la red. Por su parte, el controlador genera las entradas de las tablas de flujo cuando un nodo recibe un paquete que no puede ser clasificado. La arquitectura SDWN tiene un conjunto de reglas para clasificar los paquetes que circulan por la red y soporta el ciclo útil de radio. Los autores no precisan como se realiza la implementación del control distribuido. El punto más débil de esta arquitectura está en la baja escalabilidad que brinda la comunicación entre el controlador y el sumidero.

En la arquitectura de Gante *et al*. [47] se implementa en estación base para una WSN basada en SDN. En su propuesta, la estación base está compuesta por cinco capas: física, acceso al medio, sistema operativo de red, una capa intermedia (*middleware*) y la de aplicación. La capa intermedia está formada por un controlador, un módulo de mapeo de funciones, uno de mapeo de información y otro donde se definen las tablas de flujo. El controlador gestiona la red y mantiene actualizados su estado y topología, para lo que emplea mensajes de supervisión que le permiten adquirir información como: niveles de energía de los nodos sensores, distancia a la BS, lista de vecinos y parámetros de estado del enlace (calidad de enlace, tiempo de respuesta, etc.). El módulo de mapeo de funciones procesa los datos adquiridos por los nodos y crea un mapa de red (vista topológica). La información adquirida de la red se almacena en el módulo de mapeo de información. En la capa de aplicación se implementan algoritmos de localización y seguimiento para mantener información precisa sobre la posición de cada nodo dentro del área que se va a supervisar. Este trabajo se centra más en el controlador y no da detalles sobre la comunicación con el resto de los planos.

La arquitectura SDSN [48], de Zeng *et al.*, combina las WSN definidas por software con la computación en la nube para crear un nuevo paradigma: detección como servicio (*Sensing-as-a-Service*). Al igual que otros servicios en la nube, un usuario solicita la detección a un portal de entrada. Esta solicitud se envía a un controlador de núcleo que posteriormente la hace llegar a una WSN que ofrezca el servicio de detección. Para implementar la nueva tarea de detección, el controlador reprograma algunos nodos, distribuyéndoles un programa para ello. Solo los nodos sensores reprogramados cumplirán la tarea de detección dentro de su área de cobertura. Las SDN sirven para normalizar en una red común varias WSN que coexisten en un mismo espacio físico. La red resultante tiene la flexibilidad necesaria para cumplir con las solicitudes de detección de diferentes aplicaciones. Las ideas plasmadas en este trabajo son puramente teóricas pues no se conocen implementaciones prácticas.

Bera *et al.* propusieron en [13] una arquitectura WSN definida por software para tratar los requisitos específicos de IoT. En su trabajo presentan la arquitectura *Soft-WSN* que se centra en la gestión de los dispositivos sensores y de la topología de la red. El controlador está compuesto por dos componentes, uno para cada tipo de gestión. El gestor de dispositivo está concebido para reconfigurar, en tiempo real, las tareas que realizan los nodos sensores (adquisición de los datos, la demora en la

adquisición de los datos y el ciclo de vida útil de radio) y el gestor de topología es responsable de la conformación de la topología de la red para asegurar la calidad de servicio (QoS).

En [49], Hakiri y Gokhale presentan la arquitectura SensorSDN que puede ser utilizada por varios sistemas IoT. Los autores proponen nuevos servicios para el plano de control en aras de soportar descubrimiento automático de topología, movilidad y virtualización de los nodos, así como gestión de políticas de red. Además, proponen nuevos campos para la tabla de flujos teniendo en cuenta las tecnologías LR-WPAN que ya existen, así como nuevas reglas para el enrutamiento de paquetes. También introducen una capa MAC programable (pTDMA) para mejorar el procesamiento de los flujos. Esta arquitectura usa una adaptación del protocolo *OpenFlow* para la comunicación entre el plano de datos el de control.

Para superar los inconvenientes del controlador centralizado, en varias arquitecturas WSN basadas en SDN se propone distribuir la lógica de control. Este enfoque mejora la escalabilidad de la red, permite que la respuesta a consulta de los nodos sensores sea más rápida y permite implementar una estrategia de seguridad escalonada, colaborativa, entre otras. Ejemplo de este tipo de control son las arquitecturas TinySDN [45], SDWSN cognitiva [43] y SDCSN [52].

TinySDN [44,45] , de Oliveira *et al.*, está basada en TinyOS. Es una arquitectura redes de sensores definidos por software con múltiples controladores que tiene dos componentes principales: un nodo sensor SDN y un controlador SDN. El controlador SDN se compone de un nodo receptor y un controlador conectados por USB o una conexión serie. El controlador mantiene un mapa de la topología de la red y gestiona los flujos.

En TinySDN uno de los controladores está siempre cerca de los nodos finales. Cuando la red inicia, los nodos sensores tienen como primera tarea encontrar un controlador al que unirse. Este proceso de descubrimiento se realiza con el protocolo CTP (del Inglés C*ollection Tree Protocol*) [53], ampliamente utilizado por TinyOS para redes multisaltos. Este protocolo es independiente del hardware, da la posibilidad de utilizar múltiples controladores y también se utiliza para enviar información al controlador (ej. calidad de enlace). En esta arquitectura el control distribuido puede presentar un cuello de botella en la escalabilidad porque resulta complicado mantener la consistencia global de la red. Por ejemplo, si el descubrimiento de nodos en un segmento de la red es inconsistente o falla podría generar información de topología inconsistente al resto de la red, lo que provocaría serios problemas de rendimiento.

El mecanismo de aprendizaje por refuerzo de la SDWSN cognitiva [43] permite definir políticas de enrutamiento con balanceo de carga para mejorar la eficiencia energética y la adaptabilidad de la red, ante los cambios ambientales. Esta arquitectura cuenta con tres funcionalidades fundamentales: un módulo de configuración de QoS en el plano de aplicación, una capa intermedia de información cognitiva (CIM, del inglés C*ognitive Information Middleware*) en el plano de control y el módulo de procesamiento de información en el plano de datos. Los autores no especifican cómo se realiza la comunicación entre el plano de control y el de aplicación, ni cómo sería la comunicación entre múltiples controladores. Esta arquitectura está enfocada principalmente a la gestión de energía y a la QoS.

En [52] Olivier *et al.* aplican el modelo SDN a una WSN con topología clúster para formar una red de sensores en clúster definida por software (SDCSN, del inglés *Software Defined for Clustered Sensor Networks*). Esta red está organizada en clústeres, o dominios SDN, cada uno compuesto por nodos sensores, enrutadores y un cabeza de clúster definido por software (SDNCH, del inglés *SDN Cluster Head*). El SDNCH es una estación base que controla y coordina los nodos sensores de su dominio, puede implementar sus propias políticas de seguridad y, por lo tanto, proteger al dominio de ataques externos; ante caídas de un SDNCH, los nodos del dominio seleccionan uno nuevo. Cada SDNCH construye un mapa de la topología de su dominio que intercambia con otros a través de una interfaz WE-Bridge [54]. Este trabajo se centra principalmente en el diseño de un plano de control distribuido y tiene como desventaja que sólo puede utilizarse en WSN con topología clúster, apropiada principalmente para despliegues de media escala [55,56]. Además, esta arquitectura usa el protocolo *OpenFlow* estándar, demasiado pesado para los nodos sensores.

Han y Ren en [57] también proponen utilizar el protocolo *OpenFlow* nativo de SDN. Los autores consideran que una WSN basada en SDN está formada por tres tipos de nodos: central (cabeza del clúster), maestro (sumidero) y normal (nodo sensor). El nodo maestro actúa como controlador, el nodo central es similar a un conmutador *OpenFlow* por lo tanto, se encarga del reenvío de paquetes y el nodo normal es el dispositivo final que solo recibe datos. En esta estructura se comparte el control de la red utilizando *FlowVisor* [58], una aplicación de virtualización basada en SDN que permite a múltiples controladores gestionar un conmutador *OpenFlow* de forma simultánea. El nodo maestro determina la ruta al recibir una nueva solicitud y los nodos centrales mantienen las tablas de flujo. La red se organiza en clúster, con los nodos centrales como cabezas de grupo. Los autores reconocen que en esta propuesta se deben resolver algunos problemas como la limitación de energía en los nodos y la topología, entre otros. Además, en las redes SDN cableadas los dispositivos finales son considerados periféricos y están más allá del protocolo *OpenFlow*, justo lo contrario a las WSN donde los nodos sensores se comportan como dispositivos finales que generan datos.

El plano de control distribuido requiere una comunicación consistente y sincronizada entre los controladores. Para esto, es necesario la API "este/oeste" que permite la interacción entre los controladores. Sin embargo, al igual que la API "hacia el norte", hay poca estandarización al respecto [3].

La arquitectura SDWSN [14] de Jacobsson y Orfanidis emplea un modelo de control híbrido con un controlador central en el plano de control y un controlador local en cada nodo sensor. El propósito del controlador local es configurar, reconfigurar, monitorear y también ejecutar instrucciones desde el controlador. Los nodos sensores también están equipados con máquinas virtuales (VM) que ayudan a instalar nuevos protocolos o actualizaciones de código cuando sea necesario. En este artículo no se describe como se realiza la comunicación entre el controlador central y los locales, tampoco se argumenta como sería la comunicación entre el plano de control y el de aplicaciones. El problema principal de esta arquitectura es que las funcionalidades de control que se delegan al nodo sensor están en contradicción con la escasa capacidad de cómputo y limitación de energía de los nodos sensores.

SDN-WISE es una implementación práctica de WSN basada en SDN propuesta por Galluccio et al. en [19,36] que permite a los desarrolladores programar el controlador en el lenguaje de su preferencia. Los autores diseñan una arquitectura completa que limita el intercambio de información entre los nodos y el controlador e implementan un nodo sensor programable que puedan adoptar políticas adecuadas (definidas previamente por el controlador) para minimizar la sobrecarga de mensajes y el consumo de energía en la red. SDN-WISE presenta una capa de software que permite que varias redes virtuales se ejecuten en el mismo sensor inalámbrico físico o red WPAN, de manera similar a como hace *FlowVisor* en las redes *OpenFlow* y brinda las herramientas necesarias para probar el controlador en OMNeT++ [59], de forma similar a Mininet [60,61].

La arquitectura SDN-WISE está compuesta por nodos sensores y uno o más sumideros (*sink*). Los sumideros sirven, además, de pasarelas (*gateway*) entre los nodos sensores y los elementos que implementan el plano de control. En este plano, la lógica de gestión de la red se dicta por uno o varios controladores, uno de los cuales implementa la virtualización de red (*WISE-Visor*). *WISE-Visor* incluye una capa de gestión de topología (TM) que abstrae los recursos de la red para que diferentes redes lógicas, con diferentes políticas de administración establecidas por diferentes controladores, se pueda ejecutar en el mismo conjunto de dispositivos físicos. La capa TM, de SDN-WISE, tiene acceso a las API ofrecidas por todas las capas de protocolo para controlar su comportamiento y, por lo tanto, implementar operaciones de capa cruzada. Para este propósito, los autores implementan una capa de adaptación entre el sumidero y el *WISE-Visor* que se encarga de formatear los mensajes recibidos por el sumidero de manera tal que se puedan manejar por el *WISE-Visor* y viceversa. Los controladores se pueden implementar en el mismo nodo que aloja a *WISE-Visor* o en servidores remotos.

En SDN-WISE los autores definen sus propias tablas de flujo y protocolos. Como se limita el intercambio de información entre los nodos y el controlador, los nodos necesitan algunas funcionalidades para tomar decisiones sin contar con el controlador lo que aumenta su complejidad y el requerimiento de recursos.

3.3.- RESUMEN

En la Tabla 1 se resumen las principales arquitecturas de redes de sensores definidos por software. Se observa que estas propuestas se centran en la estructura y funciones de los nodos sensores y controlador(es) definidos por software y en la comunicación entre el plano de datos y el plano de control. Con excepción de la arquitectura SDN cognitiva [43], donde se exponen algunas ideas, en los trabajos consultados no hay referencias a las aplicaciones SDWSN; algo similar sucede con la interfaz "hacia el norte".

4.- CONCLUSIONES

En el futuro se espera que las redes de sensores inalámbricos definidas por software constituyan una excelente solución a las problemáticas de las WSN. Al separar los planos de control y datos, se espera que disminuya el consumo de energía en los nodos sensores, que mejore la gestión de la red, así como la seguridad y escalabilidad.

Los principales trabajos en esta tecnología SDN WSN se dedican a definir la estructura de los nodos sensores y controlador(es) definidos por software, y a la "hacia el sur". En pocos se tratan la interfaz "este/oeste" y la "hacia el norte. Esta última es muy importante por su conexión con la mayoría de las funcionalidades que se eliminan del nodo sensor.

Existe consenso en utilizar la filosofía *OpenFlow* en la interfaz "hacia el sur". En la arquitectura *Flow-Sensor* se utilizan este protocolo nativo SDN y en otras, como *SensorSDN*, SD-WSM y SDN cognitiva, una adaptación para soportar el procesamiento de datos en la red y el direccionamiento de las WSN. *OpenFlow* es un protocolo pesado para los nodos sensores y en la adaptación del formato TLV, usado para redefinir la identificación de los nodos sensores, introduce campos

de encabezado que son irrelevantes para WSN y que son analizados por los enrutadores WSN con buffers limitados. Conscientes de esas limitaciones, trabajos como SDN-WISE, TinySDN y otros definen sus propios protocolos para la interfaz "hacia el sur".

En la implementación de la lógica de control en las SDWSN se siguen varias variantes: en algunos se aloja en el sumidero y en otros en un nivel más alto. El control puede ser centralizado, distribuido e híbrido. En el primer caso, un fallo en el controlador central compromete la red, además la red se vuelve ineficiente a medida que crece. La escalabilidad es esencial en IoT donde las WSN juegan un papel importante. El control distribuido mejora estos aspectos, pero requiere una comunicación constante entre los controladores para la cual es importante definir la interfaz "este/oeste". El control híbrido permite que la red se reconfigure y autorepare ante fallos, pero a costa de recargar al nodo sensor.

De este estudio se puede concluir que las SDWSN están aún en su etapa inicial por lo que no existe un estándar. En trabajos futuros se deben considerar aspectos claves como la gestión de energía con la implementación de ciclos de trabajo, agregación de datos, movilidad de los nodos y capacidad de autoreparación. A pesar de las problemáticas a resolver, los beneficios de la implementación de SDN en las WSN superan los inconvenientes por lo que esta nueva tecnología se considera como un área de investigación prometedora.

Tabla 1
Áreas de trabajo en las principales arquitecturas de redes de sensores definidos por software

Arquitecturas	Nodo sensor definido por software	API "hacia el sur"	Controlador SDWSN	API "hacia el norte"	Aplicación SDWSN
Flow-Sensor [46]	Si	*OpenFlow*	Centralizado	No	No
SDWN [35]	Si	Propio	Centralizado	No	No
SD-WSN [33]	Si	*Sensor OpenFlow*	Centralizado	No	No
SDSN [48]	Si	No	Centralizado	No	No
SDWSN [14]	Si	No	Híbrido	No	No
SDWSN [47]	No	No	Centralizado	No	No
TinySDN [45]	Si	Propio	Distribuido	No	No
SDWSN cognitiva [43]	Si	*Sensor OpenFlow*	Distribuido	No	QoS
SDCSN [52]	No	No	Distribuido	No	No
SDN-WISE [19]	Si	Propio	Híbrido	No	No
SensorSDN[49]	Si	*OpenFlow*	Centralizado	No	No
Soft-WSN [13]	Si	No SDN	Centralizado	No	No

REFERENCIAS

1. Ndiaye M, Hancke G, Abu-Mahfouz A. Software Defined Networking for Improved Wireless Sensor Network Management: A Survey. Sensors. 2017;17(5):1031.
2. Ding J. Advances in network management. CRC Press; 2010.
3. Kobo HI, Abu-Mahfouz AM, Hancke GP. A Survey on Software-Defined Wireless Sensor Networks: Challenges and Design Requirements. IEEE Access. 2017;5:1872–1899.
4. Xia W, Wen Y, Foh CH, Niyato D, Xie H. A Survey on Software-Defined Networking. IEEE Commun Surveys & Tutorials. 2015;17(1):27–51.
5. Kreutz D, Ramos FM V., Esteves Verissimo P, Esteve Rothenberg C, Azodolmolky S, Uhlig S. Software-Defined Networking: A Comprehensive Survey. Proceedings of the IEEE. 2015;103(1):14–76.
6. Modieginyane K, Letswamotse B, Malekian R, Abu-Mahfouz A. Software defined wireless sensor networks application opportunities for efficient network management: A survey. Computers and Electrical Engineering. 2018;66:274-287

7. Sood K, Yu S, Xiang Y. Software-Defined Wireless Networking Opportunities and Challenges for Internet-of-Things: A Review. IEEE Internet of Things Journal. 2016;3(4):453–463.
8. Mostafaei H, Menth M. Software-defined wireless sensor networks: A survey. Journal of Network and Computer Applications. 2018;119:42-56
9. Núnez G, Margi C. Energy Map Model for Software-Defined Wireless Sensor Networks. XXXV Simpósio Brasileiro de Telecomunicações e Processamento de Sinais. São Pedro; Brasil; 2017.p. 826–830
10. Wang Y, Chen H, Wu X, Shu L. An energy-efficient SDN based sleep scheduling algorithm for WSNs. Journal of Network and Computer Applications. 2016;59:39–45.
11. Valdivieso Caraguay ÁL, Benito Peral A, Barona López LI, García Villalba LJ. SDN: Evolution and opportunities in the development IoT applications. International Journal of Distributed Sensor Networks. 2014;10(5):1896–1899.
12. Nguyen TMC, Hoang DB, Chaczko Z. Can SDN technology be transported to software-defined WSN/IoT?. IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData). Chengdu; China; 2016.
13. Bera S, Misra S, Roy SK, Obaidat MS. Soft-WSN: Software-Defined WSN Management System for IoT Applications. IEEE Systems Journal. 2018;12(3):2074–2081.
14. Jacobsson M, Orfanidis C. Using software-defined networking principles for wireless sensor networks. 11th Swedish National Computer Networking Workshop. Karlstad; Sweden; 2015. p. 1–5
15. Anadiotis ACG, Galluccio L, Milardo S, Morabito G, Palazzo S. Towards a software-defined Network Operating System for the IoT. In: IEEE 2nd World Forum on Internet of Things (WF-IoT). Milan; Italy; 2015. p. 579–584
16. Hu F, Hao Q, Bao K. A survey on software-defined network and OpenFlow: From concept to implementation. IEEE Communications Surveys & Tutorials. 2014;16(4): 2181–2206.
17. Huang R, Chu X, Zhang J, Hu YH. Scale-free topology optimization for software-defined wireless sensor networks: A cyber-physical system. International Journal of Distributed Sensor Networks. 2017;13(6):1–12.
18. Abdolmaleki N, Ahmadi M, Hadi Tabatabaee Malazi HT, Milardo S. Fuzzy topology discovery protocol for SDN-based wireless sensor networks. Simulation Modelling Practice and Theory. 2017;79:54–68.
19. Galluccio L, Milardo S, Morabito G, Palazzo S. SDN-WISE: Design, prototyping and experimentation of a stateful SDN solution for WIreless SEnsor networks. In: 2015 IEEE Conference on Computer Communications (INFOCOM) [Internet]. Kowloon; Hong Kong; 2015 p. 513–521.
20. IEEE 802.15.4-2011 - IEEE Standard for Local and metropolitan area networks--Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) [Internet]. [cited 2018 Jan 30]. Available from: http://standards.ieee.org/findstds/standard/802.15.4-2011.html
21. Hassan MA, Vien Q-T, Aiash M. Software Defined Networking for Wireless Sensor Networks: A Survey. Advances in Wireless Communications and Networks. 2017;3(2):10–22.
22. Sendra S, Parra L, Lloret J, Jiménez JM. Oceanographic multisensor buoy based on low cost sensors for posidonia meadows monitoring in mediterranean sea. Journal of Sensors. 2015;2015
23. Mshali H, Lemlouma T, Moloney M, Magoni D. A survey on health monitoring systems for health smart homes. International Journal of Industrial Ergonomics. 2018;1;66:26–56.
24. Bapat V, Kale P, Shinde V, Deshpande N, Shaligram A. WSN application for crop protection to divert animal intrusions in the agricultural land. Computers and Electronics in Agriculture. 2017;133(C):88–96.
25. Lazarescu MT, Lavagno L. Wireless sensor networks. In: Handbook of Hardware/Software Codesign. Springer, Dordrecht. 2017.
26. Correia FP, De Alencar MS, Lopes WTA, De Assis MS, Leal BG. Propagation analysis for wireless sensor networks applied to viticulture. International Journal of Antennas and Propagation. 2017;2017:1–10.
27. Cloete NA, Malekian R, Nair L. Design of Smart Sensors for Real-Time Water Quality Monitoring. IEEE Access. 2016;4:3975–3990.
28. Gangurde P, Bhende M. A Review on Precision agriculture using Wireless Sensor Networks. International Journal of Engineering Trends and Technology (IJETT). 2015;23(9):426–431.

29. Adamo F, Attivissimo F, Guarnieri Calo Carducci C, Lanzolla AML. A Smart Sensor Network for Sea Water Quality Monitoring. IEEE Sensors Journal. 2015;15(5):2514–2522.
30. Carrano RC, Passos D, Magalhaes LCS, Albuquerque CVN. Survey and Taxonomy of Duty Cycling Mechanisms in Wireless Sensor Networks. I IEEE Communications Surveys & Tutorials. 2014;16(1):181–194.
31. Dhand G, Tyagi SS. Data Aggregation Techniques in WSN:Survey. Procedia Computer Science. 2016;92(2016):378–384.
32. Open Networking Foundation. OpenFlow Switch Specification. Version 1.5.0. 2014.
33. Luo T, Tan HP, Quek TQS. Sensor openflow: Enabling software-defined wireless sensor networks. IEEE Communications Letters. 2012;16(11):1896–1899.
34. Karapistoli E, Pavlidou FN, Gragopoulos I, Tsetsinas I. An overview of the IEEE 802.15.4a Standard. IEEE Communications Magazine. 2010;48(1):47–53.
35. Costanzo S, Galluccio L, Morabito G, Palazzo S. Software Defined Wireless Networks (SDWNs): Unbridling SDNs. In: 2012 European Workshop on Software Defined Networking (EWSDN). Darmstadt; Germany; 2012. p. 1–6.
36. Galluccio L, Milardo S, Morabito G, Palazzo S. Reprogramming Wireless Sensor Networks by using SDN-WISE: A hands-on demo. In: IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS). Hong Kong; China; 2015. p. 19-20.
37. Mazzer Y, Tourancheau B. Comparisons of 6LoWPAN Implementations on Wireless Sensor Networks. In: 2009 Third International Conference on Sensor Technologies and Applications. IEEE; 2009 [cited 2018 May 28]. p. 689–692. Available from: http://ieeexplore.ieee.org/document/5210833/
38. Mudumbe MJ, Abu-Mahfouz AM. Smart water meter system for user-centric consumption measurement. In: 2015 IEEE 13th International Conference on Industrial Informatics (INDIN). Athens; Glyfada; Greece; 2015. p. 993–998.
39. Catapang SA, Roberts ZJM, Wang KI-K, Salcic Z. An infrastructure for integrating heterogeneous embedded 6LoWPAN networks for Internet of Things applications. In: 2013 Seventh International Conference on Sensing Technology (ICST). Wellington; New Zealand; 2013. p. 741–746.
40. Dunkels A, Gronvall B, Voigt T. Contiki - a lightweight and flexible operating system for tiny networked sensors. In: 29th Annual IEEE International Conference on Local Computer Networks. Tampa; FL; USA; 2004. p. 455–462.
41. Hill J, Szewczyk R, Woo A, Hollar S, Culler D, Pister K, et al. System architecture directions for networked sensors. ACM Sigplan notices. 2000;35(11):93–104.
42. Levis P, Madden S, Polastre J, Szewczyk R, Whitehouse K, Woo A, et al. TinyOS: An Operating System for Sensor Networks. In: Ambient Intelligence. Springer; Berlin; Heidelberg;2005. p. 115–48.
43. Huang R, Chu X, Zhang J, Hu YH. Energy-Efficient Monitoring in Software Defined Wireless Sensor Networks Using Reinforcement Learning: A Prototype. International Journal of Distributed Sensor Networks. 2015;11(10): 360428.
44. Trevizan de Oliveira B, Batista Gabriel L, Borges Margi C. TinySDN: Enabling Multiple Controllers for Software-Defined Wireless Sensor Networks. IEEE Latin America Transactions. 2015;13(11):3690–3696.
45. De Oliveira BT, Margi CB. TinySDN: Enabling TinyOS to Software-Defined Wireless Sensor Networks. XXXIV Simpósio Brasileiro de Redes de Computadores. Bahia; Brasil; 2016. p. 1229–1237.
46. Mahmud A, Rahmani R. Exploitation of OpenFlow in wireless sensor networks. In: Proceedings of 2011 International Conference on Computer Science and Network Technology. Harbin; China; 2011. p. 594–600.
47. De Gante A, Aslan M, Matrawy A. Smart wireless sensor network management based on software-defined networking. In: 2014 27th Biennial Symposium on Communications (QBSC). Kingston; ON; Canada; 2014. p. 71–5.
48. Zeng D, Miyazaki T, Guo S, Tsukahara T, Kitamichi J, Hayashi T. Evolution of Software-Defined Sensor Networks. In: 2013 IEEE 9th International Conference on Mobile Ad-hoc and Sensor Networks. Dalian; China; 2013. p. 410–3.
49. Hakiri A, Gokhale A. Rethinking the design of LR-WPAN IoT systems with software-defined networking. In: Proceedings - 12th Annual International Conference on Distributed Computing in Sensor Systems, DCOSS 2016. 2016.
50. Rodrigues JJPC, Neves PACS. A survey on IP-based wireless sensor network solutions. Int J Commun Syst [Internet]. 2010 [cited 2018 May 24];23(8):n/a-n/a. Available from: http://doi.wiley.com/10.1002/dac.1099
51. Budhwar P. A Survey of Transport Layer Protocols for Wireless Sensor Networks. JETIR1504026 J Emerg Technol Innov Res [Internet]. 2015 [cited 2018 May 24];2. Available from: www.jetir.org

52. Olivier F, Carlos G, Florent N. SDN Based Architecture for Clustered WSN. In: 2015 9th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing [Internet]. IEEE; 2015 [cited 2018 Jan 25]. p. 342–7. Available from: http://ieeexplore.ieee.org/document/7284972/
53. Gnawali O, Fonseca R, Jamieson K, Moss D, Levis P. Collection tree protocol. In: Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems - SenSys '09 [Internet]. New York, New York, USA: ACM Press; 2009 [cited 2018 Sep 6]. p. 1. Available from: http://portal.acm.org/citation.cfm?doid=1644038.1644040
54. Lin P, Bi J, Chen Z, Wang Y, Hu H, Xu A. WE-bridge: West-east bridge for SDN inter-domain network peering. In: 2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS) [Internet]. IEEE; 2014 [cited 2018 Sep 6]. p. 111–2. Available from: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6849180
55. Kaur S, Scholar MT, Faridkot A, Singh H, Singh PG. Examine the Performance of different Topologies using Opnet 14.5 in ZigBee Sensor Network. Int J Comput Appl. 2014;108(7):975–8887.
56. A Study of ZigBee Network Topologies for Wireless Sensor Network with One Coordinator and Multiple Coordinators. Tikrit J Eng Sci. 2012;19(4):65–81.
57. Han Z-J, Ren W. A Novel Wireless Sensor Networks Structure Based on the SDN. Int J Distrib Sens Networks. 2014;10(3):1–8.
58. Sherwood R, Gibb G, Yap K-K, Appenzeller G, Casado M, Mckeown N, et al. FlowVisor: A Network Virtualization Layer [Internet]. 2009 [cited 2019 Jun 20]. Available from: http://openflowswitch.org/downloads/technicalreports/openflow-tr-2009-1-flowvisor.pdf
59. OMNeT++ Discrete Event Simulator - Home [Internet]. [cited 2018 Sep 6]. Available from: https://www.omnetpp.org/
60. Kaur K, Singh J, Singh Ghumman N. Mininet as Software Defined Networking Testing Platform [Internet]. [cited 2018 Sep 6]. Available from: https://pdfs.semanticscholar.org/b1c7/f8ac477a5553303802bb7785dd3b53372057.pdf
61. Wang S-Y. Comparison of SDN OpenFlow network simulator and emulators: EstiNet vs. Mininet. In: 2014 IEEE Symposium on Computers and Communications (ISCC) [Internet]. IEEE; 2014 [cited 2018 Sep 6]. p. 1–6. Available from: http://ieeexplore.ieee.org/document/6912609/

CONFLICTO DE INTERESES

No existe conflicto de intereses entre los autores, aunque dos pertenecen a la Universidad de Oriente y otro a la Universidad Tecnológica de La Habana José Antonio Echeverría. Tampoco existe conflicto de intereses entre los autores y las instituciones a las que están afiliados, ni con ninguna otra institución.

CONTRIBUCIONES DE LOS AUTORES

- **Lidice Romero Amondaray**: conceptualización, preparación, creación y desarrollo del artículo.
- **Fernando José Artigas Fuentes**: revisión crítica de cada una de las versiones del borrador del artículo y aprobación de la versión final a publicar.
- **Caridad Anías Calderón**: contribución a la conceptualización y organización del artículo, sugerencias acertadas para la conformación de la versión final.

Todos los autores contribuyeron con las ideas que se plasman en el artículo.

AUTORES

Lidice Romero Amondaray, Master en Sistemas de Telecomunicaciones y Profesora Auxiliar de la Universidad de Oriente, Santiago de Cuba, Cuba. E-mail: lidice@uo.edu.cu. ORCID: https://orcid.org/0000-0001-5573-3170. Sus investigaciones se centran en las redes de sensores inalámbricos.

Fernando José Artigas Fuentes, Doctor en Ciencias y Profesor Titular de la Universidad de Oriente, Santiago de Cuba, Cuba. E-mail: artigas@uo.edu.cu. ORCID: https://orcid.org/0000-0003-4977-2135. Sus investigaciones se centran en la computación paralela y reconocimiento de patrones.

Caridad Anías Calderón, Doctora en Ciencias y Profesora Titular de la Universidad Tecnológica de La Habana José Antonio Echeverría (CUJAE), La Habana, Cuba. E-mail: cacha@tesla.cujae.edu.cu. ORCID: https://orcid.org/0000-0002-5781-6938. Sus investigaciones se centran en las redes de datos y en particular en la gestión de redes y servicios.