

Consideraciones para el Control de congestión en redes inalámbricas de sensores utilizando la optimización crosslayer

Raymundo Buenrostro-Mariscal, Juan Iván Nieto-Hipólito, María Cosío-León, Mabel Vázquez-Briseno, and Juan de Dios Sánchez-López

Facultad de Ingeniería, Arquitectura y Diseño, UABC
Carr. Tijuana-Ensenada Km. 103, 22860 Ensenada, BC., México
{rbuenrostro, jnieto, maria.cosio, mabel.vazquez, jddios}@uabc.edu.mx
<http://telematica.uabc.mx>

Paper received on 03/10/12, Accepted on 24/10/12.

Resumen La utilización de las redes inalámbricas de sensores (WSN) ha evolucionado rápidamente. Hoy se proponen para la recolección de información de naturaleza crítica. Estas aplicaciones tienen requerimientos muy específicos en la transmisión de datos, como la tasa de paquetes perdidos y el retardo. La congestión en la red es un problema que compromete el logro de los requerimientos; por ello, requiere una atención especial. Existen mecanismos para controlar la congestión, como los protocolos de transporte, que incluyen funciones para detectarla, notificarla y resolverla. En el estado del arte para el control de congestión en WSN, las propuestas son diseñadas para aplicaciones con requerimientos más simples y de forma independiente de otras capas involucradas en la comunicación (Red y Enlace de Datos). En este trabajo se presentan las consideraciones y requerimientos para un protocolo de Control de Congestión crosslayer consciente de las aplicaciones, la subcapa MAC y la capa Física.

Keywords: Congestión, Crosslayer, QoS, WSN

1. Introducción

Las Redes Inalámbricas de Sensores (WSN, del inglés Wireless Sensor Networks) son redes compuestas por dispositivos interconectados entre sí, que trabajan de forma cooperativa para recolectar datos de interés y transmitirlos de forma inalámbrica a otro punto. En los últimos años, se ha incrementado los trabajos que exploran su uso para aplicaciones críticas, como la supervisión de la salud de pacientes de forma remota [1], [2]. Estas aplicaciones son consideradas como heterogéneas, ya que generan distintos tipos y volúmenes de tráfico; que a su vez tienen diferentes requerimientos de transmisión en términos de velocidad de datos, confiabilidad y prioridad en la entrega. Las WSN tienen limitaciones importantes para cumplir con estos requerimientos [3], debido a su naturaleza inalámbrica y a su medio compartido, donde todos los nodos contendrán por el acceso al medio. Además, la mayor parte del tiempo la red mantiene flujos de datos desde cada nodo sensor hacia el nodo final (Sink), tráfico muchos-a-uno

(*many-to-one*); provocando que los nodos intermedios manejen una mayor cantidad de tráfico y se saturen rápidamente. Este problema se conoce como *efecto embudo o funneling* [4]. Aunado a lo anterior, los dispositivos que integran la red tienen capacidades limitadas de hardware, especialmente de procesamiento, capacidad de almacenamiento y suministro de energía; que incrementan los retos de la transmisión. Bajo este contexto, la *congestión* es un problema que puede presentarse frecuentemente tanto en los nodos como en el propio enlace inalámbrico. La congestión tiene un impacto directo en el desempeño global de la red y en la calidad de servicio (QoS, del inglés Quality of Service) entregada a las aplicaciones; ya que incrementa la cantidad de paquetes perdidos, retrasa la entrega de datos y aumenta el gasto de energía [5]. Por lo tanto, solucionar la congestión en las redes WSN es un reto obligado para cumplir con el objetivo primario de la recolección remota de datos. Las soluciones deben incluir tres funciones principales: *detección de congestión*, *notificación* y *resolución*. Las propuestas existentes para las WSN implementan las funciones de forma distinta y con objetivos diferentes, pero en su mayoría están diseñadas para aplicaciones homogéneas con requerimientos más sencillos y trabajando de forma independiente de otros mecanismos claves en la transmisión de datos. De lo anterior, crear un protocolo de control de congestión para aplicaciones críticas y heterogéneas es necesario. Que además, optimice el uso de los recursos y alcance el mayor desempeño. El diseño *crosslayer* se propone como una de las mejores alternativas para este escenario, ya que obtiene mejores rendimientos que las soluciones tradicionales que trabajan de forma independiente [6]. El diseño *crosslayer* propone aprovechar las funciones de los protocolos de diferentes capas, por ejemplo transporte, red y subcapa MAC (del inglés Medium Access Control) para trabajar en conjunto. En las siguientes secciones se presentan las principales consideraciones para el Control de Congestión en WSN, para guiar el diseño de nuevos protocolos, lo cual es la principal contribución de este trabajo.

2. Clasificación de las principales líneas de diseño para el control de congestión

A partir de los trabajos consultados se realiza una propuesta de clasificación de las soluciones existentes para el control de congestión; específicamente los mecanismos utilizados para la detección, notificación y resolución de congestión. El cuadro 1 resume esta clasificación, resalta la preferencia por la notificación implícita y el ajuste de la tasa de transmisión en los nodos, como mecanismos para mitigar la congestión. De igual forma se aprecia que sólo tres propuestas utilizan el diseño *crosslayer*. A continuación se presenta el trabajo relacionado, dividido por las funciones propuestas del control de congestión.

2.1. Detección de congestión

STCP [7] utiliza el grado de utilización del búfer en cada nodo intermedio involucrado en una transmisión de datos, para detectar congestión de forma local; mientras que SenTCP [8] utiliza la relación del tiempo promedio de arribo de paquetes contra

Cuadro 1. Clasificación de los protocolos analizados

Propuesta	Detección	Notificación	Resolución	Diseño
STCP	Nivel de búfer	Implícita	Ajusta velocidad Re-dirige tráfico	Tradicional
SenTCP	Nivel de búfer T. servicio p. T. inter-arribo p.	Implícita	Ajusta velocidad	Tradicional
TRCCIT	Tasa de datos	Implícita	Ajusta velocidad Re-dirige tráfico	Tradicional
CODA	Carga del canal Nivel de búfer	Explícita	Descarta paquetes Ajusta velocidad	Tradicional
CTCP	Tasa de error T. Nivel de búfer	Explícita	Ajusta velocidad	Tradicional
RT^2	Retardo del nodo Nivel de búfer	Implícita	Ajusta velocidad	Crosslayer
LCART	Carga de canal T. servicio p. T. inter-arribo p. Nivel de búfer	Implícita	Ajusta velocidad	Crosslayer
PCCP	T. inter-arrib p. T. servicio p.	Implícita	Ajusta velocidad	Crosslayer
CRRT	Nivel de búfer Tasa de datos	Implícita	Ajusta velocidad	Tradicional

el tiempo promedio de servicio de paquetes en combinación con la cantidad empleada del búfer para detectar el grado de congestión local en cada nodo. De forma similar TRCCIT [9] utiliza la tasa de datos, donde compara la tasa de arribos de paquetes contra la tasa de envío de paquetes, para declarar congestión en el nodo. TRCCIT asume un control de congestión pro-activo, al monitorear la carga local de la red y actuar en consecuencia. En CODA [10], además del nivel de ocupación del búfer, se utiliza una combinación de las condiciones de carga actual y pasada del canal inalámbrico para inferir una detección más precisa de congestión en cada receptor. CTCP [11] determina la presencia de congestión cuando se rebasa un umbral de búfer y una tasa de error de bit o pérdida de paquetes, mediante el mecanismo de reconocimiento de paquetes (Ack, del inglés Acknowledgment) en modo *salto-por-salto* (HbH, del inglés Hop-by-Hop); CTCP debido a éstos mecanismos, puede distinguir si la causa de congestión es por desbordamiento de búfer o por errores de transmisión. RT^2 [12] define un umbral de retardo promedio del nodo, con el cual, el nodo tiene una idea acerca del nivel de contención o saturación del medio en su entorno; y lo combina con un nivel de búfer para detectar de forma precisa la congestión. PCCP [13] utiliza el tiempo promedio de inter-arribo de paquetes t_a^i y el tiempo promedio de servicio de paquete t_s^i en la subcapa MAC para determinar el *grado de congestión* $d(i)$ actual en el nodo o del enlace; este nuevo índice de congestión es definido como la proporción del tiempo de servicio sobre

el tiempo de arribo de paquetes $d(i) = t_s^i/t_a^i$; entonces cuando $d(i) > 1$ asume que el nodo experimenta congestión e informa a los nodos vecinos. PCCP nombra a ésta forma de detección como *Detección de Congestión Inteligente*. LCART [14] resuelve la detección de congestión de la red mediante el uso simultáneo de varios eventos como el t_s , t_a , nivel de uso del búfer y un umbral límite de carga del canal.

2.2. Notificación de congestión

La mayoría de las propuestas de notificación de congestión utiliza el concepto de piggyback para enviar de forma implícita el aviso de congestión a los nodos de la red. STCP [7] incorpora esta forma de notificación y lo realiza dentro del paquete de datos y en el paquete de reconocimiento Ack, fijando un bit en el campo de *notificación de congestión* (CN, del inglés Congestion Notification) del encabezado del paquete. En STCP cada nodo de la red tiene la posibilidad de detectar la congestión y generar la notificación hacia el Sink para informarle del estado de la red, a su vez el Sink genera el aviso de congestión utilizando el paquete Ack hacia los nodos fuente. TRCCIT utiliza la notificación implícita HbH, pero a diferencia de STCP, éste avisa de la congestión local de forma inmediata a los nodos vecinos, mediante la activación de un bit en cada paquete que sale del nodo. PCCP utiliza el mismo tipo de notificación implícita que STCP y TRCCIT pero no sólo cambia el estado de un bit, si no que agrega información útil al mensaje de notificación: nivel de prioridad global, t_s^i , t_a^i y cantidad de nodos hijos; para que los nodos que reciben el mensaje realicen acciones diferenciadas en base a la información. Además, PCCP tiene dos eventos que pueden provocar el envío del mensaje de notificación, la primera ocurre cuando el número de paquetes reenviados por el nodo rebasa un umbral predefinido y la segunda cuando el nodo recibe el mensaje de notificación de congestión enviado por otros nodos. CODA utiliza notificación explícita para difundir el aviso de congestión hacia los nodos fuente en un modo de comunicación HbH; el nodo al recibir este mensaje reduce la tasa de envío y retransmite el aviso de congestión a los nodos vecinos, esta forma es conocida como *Backpressure*. CTCP utiliza la notificación explícita mediante la cual, el nodo que detecta la congestión genera un mensaje a todos los nodos de su vecindario, para indicarles que no puede recibir más paquetes. De la misma forma, cuando se resuelve la congestión, los nodos son informados para re-establecer las tasas de transmisión.

2.3. Resolución de congestión

CRRT [15] implementa un mecanismo centralizado en el Sink para eliminar la congestión de la red en modo *extremo-a-extremo* (E2E, del inglés End-to-End); esta forma contribuye a una decisión imparcial ya que el Sink tiene una visión global de la red y puede controlar la velocidad de todos los nodos en la red. El mecanismo base para resolver la congestión en CRRT es el esquema "Incremento Aditivo y Decremento Aditivo" (AIAD, del inglés Additive Increase Additive Decrease), el cual ajusta gradualmente la tasa de transmisión en los nodos fuente evitando una reducción agresiva. ART [16] utiliza un esquema distribuido para manejar la congestión, en los nodos marcados como

esenciales *E-Nodes*, con el cual ajustan la velocidad de transmisión en la red; sin embargo, sólo aplican este ajuste a un conjunto de nodos etiquetados como *no esenciales* para detener temporalmente su tráfico. Los nodos no esenciales podrán reactivar el envío de datos hasta que reciban de la red el aviso de operación normal. La clasificación de los nodos es realizada por un algoritmo que se basa en la energía residual de cada nodo en cada ejecución. En STCP cuando un nodo recibe la notificación de congestión, éste puede en-rutar los paquetes sucesivos del flujo de datos por una ruta diferente, siempre y cuando se tenga un algoritmo en la capa de red que permita este proceso, o disminuir la velocidad de transmisión de los nodos fuente. CODA utiliza dos mecanismos para resolver la congestión: un esquema de lazo abierto HbH y un esquema de lazo cerrado E2E. En el primer esquema los nodos utilizan el método Backpressure para propagar el aviso e indicarles a los nodos vecinos reducir la tasa de envío, esquema conocido como auto-regulación de nodo fuente. En el esquema de lazo cerrado, el que controla la tasa de transmisión de los nodos fuente es el Sink, que emite un mensaje de regulación hacia todos los nodos cuando existe congestión persistente, esquema llamado regulación múlti-fuente. CODA decide que esquema utilizar de acuerdo al grado de congestión de la red, si la tasa de eventos de los nodos fuente está por debajo de cierta fracción de la capacidad teórica del canal utiliza auto-regulación de nodo, de lo contrario se utiliza regulación desde el Sink. PCCP implementa un control de congestión HbH de acuerdo al *grado de congestión $d(i)$* y a los *índices de prioridad del nodo*; con lo cual garantiza que el nodo que tiene un mayor índice de prioridad obtenga mas ancho de banda de la red. Cada nodo cuenta con dos "colas", una para el tráfico propio y otra para el tráfico en tránsito que recibe; entonces, de acuerdo a la prioridad definida para cada tráfico el nodo ajusta y programa el envío de los datos. Bajo esta base, PCCP ofrece una resolución de congestión flexible, diferenciada y distribuida para ofrecer *equidad ponderada* a los nodos de la red. TRCCIT resuelve la congestión de la red de forma pro-activa mediante la selección de varias rutas para redirigir el tráfico de los nodos fuente. La definición de congestión pro-activa se refiere a la capacidad de seleccionar sobre la marcha una ruta múltiple o individual para controlar la congestión. TRCCIT evita la congestión en la red, esto lo fundamenta al resolver rápidamente la *congestión transitoria* en los nodos. Entonces, en la medida que un nodo se adapta o resuelve la congestión transitoria evita el incremento y una condición prolongada de congestión, en una trayectoria dada.

3. Control de congestión en WSN

Existen principalmente dos causas que provocan *congestión* en las WSN. La primera se presenta a nivel de nodo, cuando la velocidad de arribo de paquetes excede la tasa de servicio de paquetes. Esto es, el tiempo para procesar los paquetes que se reciben en un nodo es mayor a los tiempos de inter-arribo de paquetes; ocasionando que el búfer del nodo se sature y *tire* los nuevos paquetes que arriban. Esta pérdida se clasifica como un *paquete descartado* o *packet dropped*. Los paquetes descartados provocan un gran número de retransmisiones en la red, incrementando considerablemente el retraso en la entrega de los paquetes y el consumo de energía [17]. Considerando la dirección del flujo de datos en las WSN, tráfico many-to-one, se espera que este problema se agudice en los nodos intermedios cercanos al Sink; ya que ellos manejan un mayor tráfico com-

binado de otros nodos [18]. La segunda causa de congestión es responsabilidad de los problemas del enlace inalámbrico, donde altas tasas o ráfagas de datos pueden exceder la capacidad límite del enlace y provocar congestión. Además, problemas relacionados con el método de acceso por contienda, la interferencia de las señales y las colisiones son otra causa a *nivel enlace* (subcapa MAC y capa Física). Los paquetes dañados por estas causas son considerados como *paquetes perdidos o packet loss*.

Los paquetes descartados y perdidos afectan fuertemente la *confiabilidad de la red* (capacidad de la red para transmitir paquetes con éxito), la *utilización del enlace* inalámbrico y el consumo de energía del nodo; ya que independientemente de la causa es necesario retransmitir los paquetes. Bajo este contexto, la congestión impacta negativamente el rendimiento global de la WSN y el cumplimiento de QoS de las aplicaciones, de ahí la importancia de incorporar mecanismos que controlen sus efectos o impidan que se presente.

3.1. Modelo funcional propuesto para el protocolo de control de congestión

Un protocolo de control de congestión bajo nuestra perspectiva debe incluir tres módulos funcionales: *detección de congestión*, *módulo de notificación* y *resolución de congestión*. La figura 1 muestra los tres módulos funcionales y propone los mecanismos que pueden ser utilizados para lograr el control de congestión en las WSN. El protocolo de control de congestión debería ser diseñado para trabajar, tanto en modo *salto-por-salto*, para ejecutar los procesos entre nodos vecinos a un salto; como en modo *extremo-a-extremo*, que considera la ejecución de los procesos entre nodos finales (nodo sensor y Sink). Con la capacidad para seleccionar el modo de acuerdo a los procesos que realice; con la intención de optimizar los tiempos de respuesta y el consumo de recursos de la red. Además, es importante que opere en ambas direcciones de la red; privilegiando el flujo en dirección hacia el Sink que tiene mayor presencia en la red.

El módulo de *detección* identifica la presencia de congestión mediante eventos definidos en los nodos de la red, principalmente en los nodos involucrados en una transmisión de datos. Existen mecanismos que pueden utilizar diferentes eventos para alcanzar una detección eficiente; por ejemplo, revisar el nivel de uso de los búfer de cada nodo, monitorear el tiempo de arribo de paquetes contra el tiempo de servicio de paquetes, el número de paquetes retransmitidos, entre otros.

El módulo de *notificación* debe difundir el aviso de congestión, desde el nodo que la detecta, a los nodos de la red o idealmente a los nodos involucrados en la congestión. Esta notificación puede ser *explícita* o *implícita*, en el primer caso se debe crear un paquete de control diseñado exclusivamente para este fin. En la notificación implícita el nodo utiliza el paquete de datos para incluir la información de control necesaria para informar a los nodos de la presencia de congestión; esta forma de empaquetado se conoce como "Piggybacks". La información de control que se incluya, en cualquiera de los dos casos, puede ser tan solo un cambio de bit de algún campo de control o información adicional para indicar la velocidad de transmisión permitida, nivel de prioridad o grado de congestión.

El módulo de *resolución de congestión* debe utilizar los medios posibles para eliminar la congestión o disminuirla; el principal mecanismo, y el más utilizado, es ajustar la velocidad de transmisión de paquetes en los nodos de la red a un valor límite, llamado

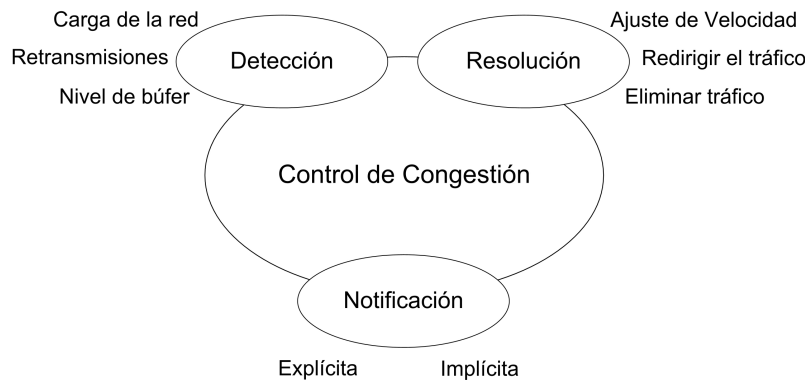


Figura 1. Modelo funcional del control de congestión para WSN

control de flujo de datos. Este mecanismo funciona para los problemas de desbordamiento de búfer o retardo de procesamiento del nodo. Sin embargo, en la pérdida de paquetes debido a los errores en la transmisión, una mejor alternativa es re-dirigir el tráfico a otro canal para reducir la congestión presente; si la topología y los protocolos de comunicación lo permiten.

Por lo general los tres módulos se consideran procesos exclusivos de la capa de transporte; sin embargo, es necesario un diseño crosslayer, con el cual la información y las capacidades de cada capa se comparten para el logro del objetivo del control de congestión.

4. Consideraciones de diseño del modelo funcional propuesto para el control de congestión

En esta sección se exponen las *consideraciones de diseño* claves para un protocolo de control de congestión crosslayer en WSN. Explicando en una primera fase aquellas que impactan en su operación de forma global. Para después introducir las consideraciones relevantes a cada uno de los módulos funcionales de la Figura 1; concluyendo con la heurística que guiara el diseño del protocolo crosslayer de control de congestión en WSN:

1. El *consumo de energía* tiene que estar presente en todas las etapas y procesos del control de congestión, con el fin de extender la vida útil de la red. Por lo tanto, es necesario reducir la pérdida de paquetes y mantener al mínimo la carga de tráfico de paquetes de control.
2. Habilidad para decidir cuándo operar en modo *HbH* y/o *E2E* para agilizar el tiempo de respuesta de los procesos de detección, notificación y resolución.
3. *Control de congestión dinámico* que le permita adaptarse a las prioridades de la aplicación y al estado actual de la red; a través del cual pueda asignar diferentes tasas de transmisión a los diferentes niveles de prioridad.

4. Ofrecer *equidad ponderada* a los nodos de la red en el uso del canal de comunicaciones, que se adapta a las prioridades del tráfico o del nodo.
5. Centrar el diseño en *evitar congestión*, más que pensar en mitigarla; lo cual necesita de mecanismos rápidos y efectivos de detección y notificación.
6. Con la intención de optimizar la notificación de congestión es necesario que el mecanismo de detección distinga casos de *congestión transitoria*; que pueden ser resueltos de forma local con el objetivo de evitar que la congestión evolucione a niveles más severos.
7. Es importante retroalimentarse de los parámetros de confiabilidad, tales como porcentaje de paquetes perdidos, retardo en la entrega de los paquetes, tasa de error de bit del canal, entre otros; para contar con mayor información que optimice el proceso de evaluación del módulo de detección. Lograr la interacción con el canal de comunicaciones es fundamental para este fin.
8. Diferenciar el origen de la pérdida del paquete, por desbordamiento del búfer o por problemas del enlace inalámbrico, es vital para las decisiones del módulo de resolución de congestión.

El conjunto de estas consideraciones involucran diferentes capas del modelo *OSI* (del inglés Open System Interconnection) para su desarrollo, lo cual requiere del paradigma de diseño crosslayer. En la literaturas encontramos propuestas que incluyen al menos dos capas [14] o algunas tan extensas como en [6].

4.1. Consideraciones para el módulo de detección de congestión

Se propone utilizar el esquema HbH para la detección de congestión por su rapidez y por la posibilidad de identificar el sitio de congestión dentro de la red. Sólo hay que tener en cuenta, en el diseño, que incrementa los procesos en cada nodo y en algunos casos el uso del búfer. Las consideraciones más importantes para este módulo son:

- Medir constantemente el *nivel del búfer* en los nodo. Es un método sencillo y rápido que se implementa tanto en modo HbH como E2E. Sin embargo, esta medida por si sola no es efectiva, ya que sería difícil establecer con precisión el grado de congestión. Ya que puede estar con una nivel bajo del nivel del búfer, mientras experimenta congestión por el trafico de otros nodos cercanos a él.
- Adicionar al nodo la habilidad para conocer la *carga de tráfico actual* del canal. Aprovechando la naturaleza compartida del medio inalámbrico, un nodo puede escuchar el canal para medir la actividad o el tráfico local de la red. Sin embargo, activar el dispositivo de radio del nodo es costoso, en términos de energía. Por lo cual, debe hacerse eficientemente, por ejemplo, a intervalos regulares y en cortos periodos de tiempo.
- Medir el *tiempo de arribo de paquetes* y el *tiempo de servicio de paquetes* en cada nodo; con lo cual no sólo mejora las medidas del grado de congestión del nodo y la carga local de la red; si no que además, puede disminuir la frecuencia de escucha del canal y ahorrar energía.

4.2. Consideraciones para el módulo de notificación de congestión

La notificación debe ser enviada lo más rápido posible para que el mecanismo de resolución de congestión inicie su proceso y elimine la congestión de la red. Se propone el método implícito como base para el mecanismo de notificación de congestión; que incluya como información de control: niveles de prioridad para nodos y tráfico, tasa máxima de transmisión y grado de congestión. Con la restricción de ocupar el menor espacio posible del paquete de datos:

- El *método explícito* genera un paquete especial de control. Sus ventajas son, el uso del espacio total del paquete, para incluir la información de control, y que puede ser enviado en cualquier momento. La principal desventaja es el aumento del tráfico en la red, contribuyendo a la congestión, al consumo de energía del nodo y al deterioro global de la utilización del enlace.
- El *método implícito* utiliza el concepto de piggyback; con lo cual resuelve el problema anterior. Sus desventajas son: el uso del espacio útil del paquete de datos, para incorporar los datos de control; y la espera para notificar la congestión hasta que se genere un paquete de datos.
- Por último, un evento importante del mecanismo de notificación, es generar un mensaje de *aviso de congestión terminada* a los nodos de la red, con la intención de recuperar la operación normal de la red lo más pronto posible.

4.3. Consideraciones para el módulo de resolución de congestión

La resolución de congestión debe ser dinámica, para fijar diferentes tasas de transmisión a los nodos y/o flujos de datos, de acuerdo a los requerimientos de la aplicación y a los recursos del nodo. Para evitar que la utilización del canal se degrade y se incumpla con los requisitos de QoS:

- El *ajuste de velocidad* puede ser E2E centralizado en el Sink o distribuido en cada nodo en modo HbH. El primero ofrece un ajuste imparcial y global a todos los nodos. Sin embargo, el esquema distribuido puede resolver la congestión en menor tiempo, a costa de mayor complejidad. Se propone éste último, ya que mitigar rápidamente la congestión es el objetivo principal.
- *Diferenciar el tráfico del nodo*, en tráfico propio y de tránsito, ayuda aplicar políticas de reducción diferencial de la tasa de transmisión, dándole mayor prioridad a sus datos o incluso el descarte del tráfico de tránsito.
- *Re-dirigir el tráfico* en rutas alternas cuando se tiene una ruta saturada es una opción. Sin embargo, se incrementa la complejidad de la capa de ruteo y la necesidad de vigilar el orden de los paquetes que llegan al Sink.

4.4. Heurística del protocolo crosslayer para el control de congestión

Derivado del análisis anterior y a las necesidades de los nuevos entornos de aplicación de las WSN proponemos una heurística para un protocolo crosslayer de control

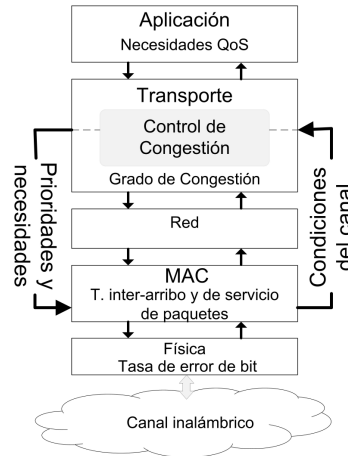


Figura 2. Esquema de la heurística del protocolo crosslayer para control de congestión

de congestión pro-activo, que ajuste la tasa de transmisión de forma dinámica, diferenciada y exacta; para indicar a los nodos que tanto aumentar o disminuir sus tasas de transmisión en relación al grado de congestión experimentado, a la ubicación del nodo dentro de la red y al tipo de tráfico de la aplicación. En nuestra propuesta la definición de optimización crosslayer se entiende como: la capacidad del protocolo de control de congestión para acceder a la información de parámetros de dos o más capas. Información que puede utilizar o modificar con el objetivo de controlar la congestión (Figura 2). Se propone utilizar la subcapa MAC y las capas Física y Transporte para la operación de los diferentes procesos. La subcapa MAC y la capa física proporcionan las condiciones actuales del canal y del comportamiento del tráfico para estimar el grado de congestión mediante los parámetros: *tasa de error de bit*, *el tiempo de servicio y de inter-arribo de paquetes*. En la capa de transporte se ubica la base del control de congestión, que proporciona información a la MAC para informarle de las *necesidades de operación* del protocolo y las *prioridades de las aplicaciones* soportadas en la capa de aplicación. La Figura 2 muestra la interacción de las capas, la información y parámetros que comparten. La heurística propone dos estrategias: *ajustar las tasas de transmisión* de cada nodo, de acuerdo al grado de congestión presente en el nodo y en la red. Y *gestionar el acceso al medio*, ajustando en el nodo la prioridad de servicio del búfer de salida y del mecanismo de control de acceso al medio, en aquellos casos que el nodo este en congestión o necesite mayor prioridad. Esta sinergia debería mejora los resultados de operación de los módulos de detección, notificación y resolución del protocolo.

5. Conclusiones

En este trabajo se expusieron los principales mecanismos que pueden ser utilizados para el control de congestión, bajo el nuevo paradigma crosslayer; utilizado para el diseño de protocolos de comunicación en redes de conmutación de paquetes, con mayor énfasis en las WSN. Se mostró en la sección 3 que la congestión es uno de los problemas que afectan fuertemente el rendimiento global de la red y los niveles de QoS ofrecidos. Además, que para abordar los problemas de congestión en las WSN es necesario que el protocolo integre las funciones de detección, notificación y resolución de congestión; y que estén diseñados bajo optimización crosslayer. Lo presentado puede servir de guía de diseño para protocolos de transporte que traten de mitigar el problema de congestión de la red; lo cual haría un uso más eficiente del ancho de banda y los recursos de los dispositivos de la red. Que en el caso de las WSN es un recurso muy limitado si se compara con tecnologías inalámbricas como Bluetooth, Wi-Fi o 3G. Como trabajo futuro de investigación se propone: Determinar las funciones y parámetros que deben tomarse en cuenta para la optimización crosslayer y evaluar la eficiencia de los mecanismos propuestos.

Referencias

1. M.A. Ameen, A. Nessa, and Kyung Sup Kwak. QoS issues with focus on wireless body area networks. In *Convergence and Hybrid Information Technology, 2008. ICCIT '08. Third International Conference on*, volume 1, pages 801–807, November 2008.
2. Benoît Latré, Bart Braem, Ingrid Moerman, Chris Blondia, and Piet Demeester. A survey on wireless body area networks. *Wireless Networks*, 17(1):1–18, 2011.
3. M. Aykut Yigitel, Ozlem Durmaz Incel, and Cem Ersoy. QoS-aware MAC protocols for wireless sensor networks: A survey. *Comput. Netw.*, 55(8):1982–2004, June 2011.
4. Chieh-Yih Wan, Shane B. Eisenman, Andrew T. Campbell, and Jon Crowcroft. Siphon: overload traffic management using multi-radio virtual sinks in sensor networks. In *Proceedings of the 3rd international conference on Embedded networked sensor systems*, SenSys '05, page 116–129, New York, NY, USA, 2005. ACM.
5. Phumzile Malindi. QoS in telemedicine. In Georgi Grasczew, editor, *Telemedicine Techniques and Applications*. InTech, June 2011.
6. M.C. Vuran and I.F. Akyildiz. XLP: a cross-layer protocol for efficient communication in wireless sensor networks. *IEEE Transactions on Mobile Computing*, 9(11):1578–1591, November 2010.
7. Y.G. Iyer, S. Gandham, and S. Venkatesan. STCP: a generic transport layer protocol for wireless sensor networks. In *14th International Conference on Computer Communications and Networks, 2005. ICCCN 2005. Proceedings*, pages 449–454, October 2005.
8. Md. Abdur Rahman, Abdulmotaleb El Saddik, and Wail Gueaieb. Wireless sensor network transport layer: State of the art. In S.C. Mukhopadhyay and R.Y.M. Huang, editors, *Sensors*, volume 21 of *Lecture Notes in Electrical Engineering*, pages 221–245. Springer Berlin Heidelberg, 2008.
9. F.K. Shaikh, A. Khelil, A. Ali, and N. Suri. TRCCIT: tunable reliability with congestion control for information transport in wireless sensor networks. In *Wireless Internet Conference (WICON), 2010 The 5th Annual ICST*, pages 1–9, March 2010.

10. Chieh-Yih Wan, Shane B. Eisenman, and Andrew T. Campbell. CODA: congestion detection and avoidance in sensor networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems*, SenSys '03, page 266–279, New York, NY, USA, 2003. ACM.
11. E. Giancoli, F. Jabour, and A. Pedroza. CTCP: reliable transport control protocol for sensor networks. In *International Conference on Intelligent Sensors, Sensor Networks and Information Processing, 2008. ISSNIP 2008*, pages 493–498, December 2008.
12. Vehbi Cagri Gungor, Özgür B. Akan, and Ian F. Akyildiz. A real-time and reliable transport (RT)² protocol for wireless sensor and actor networks. *IEEE/ACM Trans. Netw.*, 16(2):359–370, April 2008.
13. C. Wang, B. Li, K. Sohraby, M. Daneshmand, and Y. Hu. Upstream congestion control in wireless sensor networks through cross-layer optimization. *IEEE Journal on Selected Areas in Communications*, 25(4):786–795, May 2007.
14. A. Sharif, V.M. Potdar, and A.J.D. Rathnayaka. LCART: a cross-layered transport protocol for heterogeneous WSN. In *2010 IEEE Sensors*, pages 793–796, November 2010.
15. Muhammad Mahbub Alam and Choong Seon Hong. CRRT: congestion-aware and rate-controlled reliable transport in wireless sensor networks. *IEICE TRANSACTIONS on Communications*, E92-B(1):184–199, January 2009.
16. Nurcan Tezcan and Wenye Wang. ART: an asymmetric and reliable transport mechanism for wireless sensor networks. *Int. J. Sen. Netw.*, 2(3/4):188–200, April 2007.
17. Chonggang Wang, K. Sohraby, Bo Li, M. Daneshmand, and Yueming Hu. A survey of transport protocols for wireless sensor networks. *IEEE Network*, 20(3):34–40, June 2006.
18. M.O. Rahman, M.M. Monowar, and Choong Seon Hong. A QoS adaptive congestion control in wireless sensor network. In *Advanced Communication Technology, 2008. ICACT 2008. 10th International Conference on*, volume 2, pages 941–946, February 2008.