

Algoritmo para Contar Nodos en Redes Inalámbricas con Mensajes Retrasados

Manuel Contreras¹, Eric Gamess²
mcontre@ula.ve, eric.gamess@ciens.ucv.ve

¹ Escuela de Computación, Universidad Central de Venezuela, Caracas, Venezuela

² Laboratorio de Comunicación y Redes, Universidad Central de Venezuela, Caracas, Venezuela

Resumen: El conteo de objetos (personas, animales, dispositivos, vehículos, etc.) ha interesado a la comunidad científica desde hace tiempos inmemoriales, ya que puede ser utilizado como una herramienta básica en muchas aplicaciones. Hasta la fecha, la mayoría de los trabajos hechos en el área están basados en cámaras digitales, cámaras de video, torniquetes, brazos mecánicos, tubos neumáticos, luz infrarroja y otros tipos de sensores. Sin embargo, pocos trabajos han sido desarrollados usando las comunicaciones inalámbricas para contar nodos. En este artículo, introducimos una versión mejorada de un algoritmo de conteo de nodos que habíamos propuesto para contar dispositivos equipados de una interfaz de red inalámbrica, donde se consideran además en el conteo total de nodos, aquellos mensajes de conteo retrasados que pudieran ser enviados por los nodos en un tiempo posterior al inicialmente determinado. Para validar nuestra propuesta, simulamos el algoritmo con OMNeT++/INET y comparamos los resultados con la versión preliminar del algoritmo. Los resultados obtenidos en el conteo son alentadores, ya que la nueva versión del algoritmo mejora en forma sustancial el algoritmo preliminar. Además, las simulaciones muestran que el algoritmo no solamente funciona para nodos estacionarios, sino que también hace un conteo eficiente cuando los nodos son móviles con velocidades moderadas.

Palabras Clave: Redes Inalámbricas; Conteo; Algoritmos; Simulación; OMNeT++; INET.

Abstract: Counting objects (people, animals, devices, vehicles, etc.) has interested the research community since time immemorial, because it can be used as a basic tool in many applications. Up to now, most of the work done in the area is based on digital cameras, video cameras, turnstiles, mechanical arms, pneumatic tubes, infrared beams, and other types of sensors. However, just a few studies have been developed using wireless communications to count nodes. In this paper, we introduce an improved version of an algorithm for node counting that we previously proposed to count devices equipped with a wireless network interface, where we also consider in the total counting of nodes, the delayed counting messages that can be sent by nodes at a later time than the initially determined. To validate our proposal, we simulate the algorithm with OMNeT++/INET and we compare the results with the preliminary version of the algorithm. The results obtained in the counting are encouraging, since the new version of the algorithm substantially improves the preliminary algorithm. Furthermore, the simulations show that the algorithm not only works for stationary nodes, but also makes an efficient counting when nodes are moving with moderate speeds.

Keywords: Wireless Networks; Counting; Algorithms; Simulation; OMNeT++; INET.

I. INTRODUCCIÓN

Contar objetos tales como personas, animales, dispositivos, vehículos, entre otros, ha sido de gran interés en diversos ámbitos o campos de estudio. Por ejemplo, las personas pueden ser contadas en eventos políticos, deportivos, musicales o sociales, por muchas razones que incluyen estadísticas de afluencia, capacidad, optimización de flujo, seguridad, entre otros. En el transporte público, la información del número de personas en las unidades o en los paraderos podría ser usada para mejorar el servicio entregado al aumentar o disminuir la frecuencia de las unidades según el volumen de pasajeros. En un estacionamiento, el número de vehículos estacionados es

útil para determinar el número de puestos libres. Para la gestión eficaz de la fauna, la conservación y la ecología, es muy relevante obtener una estimación confiable del tamaño o densidad de las poblaciones de las especies de animales, en particular cuando están en peligro de extinción. Otro ejemplo es el conocimiento del número de personas que hay en lugares con abundante afluencia para así determinar los mejores patrones de seguridad en caso de alguna contingencia. En la actualidad, se intenta solventar todas estas tareas mediante operadores humanos o sistemas automáticos, principalmente basados en cámaras digitales, cámaras de video, torniquetes, brazos mecánicos, tubos neumáticos, luz infrarroja, sensores magnéticos, radares, o cables piezoeléctricos [1][2][3].

El método más tradicional para estimar el número de objetos es el conteo manual. En este sentido, observadores entrenados recolectan los datos necesarios que por medio de contadores automáticos [4], no se pueden obtener eficientemente. Según [5], un observador puede contar aproximadamente de 1000 a 2000 personas en una hora sin ayuda tecnológica. Ahora, con un contador (un dispositivo pequeño que cuenta pulsos), dicha persona puede duplicar la cantidad de personas contadas. Similarmente, otros objetos que pueden ser contados por un observador entrenado, son las bicicletas y los carros. Sin embargo, el método de conteo manual presenta varios inconvenientes como la necesidad de contratar personal para desempeñar este tipo de trabajo y la posibilidad que se cometan errores importantes en situaciones con alta densidad de objetos.

También, se puede estimar el número de objetos mediante los sistemas que usan contadores mecánicos. Los más representativos son los torniquetes que se emplean para contar personas en un sistema de control de acceso, una a la vez, presentando un alto grado de fiabilidad, un alto costo y un bajo grado de automatización del procesamiento de la información. Estos sistemas eliminan el costo de personal, pero reducen significativamente los espacios dedicados a la entrada/salida de personas, resultando en muchos casos en demoras e incomodidades en el tránsito de las mismas, debido a su inherente aparataje.

Otro caso es el uso de un contador térmico, el cual tiene un sensor que identifica el gradiente entre la temperatura ambiental y la temperatura corporal. Este tipo de contador es ineficiente cuando la temperatura ambiental es semejante a la corporal [6].

Otra técnica, basada en la tecnología infrarroja, usa un simple haz de luz que normalmente está conectado a una pantalla LCD (pantalla de cristal líquido) ubicada en el lado de la puerta de las unidades de transporte (por ejemplo, en trenes, autobuses, ferries). Cuando un pasajero entra en la unidad de transporte, se interrumpe o se corta el haz de luz, siendo detectado por un micro controlador que reacciona incrementando un contador. Sin embargo, esta técnica sufre de un problema denominado oclusión. La oclusión es generada por el amontonamiento de personas, es decir, al cruzar un grupo de personas al mismo tiempo, el contador sólo detecta una [6] ya que no hay discontinuidad en la intercepción del rayo.

Los sensores magnéticos o bucles inductivos [4] son usados en los sistemas de transporte donde son enterrados en el asfalto o puestos en la superficie de la carretera, de manera que cuando un vehículo pase sobre el mismo, el campo magnético del motor del vehículo modifique el campo registrado por el sensor. Esta tecnología permite además de detectar y contar vehículos, obtener su velocidad y clasificarlos según su tipo.

No obstante, durante los últimos años la comunicación de datos ha ganado un mayor interés e importancia, razón por la cual hoy en día, es común que los objetos (personas, animales, dispositivos, vehículos, etc) estén equipados de una interfaz inalámbrica para fines de comunicación, de modo que los algoritmos pueden aprovechar esta tecnología para contar objetos. Como principal objetivo de este trabajo, se propone el desarrollo de un algoritmo para contar nodos usando tecnologías inalámbricas, partiendo de las ideas planteadas en el algoritmo propuesto en [7], dónde se toman en cuenta en el

conteo aquéllos mensajes retrasados que pudieran ser enviados por los nodos en un tiempo posterior al inicialmente establecido. Las simulaciones que corrimos muestran que esta nueva versión del algoritmo mejora sustancialmente el cálculo del número de nodos en comparación al algoritmo anterior [7], tanto en escenarios estacionarios, como para nodos móviles con velocidad moderada.

El resto de este artículo está organizado como sigue: En la Sección II, se hace una revisión de los trabajos relacionados. En la Sección III, se describe el algoritmo para contar nodos usando tecnologías inalámbricas, tomando en cuenta los mensajes retrasados. La Sección IV introduce brevemente las herramientas de simulación que se utilizaron para validar el algoritmo propuesto. En la Sección V, se presenta un análisis comparativo de los resultados de nuestras simulaciones. Finalmente, en la Sección VI se discuten las conclusiones del artículo y los trabajos futuros.

II. TRABAJOS RELACIONADOS

La detección y el conteo de objetos representan una herramienta que tiene numerosas aplicaciones, y debido a su gran utilidad se ha venido efectuando en diversas formas o disciplinas con el uso de varias tecnologías que hacen que en la realidad sea aplicable a muchas situaciones. Actualmente, se pueden encontrar diferentes alternativas que permiten llevar a cabo la tarea de detectar y contar nodos con mayor o menor precisión, basadas primordialmente en métodos, técnicas o tecnologías convencionales “in-situ”, tales como cámaras digitales, cámaras de video, torniquetes, brazos mecánicos, tubos neumáticos, luz infrarroja, sensores magnéticos, radares o cables piezoeléctricos, que son utilizados por ejemplo para: detectar y contar personas en los sistemas de vigilancia de los espacios públicos como centros comerciales, aeropuertos, eventos públicos; estimar el tamaño de una población de organismos (tal es el caso de las especies de animales); determinar el número de puestos libres en estacionamientos; contar pasajeros en el transporte público; contar vehículos en tiempo real en las carreteras, intersecciones, semáforos, entre otros. Por ejemplo, en los trabajos [8][9][10][11][12][13], los autores plantean diferentes métodos y técnicas para contar personas en los sistemas de vigilancia o contar pasajeros en el transporte público, utilizando registros de cámaras digitales o de video. En [14], los autores desarrollaron un algoritmo de conteo basado en el uso de una línea láser infrarrojo como referencia de un área de detección de personas. Los autores de [15][16] usaron el procesamiento de imágenes obtenidas de cámaras de video para contar vehículos en tiempo real. Ki [17] diseñó un sistema de conteo que hace un seguimiento de personas que transitan por una zona de detección con una red de sensores infrarrojos. Gros-Desormeaux et al. [18] implementaron dos algoritmos de conteo acústico para contar aves usando sensores equipados con micrófonos. Litzenberger et al. [19] presentan un sistema integrado que comprende un sensor óptico sensible al movimiento y un DSP de bajo costo para detectar y contar vehículos. En [20], se utiliza un dispositivo láser en carreteras con el objetivo de realizar un conteo y una clasificación de los vehículos en función de la información obtenida con el láser. Los autores de [21][22] utilizaron sensores láser para detectar y contar a las personas en una estación de autobuses. Los trabajos [23][24] se basan sobre imágenes aéreas para estimar poblaciones de animales. Kissell

y Nimmo [25] utilizan imágenes de cámaras infrarrojas para estimar la población de ciervos.

En el contexto de contar objetos utilizando las redes inalámbricas, son muy pocos los trabajos que se han desarrollado. Por ejemplo, Gamess y Mahgoub [26] presentan un enfoque basado en VANET, para determinar: (1) la posición del último vehículo y (2) el número de vehículos, en una línea de vehículos parados en un semáforo.

Tal como podemos apreciar, la mayor parte del trabajo realizado en este campo se basa en diseños con tecnologías “in-situ”.

III. ALGORITMO PARA CONTAR NODOS EN REDES INALÁMBRICAS CON MENSAJES RETRASADOS

En esta sección, se describe el algoritmo para contar nodos usando tecnologías inalámbricas, tomando en cuenta los mensajes retrasados que pudieran ser enviados por los nodos.

A. Consideraciones Básicas para Contar Nodos

El presente algoritmo representa una modificación de las ideas planteadas en la propuesta desarrollada en [7], considerándose además en el conteo total de nodos, aquellos mensajes de conteo retrasados que pudieran ser enviados por los nodos en un tiempo posterior al especificado. El enfoque básico de dicho algoritmo es:

1. Propagar un mensaje “broadcast” (llamado COUNT_REQUEST) desde el nodo origen hacia los nodos que están lejos del mismo.
2. Propagar mensajes “unicast” (COUNT_REPLY) desde los nodos que se encuentran lejos del origen hacia el mismo, con el número total de nodos contados hasta el momento (llamado *Total* en el algoritmo).

La Figura 1 muestra un diagrama de tiempo relacionado con la propagación de los mensajes COUNT_REQUEST (enviados como broadcast) y los mensajes COUNT_REPLY (enviados como unicast), donde:

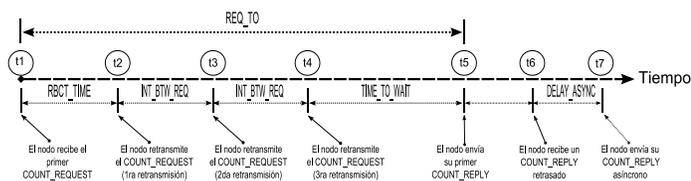


Figura 1: Diagrama de Tiempo

- t1 = Es el tiempo en el cual el nodo recibe el primer COUNT_REQUEST.
- t2 = Es el tiempo en el cual el nodo envía el primer COUNT_REQUEST.
- t3 = Es el tiempo en el cual el nodo envía el segundo COUNT_REQUEST.
- t4 = Es el tiempo en el cual el nodo envía el tercer COUNT_REQUEST.
- t5 = Es el tiempo en el cual el nodo envía su COUNT_REPLY síncrono.
- t6 = Es el tiempo en el cual el nodo recibe un posible mensaje COUNT_REPLY retrasado desde otro nodo.
- t7 = Es el tiempo en el cual el nodo envía un posible COUNT_REPLY asíncrono.

t7 = Es el tiempo en el cual el nodo envía un posible COUNT_REPLY asíncrono.

La Figura 1 también muestra que los nodos retransmitirán tres mensajes COUNT_REQUEST en fila y un mensaje COUNT_REPLY síncrono después de un tiempo específico. Es necesario enviar varias veces (3 veces en nuestro algoritmo) el mensaje COUNT_REQUEST ya que es un mensaje de broadcast que puede entrar en colisión con otros mensajes sin que esta sea detectada. Además, es posible que el nodo reciba un mensaje COUNT_REPLY retrasado. En este caso, deberá volver a hacer su conteo antes de propagarlo hacia el origen con un COUNT_REPLY asíncrono. Los siguientes parámetros permiten determinar los tiempos de envío de los mensajes COUNT_REQUEST y COUNT_REPLY (véase Figura 1):

- RBCT_TIME (Rebroadcast Time): Es el tiempo entre la recepción del primer COUNT_REQUEST, y la primera retransmisión del COUNT_REQUEST por el nodo.
- INT_BTW_REQ (Interval Between Request): Es el intervalo de tiempo entre el envío de mensajes COUNT_REQUEST. Es decir, representa el tiempo entre dos mensajes COUNT_REQUEST consecutivos enviados por un nodo.
- REQ_TO (Request Timeout): Es el tiempo transcurrido entre la recepción del primer COUNT_REQUEST y el momento cuando el nodo tiene que enviar su mensaje COUNT_REPLY síncrono al nodo *nodeToGoBack*.
- TIME_TO_WAIT: Es el tiempo que un nodo espera después de la retransmisión del último mensaje COUNT_REQUEST y el envío del mensaje COUNT_REPLY al nodo *nodeToGoBack*. Este tiempo debe ser lo suficientemente grande como para permitir la propagación de los mensajes COUNT_REQUEST desde el nodo actual hacia los nodos que están lejos del origen, y la propagación de los mensajes COUNT_REPLY desde los nodos que se encuentran lejos del origen hacia el nodo actual. Este tiempo no es un valor constante sino que será computado por cada nodo de acuerdo a *HopLimit* y que tan lejos está del origen (llamado *HopAway* en el algoritmo).
- DELAY_ASYNC: Es el tiempo que un nodo espera después de la recepción de posibles mensajes COUNT_REPLY retrasados, para enviar un COUNT_REPLY asíncrono a su respectivo nodo *nodeToGoBack*.

B. Mensajes COUNT_REQUEST y COUNT_REPLY

Los mensajes COUNT_REQUEST y COUNT_REPLY comparten la misma Unidad de Protocolo de Datos (PDU) que se compone de seis campos (ver Figura 2).

Message Type	Sequence Number	Timestamp	HopAway	HopLimit	Total
2 bytes	4 bytes	4 bytes	2 bytes	2 bytes	4 bytes

Figura 2: Mensajes COUNT_REQUEST y COUNT_REPLY

El campo *Message Type* puede ser 0 o 1, y es usado para identificar el tipo de mensaje. Un valor de 0 denota un

COUNT_REQUEST, mientras 1 indica un COUNT_REPLY. *Sequence Number* es usado para corresponder los mensajes COUNT_REQUEST con sus respectivos mensajes COUNT_REPLY, y para distinguir entre diferentes peticiones de un mismo nodo. *Timestamp* es una muestra de tiempo tomada en el origen cuando este inicia el proceso de conteo. Su finalidad es controlar los mensajes que están fuera de tiempo y los ataques de repetición. El origen y los nodos difunden mensajes COUNT_REQUEST junto con el argumento *HopAway* que representa el número de saltos que hay entre el origen del mensaje COUNT_REQUEST y su receptor. El origen que comienza el proceso debe especificar un valor de *HopAway* igual a 1. Cada nodo que retransmite el mensaje, seleccionará el más pequeño *HopAway* recibido hasta el momento y lo incrementará en 1. Cada vez que un nodo actualiza su campo *HopAway*, debido a la recepción de un mejor COUNT_REQUEST (más cercano al origen), tiene que actualizar su variable *nodeToGoBack*, que es una referencia al nodo con el más bajo *HopAway*, del cual el nodo actual ha recibido un mensaje COUNT_REQUEST. El campo *HopLimit* permite controlar hasta qué distancia (número de saltos) los mensajes COUNT_REQUEST son propagados desde el origen hacia los demás nodos, delimitando así el rango de conteo. El campo *Total* representa el número de nodos contados hasta el momento, siendo igual a 0 en mensajes COUNT_REQUEST. Es importante mencionar que antes de enviar un mensaje COUNT_REPLY (al nodo identificado por *nodeToGoBack*), un nodo debe actualizar este campo de acuerdo a los mensajes COUNT_REPLY recibidos hasta ahora, y agregar 1 a la suma para contarse a sí mismo.

C. Algoritmo

El origen primero propaga tres mensajes de tipo COUNT_REQUEST con un valor de *HopAway* igual a 1 (separado por INT_BTW_REQ). Cuando un nodo recibe el primer mensaje COUNT_REQUEST, hará lo siguiente:

- RBCT_TIME: El nodo propagará un COUNT_REQUEST con un *HopAway* igual al mínimo *HopAway* recibido en este tiempo +1. También almacenará en la variable *nodeToGoBack* el ID del nodo que envió el COUNT_REQUEST con el más pequeño valor de *HopAway*.
- RBCT_TIME + (1 * INT_BTW_REQ): El nodo retransmite el mensaje COUNT_REQUEST (debido a una posible colisión con el primer COUNT_REQUEST enviado). Si es necesario, los campos del mensaje COUNT_REQUEST son actualizados.
- RBCT_TIME + (2 * INT_BTW_REQ): El nodo retransmite el mensaje COUNT_REQUEST (debido a una posible colisión con el primer y segundo COUNT_REQUEST enviados). Si es necesario, los campos del mensaje COUNT_REQUEST son actualizados.
- REQ_TO: Si el nodo recibió mensajes COUNT_REPLY, entonces el nodo calcula el total de nodos basado en la variable *Total* recibida en los mensajes COUNT_REPLY (+1 para sumarse a sí mismo en el conteo total) y envía el resultado a su nodo *nodeToGoBack* como un mensaje unicast. Si después de REQ_TO el nodo no ha recibido ningún mensaje COUNT_REPLY, entonces genera un

COUNT_REPLY con *Total* igual a 1 (este 1 representa el conteo de sí mismo) y lo envía al nodo *nodeToGoBack* como un mensaje unicast.

- DELAY_ASYNC: Si el nodo recibe mensajes COUNT_REPLY retrasados, entonces nuevamente vuelve a computar el número total de nodos basado en la variable *Total* recibida en los mensajes COUNT_REPLY (+1 para sumarse a sí mismo en el conteo total) y luego envía el resultado a su nodo *nodeToGoBack* como un mensaje COUNT_REPLY asíncrono.

D. Ejemplo de un Escenario de Propagación de Mensajes

En la Figura 3, se aprecia un escenario con 18 nodos, que incluyen el origen (identificado por O) que requiere un conteo. Los círculos alrededor de los nodos representan el rango de propagación de los mensajes enviados por ellos mismos.

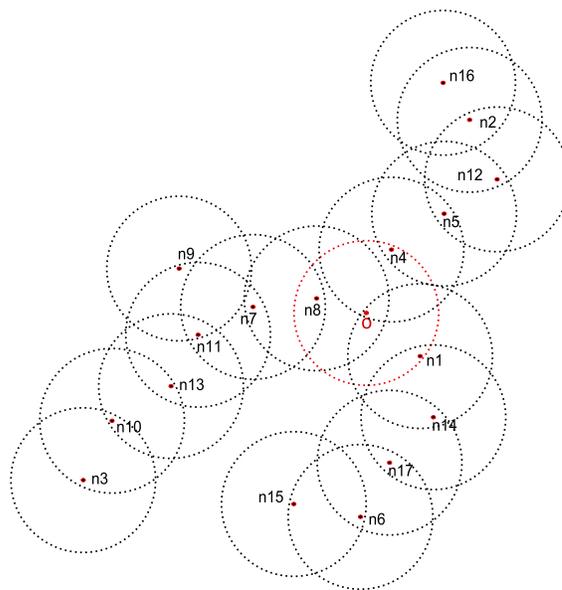


Figura 3: Ejemplo de un Escenario de Propagación de Mensajes

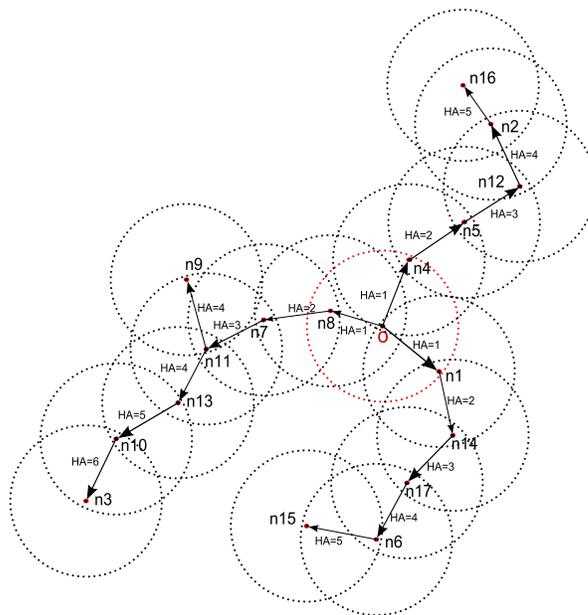


Figura 4: Propagación de Mensajes COUNT_REQUEST

interactúan entre sí, siendo una de sus principales fortalezas su interfaz gráfica de usuario (GUI), que permite crear archivos NED (un lenguaje de descripción para definir la estructura del modelo) e inspeccionar el estado de cada componente durante las simulaciones. Para simulaciones de redes de datos, OMNeT++ requiere de extensiones externas tales como INET. El “Framework” INET es un paquete de simulación de código abierto de comunicaciones de datos para el ambiente de simulación OMNeT++.

En la presente investigación, se decidió utilizar OMNeT++ por dos razones principales: (1) su potente interfaz gráfica de usuario que facilita la depuración de las simulaciones y (2) que es un proyecto muy activo con muchos modelos que se actualizan constantemente.

Se simularon diferentes escenarios donde los nodos fueron distribuidos aleatoriamente sobre un área cuadrada de 800m x 800m. El modelo de movilidad aleatorio RandomWPMobility fue seleccionado para reflejar el escenario más general de movilidad o desplazamiento de los nodos. Asimismo, para todas las simulaciones se seleccionó el estándar de comunicación inalámbrica WiFi (IEEE 802.11g) con una tasa de transferencia de 54 Mbps. El modelo de propagación en el espacio libre fue elegido para la atenuación de la señal con la distancia.

V. RESULTADOS DE LAS SIMULACIONES

En esta sección se presentan y se discuten algunos de los resultados de las simulaciones efectuadas.

A. Escenarios con Nodos y el Origen Estacionarios

Al igual que en el algoritmo planteado en [7], se ejecutó varias simulaciones para escenarios estacionarios, es decir, donde los nodos y el origen permanecen inmóviles.

Hicimos simulaciones con 50 nodos estacionarios, colocados aleatoriamente en un área cuadrada de 800m x 800m, con un valor del rango de propagación de 250 m, y un origen situado en el centro de la misma. La Tabla I muestra los resultados obtenidos cuando se varía los parámetros RBCT_TIME, INT_BTW_REQ y DELAY. Los resultados de las últimas 3 columnas son representados como valores a/b , donde a representa el resultado del algoritmo anterior [7] y b el resultado obtenido con el nuevo algoritmo. Tal como se aprecia en la Tabla I, la nueva versión del algoritmo contó un total de nodos más eficazmente que el algoritmo propuesto en [7]. Por ejemplo, para valores de los parámetros RBCT_TIME=0.01, INT_BTW_REQ=0.01 y DELAY=0.03, el origen debería contar 50 nodos. Ahora, cuando se ejecuta la simulación, el algoritmo propuesto en [7] contó 10 nodos con un tiempo de respuesta de 0.15s y un total de mensajes enviados igual a 155. Sin embargo, el nuevo algoritmo contó 47 nodos con un tiempo de respuesta de 1.08s y un número total de mensajes enviados igual a 178, resultando en una diferencia de conteo bastante notable. La Tabla I muestra la importancia de escoger bien los valores de los parámetros RBCT_TIME, INT_BTW_REQ y DELAY. En efecto, según los valores escogidos, se obtendrá un mejor tiempo de respuesta o una mayor precisión de conteo. Los desarrolladores de aplicaciones tendrán que seleccionar estos parámetros de acuerdo a sus necesidades [7].

Tabla I: Importancia de los Parámetros RBCT_TIME, INT_BTW_REQ y DELAY

RBCT_TIME, INT_BTW_REQ, DELAY	Rango de Propagación = 250m			
	Nodos Alcanzables	Nodos Contados	Tiempo de Respuesta	Número Total de Msg Enviados
0.01s, 0.01s, 0.03s	50	10/47	0.15 s/1.08 s	155/178
0.05s, 0.05s, 0.10s	50	15/49	0.63 s/1.39 s	158/181
0.05s, 0.05s, 0.20s	50	18/50	0.88 s/1.43 s	158/174
0.10s, 0.10s, 0.20s	50	23/50	1.26 s/1.90 s	167/180
0.10s, 0.10s, 0.40s	50	34/50	1.76 s/2.23 s	167/183
0.20s, 0.20s, 0.30s	50	43/50	2.28 s/2.68 s	173/177
0.25s, 0.25s, 0.30s	50	48/50	2.66 s/2.21 s	173/175
0.30s, 0.30s, 0.50s	50	50/50	3.24 s/2.93 s	173/173

En la Tabla II, variamos el número total de nodos estacionarios (10, 15, 20, 25, 30, 35, 40 y 50) ubicados aleatoriamente en un área cuadrada de 800m x 800m, y su rango de propagación (100m, 150m, 200m, 250m y 300m). Los resultados son representados como valores $a/b/c$, donde a denota el número de nodos que están dentro del alcance del origen usando enrutamiento multisalto (es decir, nodos que deben ser contados), b el número de nodos actualmente contados por el algoritmo propuesto en [7] (que de aquí en adelante denominaremos Algoritmo I), y c el total de nodos contados por el algoritmo propuesto en esta investigación (que llamaremos en lo sucesivo como Algoritmo II). Seleccionamos valores apropiados para los parámetros RBCT_TIME=0.2s, INT_BTW_REQ=0.2s y DELAY=0.4s, de acuerdo a los resultados encontrados en la Tabla I, iniciando el origen con un valor de $HopLimit=3$.

Tabla II: Conteo de Nodos en Diferentes Escenarios Estacionarios

Número Total de Nodos	Rango de Propagación en Metros				
	100m	150m	200m	250m	300m
10	1/1/1	1/1/1	3/3/3	10/10/10	10/10/10
15	2/2/2	4/4/4	7/7/7	14/14/14	15/15/15
20	2/2/2	4/4/4	11/11/11	19/19/19	20/20/20
25	4/4/4	8/8/8	14/14/14	23/22/23	25/24/25
30	5/5/5	14/14/14	24/23/24	29/28/29	30/30/30
35	7/7/7	16/16/16	29/27/29	34/33/34	35/35/35
40	8/8/8	22/22/22	34/33/34	39/39/39	40/37/40
50	11/11/11	28/26/26	48/43/47	50/50/50	50/50/50

Dado que existen nodos que no son alcanzables en estos escenarios, podemos observar que el algoritmo II mejora sustancialmente al algoritmo I. Por ejemplo, la Tabla II muestra que para un escenario de 50 nodos y un rango de propagación igual a 200 metros, el origen debería contar 48 nodos (en este caso 2 nodos no son accesibles desde el origen porque están fuera del rango de conteo). Ahora, cuando se ejecuta la simulación, el algoritmo I contó 43 nodos, en lugar de 48, lo cual es una buena aproximación. Sin embargo, el algoritmo II contó 47 nodos, siendo mucho más eficaz. Estas mismas diferencias se pueden apreciar para otros casos de simulación, tales como (25, 250), (25, 300), (30, 200), (30, 250), (35, 200), (35, 250), (40, 200) y (40, 300), donde el primer valor del par representa el número total de nodos mientras que el segundo valor representa el rango de propagación. Asimismo, la Tabla III proporciona información adicional (tiempo de respuesta y número total de mensajes enviados) para los mismos experimentos, para un rango de propagación de 250m. Por ejemplo, para un total de 35 nodos, 34 de los 34 nodos alcanzables por el origen fueron contados

efectivamente por el algoritmo II en 2.67s y con un número total de mensajes enviados igual a 101.

Tabla III: Tiempo de Respuesta y Número Total de Mensajes Enviados en un Escenario Estacionario Específico

Número Total de Nodos	Rango de Propagación = 250m			
	Nodos Alcanzables	Nodos Contados	Tiempo de Respuesta	Número Total de Msg Enviados
10	10	10	2.52 s	38
15	14	14	2.77 s	54
20	19	19	2.70 s	74
25	23	23	2.67 s	93
30	29	29	2.90 s	104
35	34	34	2.67 s	101
40	39	39	2.72 s	153
50	50	50	2.66 s	177

B. Escenarios con Nodos Móviles y el Origen Estacionario

En esta subsección, analizamos el rendimiento del algoritmo II en comparación con el algoritmo I, cuando los nodos tienen movilidad, pero el origen es estacionario y situado en el centro del área cuadrada de 800m x 800m. Para el desplazamiento de los nodos, se utilizó el modelo de movilidad aleatorio RandomWPMobility, variando la velocidad de los mismos de 1 mps (metros por segundo) a 45 mps, con un tiempo de espera de 0s (intervalo de tiempo entre alcanzar un destino y salir hacia un nuevo destino).

En la Tabla IV, variamos el rango de propagación de los nodos y el origen. Para todos estos escenarios, elegimos 30 nodos móviles, posicionados inicialmente de forma aleatoria. Los valores de los parámetros son: RBCT_TIME=0.2s, INT_BTW_REQ=0.2s y DELAY=0.4s. El origen inicia el proceso de conteo con HopLimit=3. Aquí también, los resultados son presentados como valores a/b/c, donde a denota el número de nodos que están dentro del alcance del origen usando enrutamiento multisalto, b el número de nodos actualmente contados por el algoritmo I, y c el total de nodos contados por el algoritmo II. Como era de suponer, podemos apreciar que el algoritmo II tiene una mejor precisión de conteo. También se puede observar que a medida que se incrementa la velocidad, en ambos algoritmos se pierde precisión en el conteo de nodos. Los mejores resultados son obtenidos cuando la velocidad varía de 1 a 20 mps.

Tabla IV: Conteo de Nodos cuando se Varía la Velocidad de los Nodos y los Valores del Rango de Propagación (Velocidad del Origen=0 mps)

Velocidad de Nodos (mps)	Rango de Propagación en Metros				
	100m	150m	200m	250m	300m
1	5/5/5	11/11/11	17/17/17	29/29/29	30/29/30
3	4/4/4	14/13/13	23/23/23	30/29/30	30/30/30
5	5/5/5	10/10/10	24/23/24	30/28/29	30/30/30
10	4/3/3	22/19/19	29/29/28	30/30/30	30/30/30
20	11/6/6	26/20/20	30/29/29	30/29/27	30/30/30
30	5/1/1	23/7/6	30/18/21	30/20/24	30/19/26
40	9/7/7	11/6/8	26/18/19	30/17/20	30/18/26
45	5/4/4	10/4/5	29/19/20	30/25/26	30/24/25

En la Tabla V, se optó por un rango de propagación fijo de 250m, variando tanto la velocidad de los nodos como el número de nodos. Según los resultados obtenidos, se deduce que en ambos algoritmos, igualmente como ocurre en los experimentos de la Tabla IV, se tiene un buen conteo de nodos para velocidades de hasta 20 mps; mientras que para velocidades altas, aunque el conteo se degrada, el algoritmo II

es mucho más preciso. Por ejemplo, como se muestra en la Tabla V, para el caso de 40 nodos en total y una velocidad de 45mps, el origen debería contar 40. Al ejecutar la simulación, el algoritmo I contó 30 nodos, mientras que el algoritmo II contó 33, siendo más efectivo al igual que para los demás casos.

Tabla V: Conteo de Nodos cuando se Varía la Velocidad de los Nodos y el Número Total de Nodos

Velocidad de Nodos (mps)	Número Total de Nodos						
	10	15	20	25	30	40	50
1	7/7/7	15/15/15	16/16/16	18/18/18	29/29/29	40/38/40	50/49/50
3	5/5/5	7/7/7	17/17/17	24/22/22	30/29/27	40/38/40	50/49/50
5	2/2/2	7/7/7	20/20/20	24/24/24	30/28/29	40/40/40	50/47/50
10	8/8/8	15/15/15	20/19/19	25/21/21	30/30/30	40/36/38	50/46/48
20	8/9/9	15/15/15	19/17/18	25/22/22	30/29/29	40/37/38	50/46/46
30	6/6/6	15/13/14	20/18/18	24/22/21	30/20/20	40/31/35	50/27/29
40	10/9/9	14/15/15	19/13/14	25/17/15	30/17/17	40/33/34	50/38/39
45	10/7/6	15/11/14	20/17/17	25/14/15	30/25/26	40/30/33	50/27/39

C. Escenarios con Nodos Estacionarios y el Origen Móvil

En esta subsección, se estudia comparativamente el comportamiento del algoritmo II con respecto al algoritmo I, para nodos estacionarios y un origen con movilidad. Al comienzo de las simulaciones, los nodos estacionarios son situados aleatoriamente en un área cuadrada de 800m x 800m, siendo colocado el origen en el centro de la misma, desplazándose de acuerdo al modelo de movilidad aleatorio RandomWPMobility, sin parar en las posiciones visitadas, es decir, con un tiempo de espera de 0s. Para estos experimentos, igualmente los valores de los parámetros son: RBCT_TIME=0.2s, INT_BTW_REQ=0.2s y DELAY=0.4s. El origen inicia el proceso de conteo con un HopLimit=3.

En la Tabla VI se presentan los resultados de la simulación de escenarios con 30 nodos, en los que se varía la velocidad del origen, así como el rango de propagación del mismo y de los nodos. Se puede observar que para ambos algoritmos, se tiene una gran precisión de conteo para velocidades de hasta 20 mps, que a medida que ésta se incrementa, la precisión del conteo disminuye, pero siendo significativamente más eficiente en el algoritmo II.

Tabla VI: Conteo de Nodos cuando se Varía la Velocidad del Origen y los Valores del Rango de Propagación

Velocidad del Origen (mps)	Rango de Propagación en Metros				
	100m	150m	200m	250m	300m
1	1/1/1	3/3/3	10/10/10	29/27/29	30/29/30
3	3/3/3	3/3/3	10/10/10	24/24/24	30/17/18
5	3/3/3	6/6/6	10/10/10	24/22/24	30/20/22
10	2/2/2	6/4/6	10/10/10	25/23/25	30/28/30
20	2/2/2	6/6/6	10/9/9	24/24/24	30/23/28
30	1/1/1	6/3/3	10/7/7	26/20/22	30/18/19
40	1/1/1	5/2/2	6/2/3	19/17/19	23/22/23
45	3/3/3	12/7/7	4/3/4	13/11/13	23/17/23

En la Tabla VII, se eligió un valor de rango de propagación fijo de 250m, pero se varió la velocidad del origen, así como el número total de nodos. De acuerdo a los resultados obtenidos en estos escenarios, el conteo de nodos fue muy similar a los experimentos llevados a cabo en la Tabla VI, es decir muy acertado para velocidades de hasta 20 mps, con una degradación del conteo para velocidades superiores, siendo el algoritmo II el que hace el mejor conteo.

Tabla VII: Conteo de Nodos cuando se Varía la Velocidad del Origen y el Número Total de Nodos

Velocidad del Origen (mps)	Número Total de Nodos						
	10	15	20	25	30	40	50
1	3/3/3	15/15/15	19/19/19	25/25/25	29/27/29	39/39/39	50/47/50
3	3/3/3	14/14/14	19/8/19	19/19/19	24/24/24	35/35/35	50/45/45
5	3/3/3	14/14/14	12/12/12	17/17/17	24/22/24	35/33/35	50/50/50
10	4/1/3	12/12/12	12/9/9	22/22/22	25/23/25	30/22/23	45/42/45
20	3/3/3	10/6/4	8/8/8	22/21/22	24/24/24	23/23/23	40/38/40
30	3/3/3	12/9/11	9/8/8	18/18/18	26/20/22	32/26/29	48/46/48
40	3/3/3	14/6/6	12/9/12	13/4/4	19/17/19	37/32/28	47/45/45
45	3/2/3	11/11/11	11/11/11	9/9/9	13/11/13	37/21/20	48/40/40

D. Escenarios con Nodos y el Origen Móviles

En esta sección, se reporta y se analiza comparativamente los resultados de las simulaciones realizadas con los algoritmos I y II, para escenarios donde los nodos y el origen tienen movilidad. Al comienzo de las simulaciones, los nodos se sitúan aleatoriamente en un área cuadrada de 800m x 800m, mientras que el origen es ubicado en el centro de la misma. Tanto los nodos como el origen se desplazan según el modelo de movilidad aleatorio RandomWPMobility, con un tiempo de espera de 0s. Asimismo, se seleccionó los valores a continuación para los parámetros: RBCT_TIME=0.2s, INT_BTW_REQ=0.2s y DELAY=0.4s, iniciando el origen el proceso de conteo con un valor de HopLimit igual a 3.

La Tabla VIII representa los resultados de la simulación de escenarios con 30 nodos, donde variamos tanto su velocidad como la del origen, así como su rango de propagación.

Tabla VIII: Conteo de Nodos cuando se Varía la Velocidad de los Nodos y del Origen así como los Valores de Rango del Propagación

Velocidad (mps)	Rango de Propagación en Metros				
	100m	150m	200m	250m	300m
1	5/4/4	11/11/11	18/18/18	29/29/28	30/30/30
3	4/4/4	9/9/9	17/17/17	30/24/24	30/30/30
5	5/4/4	8/8/8	18/18/18	30/26/29	30/26/27
10	9/8/8	15/15/15	24/25/24	29/24/24	30/26/26
20	9/6/8	5/5/5	13/12/12	27/21/27	30/29/29
30	2/1/1	17/5/5	30/19/19	28/20/24	30/19/20
40	5/1/1	17/3/5	28/16/19	30/20/20	30/20/22
45	10/2/2	12/2/2	20/2/3	22/9/17	30/11/15

En la Tabla IX, se eligió un rango de propagación fijo de 250m, pero se varió la velocidad de los nodos y del origen, así como el número total de nodos. Al igual que en los experimentos anteriores (ver Tabla VIII), se obtuvo también un conteo eficaz para velocidades de hasta 20 mps. En tanto que para velocidades más altas, la precisión del conteo se degrada, pero siendo más acertada en el algoritmo II que en el algoritmo I.

Tabla IX: Conteo de Nodos cuando se Varían la Velocidad de los Nodos y el Origen así como también el Número Total de Nodos

Velocidad (mps)	Número Total de Nodos							
	10	15	20	25	30	35	40	50
1	10/10/10	15/15/15	16/16/16	25/25/25	29/29/29	35/35/35	40/40/40	50/50/50
3	8/8/8	15/14/15	17/17/17	25/25/25	30/24/24	35/35/35	40/40/40	50/45/45
5	6/6/6	15/13/15	19/17/17	24/23/23	30/26/29	35/35/35	40/40/40	50/48/48
10	6/6/6	14/11/9	20/20/20	25/24/25	29/24/24	35/30/34	40/40/40	49/48/48
20	5/5/5	11/11/11	10/9/9	25/20/23	24/21/21	31/27/27	34/26/28	49/46/49
30	3/3/3	10/3/4	20/17/17	15/15/15	28/20/20	23/21/21	37/29/33	50/36/47
40	10/9/10	14/10/10	19/12/12	20/3/17	30/20/20	34/22/25	39/20/20	50/21/28
45	10/7/9	15/12/12	19/14/16	23/6/12	22/9/13	35/7/30	40/35/36	50/15/19

VI. CONCLUSIONES Y TRABAJOS FUTUROS

En este artículo, se presenta una nueva versión de un algoritmo para contar nodos con tecnologías inalámbricas, basado en las ideas planteadas en el algoritmo propuesto en [7], donde se toman en cuenta aquellos mensajes retrasados que pudieran ser

enviados por los nodos y por tanto ser también contados. Se hace un estudio comparativo de ambos algoritmos.

Para validar y comparar los algoritmos, se realizaron dos tipos de simulaciones: (1) cuando los nodos y el origen son estacionarios, y (2) cuando la movilidad está involucrada en los experimentos, resultando en un cambio de topología entre los dispositivos. En función de las simulaciones ejecutadas, con valores apropiados de los parámetros, se deduce que para el conteo de nodos sobre escenarios estacionarios y móviles, el algoritmo II es mucho más eficiente que el algoritmo I (propuesto en [7]).

Como trabajo futuro, se tiene la intención de mejorar el nuevo algoritmo para obtener un conteo más eficaz cuando los nodos se mueven a altas velocidades, en base a la posición actual de estos, es decir, se agregaría un GPS a cada nodo. También, se pretende adaptar el algoritmo a redes inalámbricas multicanales, con la finalidad de reducir las colisiones que ocurren entre los mensajes enviados. Finalmente, estamos interesados en el estudio de la aplicación del algoritmo en el contexto vehicular (autopistas, vías urbanas, caminos rurales, parkings, entre otros) utilizando WAVE (Wireless Access in Vehicular Environment) [27], ya que podría ser utilizado en aplicaciones para mejorar la seguridad y la comodidad de los conductores y pasajeros.

AGRADECIMIENTOS

Agradecemos al CDCH-UCV (Consejo de Desarrollo Científico y Humanístico) que en gran parte apoyó esta investigación bajo el número de subvención: PG 03-8066-2011/1.

REFERENCIAS

- [1] E. Kell and E. Mills. *Traffic Detector Handbook*. U.S. Department of Transportation, Federal Highway Administration, 2nd Ed., pp. 1–39. USA, 1990.
- [2] L. Klein. *Sensors Technologies and Data Requirements for ITS*. Artech House Publishers, Norwood, USA, 2001.
- [3] L. Mimbela and L. Klein. *A Summary of Vehicle Detection and Surveillance Technologies used in Intelligent Transportation Systems*. Handbook, Federal Highway Administration, Intelligent Transportation Systems. USA, 2007.
- [4] G. Leduc. *Road Traffic Data: Collection Methods and Applications*. European Commission, Joint Research Center, Institute for Prospective Technological Studies, Seville, Spain. 2008.
- [5] T. Schweizer. *Methods for Counting Pedestrians*, in proceedings of the 6th International Conference on Walking in the 21st Century (Walk21-VI). “Everyday Walking Culture”, Zurich, Switzerland, September 2005.
- [6] J. Rojas. *Conteo de Personas mediante Videocámaras*. Tesis de Maestría en Ciencias (Matemáticas Aplicadas e Industriales). Universidad Autónoma Metropolitana, México City, México, Abril 2013.
- [7] E. Gamess and M. Contreras. *A Proposal for an Algorithm to Count Nodes using Wireless Technologies*. International Journal of High Performance Computing and Networking. In press.
- [8] C.-H. Chen, Y.-C. Chang, and D.-J. Wang. *People Counting System for Getting In/Out of a Bus Based on Video Processing*, in proceedings of the 8th International Conference on Intelligent Systems Design and Applications (ISDA’08), pp. 565–569, November 2008.
- [9] D. Roqueiro and V. Petrushin. *Counting People using Video Cameras*. International Journal of Parallel, Emergent and Distributed Systems, vol. 22, article 3, pp. 193–209, USA, May 2007.
- [10] H.-C. Chen, Y.-C. Chang, N.-J. Li, C.-F. Weng, and W.-J. Wang. *Real-Time People Counting Method with Surveillance Cameras Implemented on Embedded System*, in proceedings of the World Congress on

- Engineering and Computer Science 2013. San Francisco, USA, October 2013.
- [11] P. Lengvenis, R. Simutis, V. Vaitkus, and R. Maskeliunas. *Application of Computer Vision Systems for Passenger Counting in Public Transport*. *Elektronika ir Elektrotechnika*, vol. 19, no. 3, pp. 69–72, March 2013.
- [12] T. Yahiaoui, L. Khoudour, and C. Meurie. *Real-Time Passenger Counting in Buses using Dense Stereovision*, *Journal of Electronic Imaging*, vol. 19, no. 3, July 2010.
- [13] H. Celik, A. Hanjalic, and E. Hendriks. *Towards a Robust Solution to People Counting*, in proceedings of the 2006 IEEE International Conference on Image Processing. Atlanta, Georgia, USA, October 2006.
- [14] G.-G. Lee, H.-K. Kim, J.-Y. Yoon, J.-J. Kim, and J.-J. Kim. *Pedestrian Counting using an IR Line Laser*, in proceedings of the 2008 International Conference on Convergence and Hybrid Information Technology (ICHIT'08), Daejeon, Korea, August 2008.
- [15] M. Lei, D. Lefloch, P. Gouton, and K. Madani. *A Video-Based Real-Time Vehicle Counting System using Adaptive Background Method*, in proceedings of the 4th IEEE International Conference on Signal Image Technology and Internet Based Systems, Bali, Indonesia, November 2008.
- [16] M. Tursun and G. Amrulla. *A Video Based Real-Time Vehicle Counting System using Optimized Virtual Loop Method*, in proceedings of the 2013 International Workshop on Systems, Signal Processing and their Applications (WoSSPA). Algiers, Algeria, May 2013.
- [17] C.-K. Ki. *Moving Object Counting with an Infrared Sensor Network*. Master Thesis in Computer Science and Engineering. Hong Kong University of Science and Technology. Hong Kong, August 2007.
- [18] H. Gros-Desormeaux, P. Hunel, N. Vidot, and E. Stattner. *Acoustic Counting Algorithms for Wireless Sensor Networks*, in proceedings of the 6th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks, pp. 79–84. New York, USA, 2009.
- [19] M. Litzemberger, B. Kohn, G. Gritsch, N. Donath, C. Posch, N. Belbachir, and H. Garn. *Vehicle Counting with an Embedded Traffic Data System using an Optical Transient Sensor*, in proceedings of the 10th International IEEE Conference on Intelligent Transportation Systems (ITSC'07), Seattle, WA, USA, September 2007.
- [20] K. Fuerstenberg, J. Hipp, and A. Liebram. *A Laserscanner for Detailed Traffic Data Collection and Traffic Control*, in proceedings of the 7th World Congress on Intelligent Transport Systems (ITS'00), Turin, Italy, November 2000.
- [21] F. Bu and C.-Y. Chan. *Pedestrian Detection in Transit Bus Application: Sensing Technologies and Safety Solutions*, in proceedings of the 2005 IEEE Intelligent Vehicles Symposium, pp. 100–105, Las Vegas, Nevada, USA, June 2005.
- [22] K. Frerstenberg and U. Lages. *Pedestrian Detection and Classification by Laser Scanners*, in proceedings of the 9th European Engineers Cooperation International Congress, Paris, France, 2003.
- [23] S. Descamps, A. Béchet, X. Descombes, A. Arnaud, and J. Zerubia. *An Automatic Counter for Aerial Images of Aggregations of Large Birds*, *Bird Study*, vol. 58, no. 3, pp. 302–308, 2011.
- [24] B. Sirmacek, M. Wegmann, A.D.P. Cross, J.G.C. Hopcraft, P. Reinartz, and S. Dech. *Automatic Population Counts for Improved Wildlife Management using Aerial Photography*, in proceedings of the 2012 International Congress on Environmental Modeling and Software. Leipzig, Germany. July 2012.
- [25] R. Kissell and S. Nimmo. *A Technique to Estimate White-Tailed Deer *Odocoileus Virginianus* Density using Vertical-Looking Infrared Imagery*, *Wildlife Biology*, vol. 17, no. 1, pp. 85–92, 2011.
- [26] E. Gameess and I. Mahgoub. *A Novel VANET-Based Approach to Determine the Position of the Last Vehicle Waiting at a Traffic Light*, in proceedings of the 2011 International Conference on Wireless Networks (ICWN'11), Las Vegas, Nevada, USA, July 2011.
- [27] *IEEE Trial-Use Standard for Wireless Access in Vehicular Environments (WAVE) - Multi-Channel Operation*. IEEE 1609.4. November 2006.