UNIVERSIDAD POLITECNICA DE VALENCIA

DEPARTAMENTO DE COMUNICACIONES

Máster Universitario en Tecnologías, Sistemas y Redes de Comunicaciones

Diseño de un protocolo para redes tolerantes a retardos: RTaDAP

Autor: Sergio Martínez Tornell

Director 1: Pietro Manzoni

Director 2: Vicente Casares Giner

Fecha de comienzo: 13/6/2011

Lugar de trabajo: Grupo de Redes de Computadores, DISCA

Objetivos – Los objetivos de la tesina son: el estudio de la aplicación de las redes tolerantes a retardos en el campo de las redes vehiculares y el diseño de un protocolo para la recolección de información generada por sensores instalados en vehículos.
Metodología – La metodología utilizada se basa en el estudio de las propuestas anteriores, la identificación de sus puntos flacos y el diseño de un protocolo basado en estas carencias. La validación de dicho protocolo se realizará mediante simulaciones.
Desarrollos teóricos realizados – Estudio detallado e identificación de los principales errores en propuestas anteriores.
Desarrollo de prototipos y trabajo de laboratorio – Implementación y prueba del protocolo en el simulador de redes Ns-3. Comparación entre propuestas anteriores y nuestra propuesta.
Resultados – Los resultados indican que nuestra propuesta tiene un rendimiento superior a las comparadas cuando el número de nodos en la red es muy alto. Sin embargo hemos observado que el rendimiento de nuestro protocolo decrece cuando la red es altamente dispersa y la densidad de nodos es baja.
Líneas futuras – Como lineas de futuro esperamos ahondar en las comparaciones con otras propuestas así como mejorar la integración entre las herramientas utilizadas para la generación de la movilidad y la simulación de redes. También tenemos como objetivo prioritario la mejora del protocolo en el caso de redes con una densidad de nodos baja.
Publicaciones – Esta tesina no ha generado ninguna publicación científica.
Abstract – In last years the European Commission has decided to make an effort in order to improve road traffic security. One of the main lines which this effort is divided in is the Intelligent Transportation System technology, which aims to increase efficient and security of road traffic using new communication techniques. The number of techniques used for this purpose varies from vehicular ad-hoc networks to cellular mobile networks. Inside this wide range of protocols and technologies we have focused in delay tolerant networks and how it can be applied to vehicular networks facing the intermittent connectivity due to rapid changes of its topology. We have studied the proposed protocols and identified the main gaps needed to be filled in them, joining all this gaps and trying to fill most of them we have designed, developed and tested our own DTN protocol based on traffic route topology awareness called Road Topology and Destination Aware Protocol (RTaDAP). Testing results, based on simulations, shown that our protocol behaves better than well known DTN protocols under some conditions but presents some problems that must be solved in future.

Autor: Sergio Martínez Tornell, email: sertinell@gmail.com
Director 1: Pietro Manzoni, email: pmanzoni@disca.upv.es
Director 2: Vicente Casares Giner, email: vcasares@dcom.upv.es
Fecha de entrega: 22-12-11

Índice

Índice de figuras

1. Introducción

Tras el gran despliegue de redes inalámbricas producido en los primeros años de este siglo, y el gran aumento de nodos conectado a internet que nos lleva cada vez más hacia lo que podemos llamar "la internet de las cosas", se ha producido un gran interes en los que se conocen como Intelligent Transport Systems (ITS). ITS puede significar el siguiente salto en lo que a seguridad y eficiencia en los sistemas de transporte se refiere mediante la utilización de diferentes tecnologías de comunicaciones. Una de estas tecnologías en las que se apoya ITS son las redes vehiculares. Estas redes presentan algunas características propias que hacen idónea la aplicación de los principios de las DTN. En esta tesina estudiaremos el funcionamiento de las DTN y presentaremos nuestra propia propuesta para un sistema de recolección de información generada por sensores dispuestos en vehículos.

1.1. Motivación

En este momento las redes vehiculares son un campo de investigación de plena actualidad como demuestra la multitud de conferencias y revistas dedicadas a ellas de manera exclusiva. En nuestra opinión las redes tolerantes a retardos pueden significar un salto en el rendimiento y aplicación de las redes vehiculares ya que esquivan algunos de los principales problemas de éstas, permitiendo la utilización de las mismas aún cuando una utilización clásica basada en la conmutación de paquetes sería imposible.

1.2. Objetivos

Los objetivos de esta tesina son básicamente dos:

- El estudio detallado de los diversos protocolos de encaminamiento para redes DTN propuestos, así como la identificación de sus mayores problemas y fallos.
- El diseño y prueba de un protocolo para redes DTN basado en las conclusiones alcanzadas en el punto anterior.

1.3. Estructura

Esta tesina se encuentra dividida en 8 capítulos, el primero de ellos, en el cual nos encontramos, ofrece una visión del trabajo que se ha realizado. En el segundo capítulo se realiza una introducción a las redes vehiculares y la normativa que se les aplica, para en el tercero entrar de manera más profunda en el tema principal de esta tesina, las redes tolerantes a retardos (DTN), en este tercer capítulo se enumerarán y describirán algunos de los protocolos recientemente propuestos en el mundo de las DTN para finalizar explicando nuestra propuesta. En el cuarto capítulo se describe la metodología de simulación utilizada para comprobar el rendimiento del protocolo deseado. Posteriormente, en el quinto capítulo se exponen y analizan los resultados obtenidos. En el siguiente capítulo se presentan las conclusiones de este trabajo. En el séptimo y penúltimo capítulo realizamos una pequeña introducción sobre el trabajo que estamos llevando a cabo en este momento

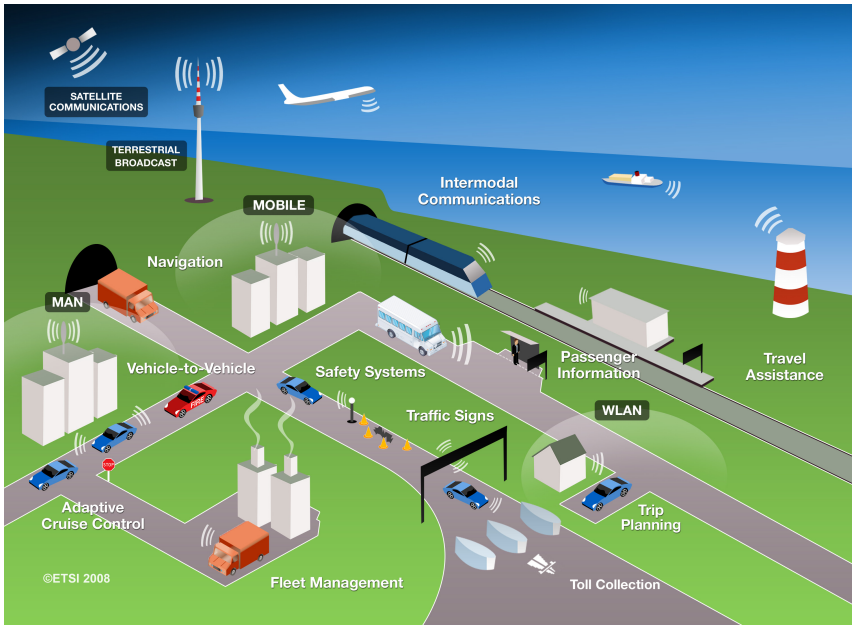


Figura 1: Sistemas ITS

y lo que será nuestro trabajo en los próximos meses. Finalmente el capítulo octavo incluye los agradecimientos a las instituciones que han hecho posible con sus fondos la ejecución de esta tesina.

2. Redes Vehiculares

Las redes vehiculares forman parte de las tecnologías que en el futuro se espera ayuden a desarrollar el ITS. ITS espera aumentar la seguridad y la eficiencia en el desplazamiento tanto de mercancías como de personas nutriéndose de diferentes tecnologías, la figura 1 ilustra perfectamente la situación en un mundo hiperconectado a la que ITS hace referencia. Dentro de este marco esta tesina se enfoca en las redes vehiculares, por red vehicular se entiende aquella en la que sus nodos se sitúan en vehículos. Dentro de las redes vehiculares podemos diferenciar entre redes Vehicle to Vehicle (V2V) y redes Vehicle to Infraestructure (V2I), la diferencia entre ambas es que en el primer caso la comunicación sucede únicamente entre los vehículos de la red mientras que en la segunda existen ciertos nodos conocidos como dispositivos Road Side Unit (RSU) situados al margen de la carretera, sin movimiento. En esta tesina estudiamos una situación combinada donde existe comunicación tanto entre los nodos como con los dispositivos RSU. A lo largo de esta sección expondremos las características principales de este tipo de redes, así como la normativa bajo la que se deben desplegar.

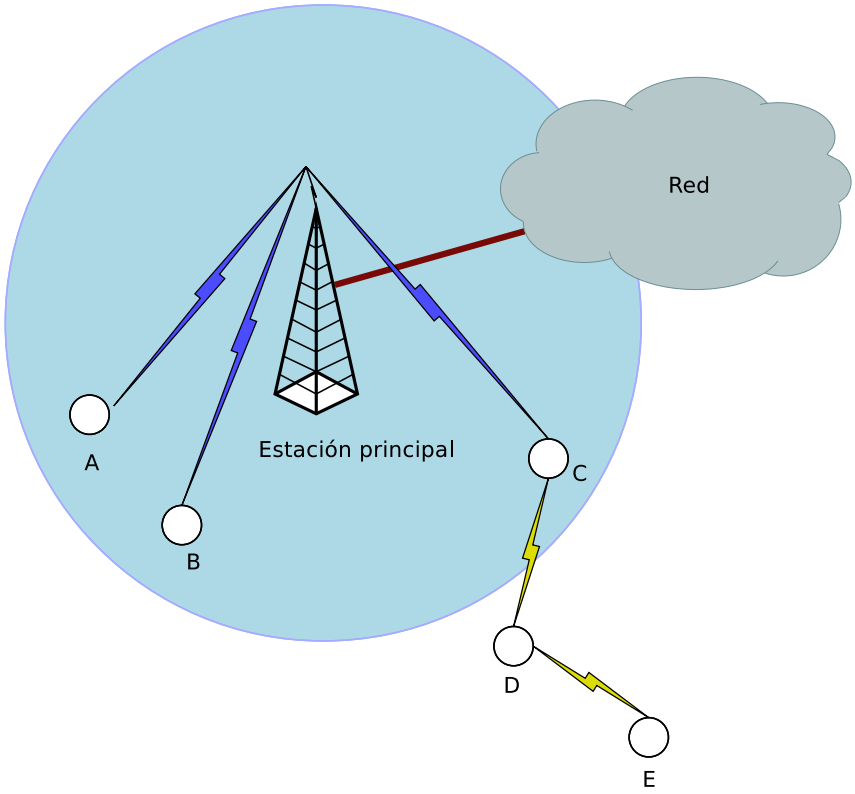


Figura 2: Ejemplo de red mallada.

2.1. Características

Las redes vehiculares presentan similitudes con otro tipo de redes conocido como redes *malladas*, en una red mallada, además de la comunicación típica que se produce dentro de las redes de infraestructura entre los nodos y el punto de acceso o estación principal, existe también comunicación directa entre los nodos que forman la red, permitiendo a aquellos nodos que no están dentro de la cobertura de la estación principal comunicarse con esta mediante comunicaciones multisalto, sin embargo, en las redes malladas los nodos no presentan movimiento y su posición suele ser fija. La figura 2 muestra el esquema típico de una red mallada donde los nodos E y D deben comunicarse entre ellos para poder alcanzar la estación principal. Además las redes vehiculares también presentan muchas similitudes con las redes Mobile Ad-hoc NETwork (MANET), en una red MANET normalmente no existe infraestructura y los nodos utilizan enlaces directos entre ellos para comunicarse, la movilidad de las redes MANET hace que éstas tengan que recalcular la ruta a seguir por los paquetes de manera frecuente.

Existen varios estándares para redes malladas y redes MANET como 802.11s [3], sin embargo las redes vehiculares presentan ciertas características propias que las hacen diferentes de estos tipos de redes e impiden a los protocolos de encaminamiento típicos de redes malladas y MANET como protocolos reactivos tipo Ad hoc On-Demand Distance Vector (AODV) [19], los cuáles determinan la ruta de un paquete en el momento en el que

se necesita, o protocolos proactivos como Optimized Link State Routing (OLSR) [15], los cuáles determinan la ruta de manera proactiva antes de que esta sea necesaria, funcionar de manera satisfactoria. Estas características son:

- Gran movilidad: En las redes malladas la movilidad es prácticamente nula y los cambios se producen por la conexión y desconexión de los nodos y en las redes MANET la movilidad de los nodos es relativamente baja (se suelen considerar velocidades similares a las de una persona a pie), en las redes vehiculares la velocidad a la que se desplazan los nodos produce grandes variaciones en la topología de la red. Estas variaciones impiden el correcto funcionamiento de los protocolos de encaminamiento.

- Número de nodos: Normalmente, en redes MANET el tamaño de la red se limita a unas decenas o centenares de nodos y la extensión se mantiene en el orden de los centenares de m^2 , por el contrario en una red vehicular el número de nodos puede alcanzar el millar y su extensión estar en el orden de decenas de km^2. Este tamaño complica el encaminamiento de paquetes basado en las direcciones de origen y destino, por lo complicado que resulta recalcular la ruta y la falta de tiempo para ello debido a la gran movilidad.

- Particionamiento de la red: Las dos características anteriores junto con la tendencia de los vehículos a agruparse en *clusters* producen que las redes vehiculares se encuentren normalmente muy particionadas y divididas.

Estas características producen que una gran cantidad de paquetes sean descartados por la falta de una ruta por la que encaminar el paquete. Para intentar minimizar este problema se ha propuesto recientemente la utilización de las redes tolerantes a retardos, conocidas como DTN, que permite que un paquete sea entregado incluso cuando jamás existe una ruta entre origen y destino. En las posteriores secciones nos centraremos en este tipo de redes.

2.2. Marco regulador en Europa

En Europa las emisiones de señal correspondientes a las redes vehiculares se encuentran reguladas por la directiva emitida por el European Communication Committee (ECC) [11] la cuál especifica, basándose en diversos estudios realizados por dicho comité, que la banda de frecuencia de operación para ITS abarcará desde los 5855 Mhz hasta los 5925 Mhz y que la densidad de potencia máxima de transmisión será de 23 dBm/Mhz dando un total de 33dBm cuando el ancho de banda es de 10 Mhz. A nivel de capa MAC el estándar utilizado es el 802.11p, este estándar es básicamente una adaptación del conocido 802.11a con una reducción del ancho de banda de los 20Mhz a 10Mhz para evitar los efectos del ensanchado de símbolo producido por la alta velocidad de los nodos. También se ha reducido la velocidad de transmisión de la red de los 54 Mbps en 802.11a a los 6Mbps para disminuir la probabilidad de error y aumentar así el rango de cobertura. Todos estos datos se pueden encontrar en los documentos [2] y [1].

3. Redes DTN

Este capítulo esta dividido en 3 secciones. En la primera sección se realizará una introducción a las DTN, sus características y paradigmas. Posteriormente analizaremos que pueden aportar las DTN a los protocolos típicamente utilizados en redes vehiculares que se describieron en el capítulo anterior 2. En el tercer apartado introduciremos algunas soluciones aportadas previamente por otros autores. Finalmente expondremos algunos de los problemas de los que en nuestra opinión adolecen dichas propuestas.

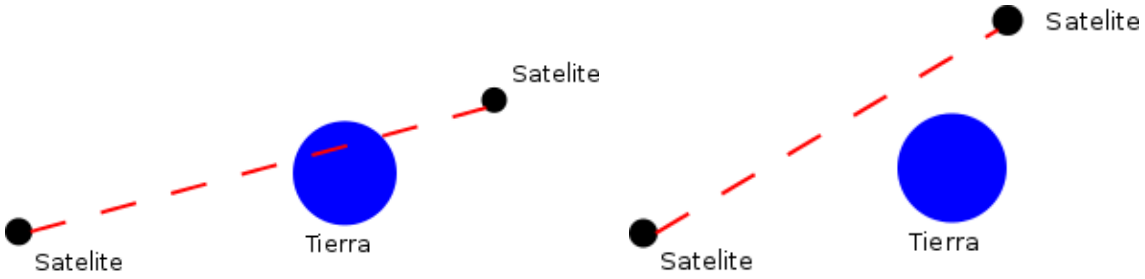
3.1. Características de las DTN

En Mayo del 2001 el grupo de especial interés para la red interplanetaria, con siglas Interplanetary Network Special Interest Group (IPNSIP), englobado dentro de Internet Society (ISOC), publica el documento *Delay-Tolerant Network Architecture:The Evolving Interplanetary Internet* [8]. En este documento se define la estructura de una red similar a internet desplegada entre nodos donde los retardos son grandes y una ruta entre los 2 extremos de una comunicación podría jamás llegar a existir. El mecanismo utilizado para proveer esta conexión es el de almacenamiento y reenvío, donde un nodo almacenará un mensaje el tiempo necesario hasta que sea posible enviarlo al nodo siguiente. Más adelante en el año 2007 el Delay Tolerant Network Research Group (DTNRG), perteneciente al Internet Engineering Task Force (IETF), publicó la RFC *Delay-Tolerant Networking Architecture* [7], donde la estructura de la red se generalizaba para su aplicación en otros tipos de redes como las redes de sensores, redes vehiculares o redes acústicas subacuáticas.

Como se ha dicho en el párrafo anterior el mecanismo básico de funcionamiento de las redes DTN es el de almacenamiento y reenvío. La diferencia entre las distintas redes DTN se encuentra en el algoritmo utilizado para determinar el siguiente nodo en la ruta que seguirá un paquete. En el siguiente apartado introduciremos algunos de los protocolos propuestos por los investigadores. Antes creemos que es importante realizar y exponer una clasificación de las redes DTN basada en la movilidad de los nodos que forman parte de la red;

- Redes con movilidad determinista: En este tipo de redes los nodos tienen un movimiento fijo y predecible en el tiempo. Con estos datos los nodos pueden calcular cuando una transmisión podrá llevarse a cabo y actuar en consecuencia. El ejemplo más claro de una red DTN con movilidad determinista es la red que formarían los satélites, sondas y estaciones espaciales desplegados por la NASA, donde las oportunidades de transmisión vienen marcadas por el tiempo en el que existe visión directa entre 2 nodos. En la figura 3 hay un ejemplo de red DTN interplanetaria. En este ejemplo los mensajes presentes en la cola en el momento 3(a) serán enviados más tarde durante la oportunidad de transmisión representada en 3(b).

- Redes con movilidad aleatoria: En este tipo de redes los nodos tienen una movilidad aleatoria o al menos autónoma. En este tipo de redes se intenta analizar el movimiento de los nodos para determinar la ruta óptima, esta decisión siempre es estadística e intenta maximizar la probabilidad de alcanzar el destino en el menor tiempo posible.



(a) Enlace no disponible. (b) Enlace disponible.

Figura 3: Ejemplo de una red DTN con movimiento determinista.

Ejemplos de este tipo de redes son el que podrían formar los portátiles de los estudiantes en un campus universitario o el que formarían múltiples sensores acoplados a una población animal. La figura 4 presenta un ejemplo de este tipo de redes.

- Redes con movilidad sujeta a restricciones: En este tipo de redes el movimiento de los nodos, aunque autónomo, esta sujeto a restricciones debidas al entorno. El ejemplo más claro de una red de este tipo son las redes vehiculares, donde el movimiento de los coches esta limitado a la red de carreteras. La figura 1 expuesta en la sección 2 muestra un escenario típico donde la movilidad de los nodos se encuentra restringida.

3.2. Aplicaciones de las DTN en Redes Vehiculares

Como se expuso anteriormente en 2 las redes vehiculares presentan una movilidad mucho mayor que las redes MANET, esta alta movilidad se traduce en que la duración de los enlaces y el tiempo de vida de las rutas suele ser muy corto, de hecho, es muy probable que la ruta entre los dos extremos de una comunicación no llegue a existir jamás, haciendo inviable la utilización de protocolos como OLSR, AODV o cualquier otro protocolo diseñado para obtener una ruta entre origen y destino. Este problema se acentúa cuando la densidad de nodos en la red es baja y el despliegue de infraestructura es altamente complicado. Estas características hacen el paradigma de las redes DTN, "almacenamiento y reenvío", resulte perfecto desde nuestro punto de vista para su aplicación en las redes vehiculares. Teniendo en cuenta la clasificación de las redes DTN realizada en el punto anterior 3.1, ahora procederemos a enumerar y analizar algunas de las propuestas de los investigadores con más impacto en los últimos años.

1. Epidemic Protocol: En este protocolo los nodos anuncian periódicamente su presencia al resto de nodos, cuando dos nodos entran en contacto estos intercambian sendas listas que contienen el identificador de cada uno de los mensajes presentes en sus respectivos buffers. Tras determinar que mensajes es necesario intercambiar se establece una conexión y se intercambian los mensajes. Dado un número de recursos suficientes y una movilidad también suficiente este protocolo presenta el mínimo tiempo de entrega para los mensajes así como el mínimo porcentaje de paquetes perdidos[22].

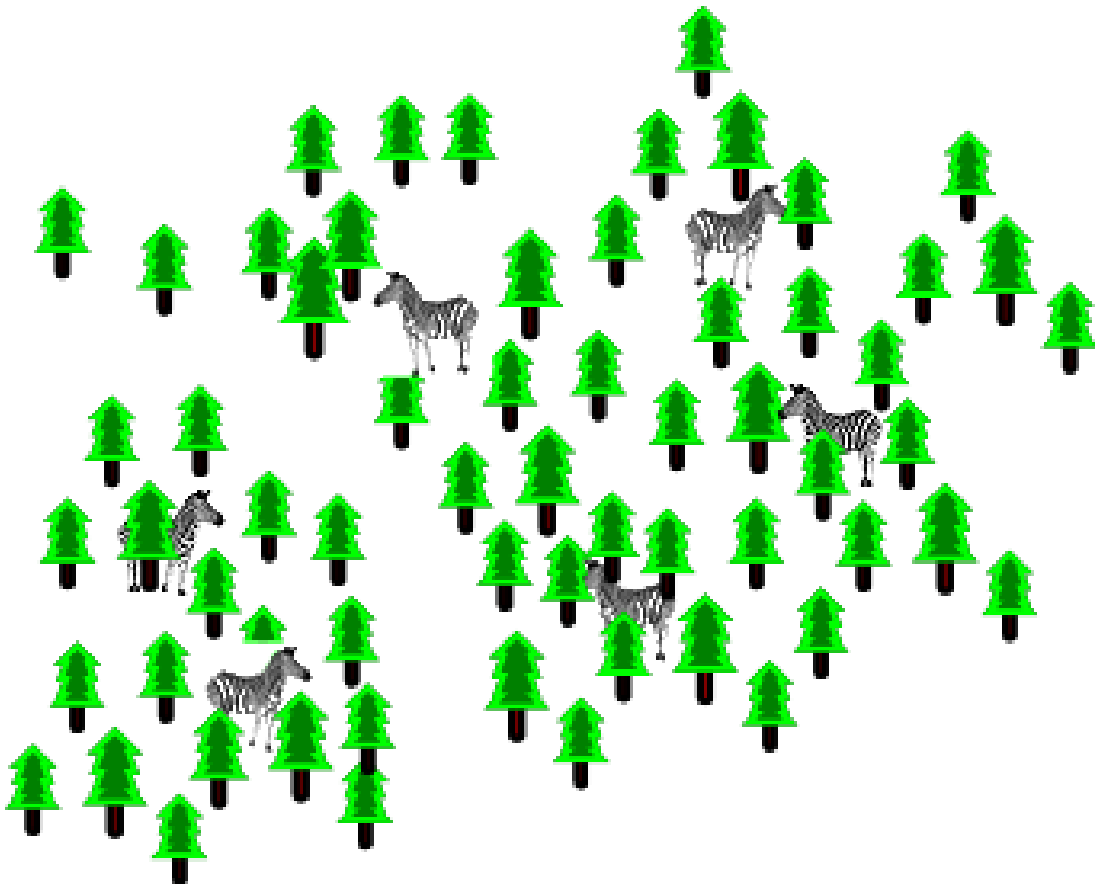


Figura 4: Ejemplo de una red DTN con movimiento aleatorio, cada zebra tiene un sensor.

2. PRoPHET: En este protocolo la elección del siguiente nodo en la ruta de un mensaje se realiza cuando el nodo portador del mensaje se encuentra con otro nodo. En ese momento el portador decide en función de la probabilidad de que el nuevo nodo encuentre al destino del mensaje si le enviará el mensaje. En el artículo donde se publicó este protocolo se deja a criterio de la aplicación la utilización de varias copias por mensaje o el umbral de probabilidad que un nodo debe superar para que se le envíe dicho mensaje [18].

3. PRoPHET+: A la definición de probabilidad de contacto con el destino del mensaje definida en PRoPHET este protocolo añade cuatro funciones más dependientes del espacio en buffer, ancho de banda, energía restante del nodo y la popularidad del nodo respectivamente, y define una función ponderada en función de estas cuatro para definir la probabilidad de contacto [14].

4. Spray and Wait: Este protocolo funciona de la misma manera que el Epidemic Protocol pero genera un número limitado de copias de cada mensaje [21].

5. Spray and Wait basado en probabilidad de entrega: Funciona igual que *Spray and Wait* pero la selección de los nodos a los que se les envía una copia del mensaje se

realiza en función de la probabilidad de entrega para cada uno de los nodos destino. Esta probabilidad se calcula de manera similar a como se realiza en PRoPHET [23].

6. TrafRoute: Este protocolo clasifica el tipo de tráfico unicast en dos clases, *inter-dominio* y *intra-dominio*, para ello define el término *dominio* como aquellos nodos que se encuentran dentro de la cobertura del mismo punto de acceso conectado al núcleo de la red. Así, el tráfico *inter-dominio* se produce entre los nodos situados en distintos dominios y el tráfico *intra-dominio*, entre aquellos situados en dominios distintos. Para las comunicaciones *intra-dominio* utiliza cualquier protocolo de encaminamiento clásico para MANET como puede ser OLSR o AODV, las comunicaciones *inter-dominio* se realizan a través del núcleo de la red, utilizando el punto de acceso [13].

7. Fastest Ferry Routing in DTN-enabled Vehicular Ad-Hoc (FFRDV): En este protocolo el mapa se divide en sectores. Cuando el portador de un mensaje cambia de sector envía un mensaje *Hello* anunciando su presencia, el resto de nodos cuando reciben este mensaje responden con su velocidad y dirección. El mensaje se envía al nodo que se mueve a mayor velocidad en dirección al destino, en caso de no encontrar ningún nodo con una velocidad superior a la suya el portador guarda el mensaje hasta el siguiente sector [24].

8. Context DTN (C-DTN): En este protocolo los mensajes tienen una dirección de destino basada en contexto, esto es, los mensajes no se dirigen hacia un nodo sino hacia entidades más generales, por ejemplo, un cruce, un área amplia, a todos los coches que se mueven en cierta dirección, etc [10].

9. DIrection based Geographic routing (DIG): En este protocolo se utiliza la dirección de los nodos para elegir como siguiente nodo aquel cuya dirección más se aproxima a la posición del nodo de destino [17].

10. GeoDTN+Nav: Este protocolo funciona mediante 2 modos distintos, primero realiza un encaminamiento clásico mediante *GeoRouting* y en caso de detectar que no existe una ruta hasta el nodo destino cambia a modo DTN, durante el funcionamiento en modo DTN el encaminamiento se realiza en función de la probabilidad para alcanzar el destino por parte de cada uno de los nodos [16].

3.3. Problemas encontrados en las soluciones propuestas

Los protocolos descritos en el punto anterior presentan varios problemas cuando se aplican a redes vehiculares, en este apartado describiremos y analizaremos algunos de los más importantes.

3.3.1. Malgasto de recursos

Los primeros protocolos que aplicaron el paradigma de almacenamiento y reenvío (1, 4 y 5) distribuyen varias copias del mismo mensaje por la red. Esta redundancia, que puede

ser útil para asegurar el la entrega del mensaje, conlleva un malgasto de recursos.

3.3.2. Envío al nodo más lejano

Algunos de los protocolos descritos en el apartado anterior (10 y 8) basan la elección del siguiente nodo en la distancia de este con el destino. Esto deriva en que el siguiente nodo será siempre el más lejano al nodo actual, como bien es sabido la probabilidad de recepción del mensaje disminuye con la distancia, por lo que la elección del nodo más lejano deriva en un exceso de transmisiones no satisfactorias malgastando gran parte de los recursos.

3.3.3. Dependencia de infraestructura

Es bastante común por parte de algunos autores asumir la existencia de infraestructura de telecomunicaciones desplegada de manera paralela a la red de carreteras. Esta asunción se convierte en un problema cuando el protocolo es incapaz de funcionar cuando la infraestructura no se encuentra disponible como es el caso de 6.

3.3.4. Uso incompleto de la información disponible

Cada vez los dispositivos tipo GPS instalados en vehículos son más comunes, estos dispositivos proveen a los nodos de información sobre la topología de la red de carreteras. De los protocolos descritos anteriormente ninguno de ellos hace uso de una de las características principales de las redes vehiculares, el movimiento de los nodos esta condicionado por la red de carreteras existente. Utilizando esta información se podrían tomar decisiones de encaminamiento mucho más acertadas.

3.4. RTaDAP

Ante los problemas anteriormente expuestos nosotros hemos propuesto y desarrollado el protocolo RTaDAP. RTaDAP es un protocolo diseñado para la recolección de datos desde sensores vehiculares (ie. Sensores desplegados en vehículos). Los datos recogidos por los sensores son encaminados mediante técnicas de DTN hacia ciertos nodos llamados *sumideros* dispuestos estratégicamente en el margen de la carretera, o lo que es lo mismo, todos los mensaje de nuestra red están dirigidos hacía cualquiera de los sumideros. La figura 5 muestra como la información generada por un sensor de la red es transportada hacía el sumidero.

Nuestro protocolo RTaDAP necesita contar con cierta información sobre la ruta que seguirán los distintos nodos de la red, así como la localización de los sumideros. Esta información sobre la ruta se puede obtener de diversas maneras con diferentes grados de confianza, a través de un dispositivo GPS acoplado al vehículo (coche privado), mediante una ruta estática previamente cargada y totalmente determinista (autobuses o trenes), etc. El grado de fiabilidad de esta ruta oscilará entre 0 y 1, siendo 1 para una ruta totalmente determinada y 0 cuando carecemos totalmente de información (este caso no debería darse nunca). Las ruta definida para un coche particular que se dirige al garaje

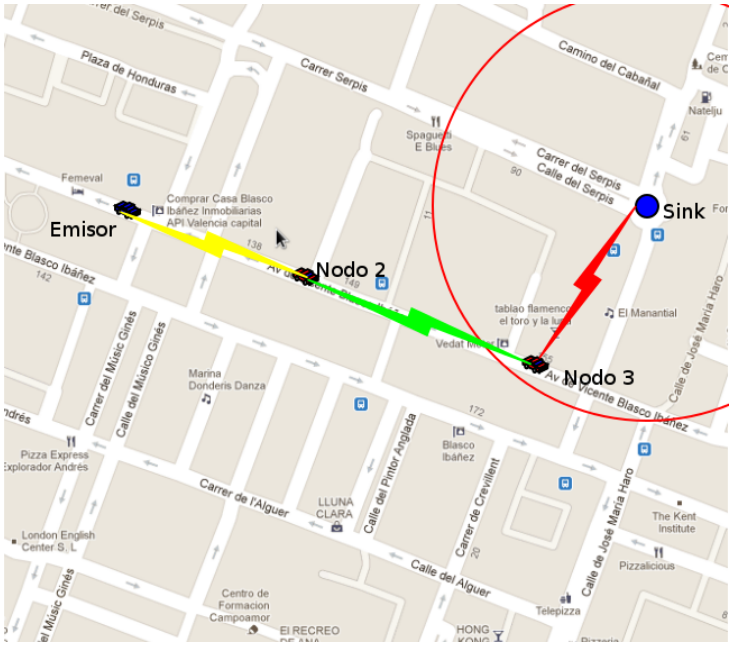


Figura 5: Ejemplo de encaminamiento de un mensaje hacia el sumidero.

o para un autobús serían rutas con gran fiabilidad, mientras que la ruta definida por un taxista donde solo especifica el destino sería una ruta con fiabilidad media, una ruta de baja fiabilidad sería aquella que sigue un vehículo con GPS pero sin destino fijado, en este ultimo caso la única información con la que contamos es la calle actual en la que se encuentra el vehículo.

3.4.1. Caraterísticas

Tras analizar el resto de propuestas y los problemas que hemos encontrado en ellas hemos diseñado RTaDAP con las siguientes características:

- Redundancia: Para intentar minimizar las pérdidas cada mensaje es subdividido en un número determinado de *chunks* o partes. A este número de partes se le añadirá cierto porcentaje de redundancia utilizando códigos Foward Error Correction (FEC) de tal forma que si un paquete es dividido en N partes, se generan $N + M$ partes y el mensaje podrá ser decodificado tan pronto como se reciban N partes de las $N + M$. Este sistema permite cierto grado de protección frente a pérdidas a pesar del esquema de copia única utilizado en RTaDAP.

- Tipos de nodos: Nosotros definimos dos tipos de nodos, *custodian* y *candidate*, un nodo *custodian* es aquel que ha sido encargado de enviar mensajes, un nodo *candidate* es todo nodo presente en la red que es candidato a recibir un mensaje para su posterior entrega a un sumidero. El protocolo asegura que un nodo *custodian* no eliminará un mensaje de su *buffer* hasta que otro nodo haya aceptado convertirse en *custodian* de dicho mensaje.

- Multi interfaces: En RTaDAP cada nodo cuenta con 2 interfaces inalámbricas, una interfaz que sigue el estándar 802.11p, utilizada para la comunicación entre nodos de la red *candidates* y *custodians*, y una interfaz 802.11g, utilizada para la comunicación nodo-sumidero.

- Mecanismo reactivo: Tan solo los nodos *custodian* anuncian su presencia a los nodos vecinos, los posibles nodos *candidate* responderán a este anuncio cuando sea conveniente. De esta manera se minimiza la utilización de recursos.

- Decisión de encaminamiento: En RTaDAP la decisión de encaminamiento se realiza de acuerdo a una función que depende de:

  1. t: El tiempo hasta encontrar un sumidero. Para dar más peso a las pequeñas diferencias entre tiempos pequeños y restar importancia a las pequeñas diferencias entre tiempos grades definimos el parámetro T que sigue la siguiente función:

  $$T = \frac{log(t+1)}{log3600}$$

  Ya que el $DtnIndex$ será inversamente proporcional a t, T debe ser creciente. En el numerador ($log(t+1)$) es necesario añadir una unidad a t ya que el logaritmo de 0 no esta definido. Por otro lado, hemos considerado que para tiempo superiores a una hora las diferencias en retardo no compensan la transmisión de un mensaje, por lo que son consideradas iguales, por lo tanto, hemos elegido 3600 como valor máximo de t. Esto nos permite normalizar T dividiendo el numerador por el logaritmo de 3600. La figura 6 muestra los valores tomados por T con respecto a t.

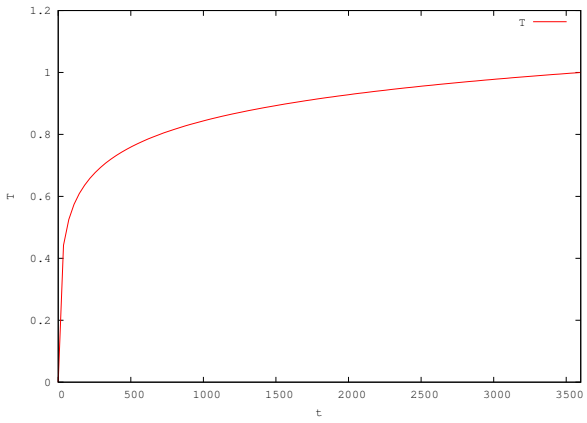


Figura 6: Variación de T con respecto de t.

  2. q: El cociente entre los datos contenidos en el *buffer* y la cantidad que se estima podrá ser entregada al siguiente sumidero presente en la ruta del nodo. Para dar más peso a los valores en los que este cociente es cercano a 0, así como para

aumentar las diferencias entre valores altos de este valor definimos y utilizamos el parámetro Q que sigue la siguiente función:

$$Q = max[\frac{log(50 * (1 - q))}{log(50)}, 0]$$

Tal y como se expone en la ecuación 1, el $DtnIndex$ será proporcional a Q por lo que Q debe ser decreciente con q, esto es, cuanto menor sea el cociente entre los datos contenidos en el buffer y los datos que podrán ser enviados al punto de acceso, mayor será Q. Q queda definida tan solo entre 0 y 1. El factor de 50 modela la curva descrita por la función Q cuando q crece, cuanto más grande es el valor de este factor, más penaliza a los valores cercanos a 1 de q. La figura 7 muestra los valores tomados por Q con respecto a q.

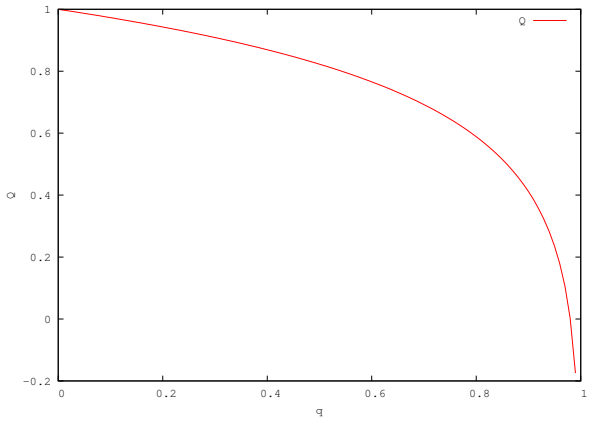


Figura 7: Variación de Q con respecto de q.

3. P:La probabilidad de encontrar al sumidero más cercano. Esta probabilidad depende de la fiabilidad de la ruta del nodo, así como de la distancia al sumidero, obviamente, a mayor distancia, mayor probabilidad de que el vehículo cambie de ruta y no encuentre al sumidero.

La ecuación 1 muestra como se calcula el que hemos llamado $DtnIndex$ en función de los parámetros definidos anteriormente.

$$DtnIndex = \frac{P^2}{T} * Q \tag{1}$$

- Distancia máxima de envío y estimación de la posición: Si atendemos a los parámetros utilizados para obtener la función que determina el siguiente nodo en la ruta de los paquetes vemos que tanto el tiempo necesario para alcanzar el sumidero como la probabilidad de alcanzarlo están estrechamente ligados con la distancia, a menor distancia menor tiempo y mayor probabilidad, tendiendo RTaDAP a encaminar los mensajes a través del nodo más lejano, cayendo así en el mismo error que

algunos de los protocolos propuestos con anterioridad. Para evitar el aumento de las transmisiones erróneas y el malgasto de recursos que conllevan hemos decidido fijar una distancia máxima de transmisión. Los nodos con una distancia superior a este parámetro serán descartados como nodos *candidate*. La posición y la velocidad de cada uno de los nodos es incluida en prácticamente todos los mensajes intercambiados por lo que el nodo *custodian* puede realizar una estimación de la posición de los nodos *candidate* bastante fiable.

- Mensajes broadcast: Todos los mensajes excepto los utilizados para intercambiar mensajes son enviados en modo broadcast, de esta manera un *custodian* podrá aprovechar las respuestas enviadas hacía otro nodo *custodian* vecino para encontrar al mejor *candidate*. Además de aprovechar los mensajes dirigidos a nodos vecinos, también se utiliza esta característica para omitir mensajes innecesarios como podría ser anunciarse justo después de que se anuncie un vecino.

- Anti efecto Ping-Pong: Dado que el DtnIndex depende de los paquetes contenidos en el buffer de los nodos es muy fácil que se de el caso en el que un mensaje produce un cambio suficiente en el $DtnIndex$ como para que el nodo que acaba de transmitir el mensaje pase a convertirse en aquel con un $DtnIndex$ más alto de los disponibles, esto provocaría un efecto Ping-Pong entre los dos nodos, intercambiándose el mensaje continuamente. Para evitar este problema hemos añadido un parámetro que penaliza las transmisiones, este parámetro esta entre 0 y 1, y el $DtnIndex$ de los nodos *candidates* debe ser multiplicado por el mismo antes de ser comparado con el $DtnIndex$ local.

- Protección anti pérdidas: A pesar de que RTaDAP asegura que un mensaje no va a ser eliminado por su nodo *custodian* hasta que no se haya confirmado la aceptación por otro nodo de la custodia del mensaje, es posible que el nodo *custodian* de un mensaje cambie su estado a "inoperativo" de manera repentina, produciéndose así la perdida de los paquetes contenidos en su buffer. Para intentar minimizar esta pérdida los mensajes generados por nuestros sensores son divididos en N paquetes a los que se les añaden M paquetes con datos de redundancia, existiendo así en la red M+N fragmentos de un mensaje, pero siendo posible su decodificación con tan sólo N mensajes.

3.5. Descripción del protocolo

En esta sección vamos a describir de manera detallada el protocolo que hemos diseñado. El diseño se ha realizado siguiendo las pautas descritas en el apartado anterior.

3.5.1. Tipos de paquetes

Primero describiremos los distintos tipos de mensajes utilizados en RTaDAP. La figura 8 resume los campos de los distintos tipos de mensajes descritos a continuación.

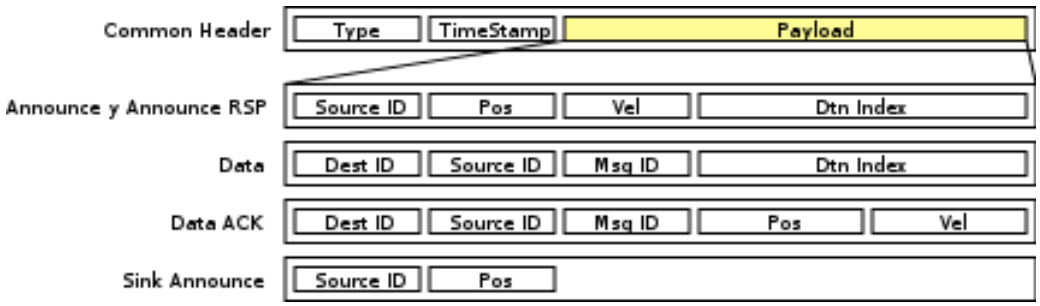


Figura 8: Formato de los paquetes.

Cabecera DTN La cabecera común a todos los paquetes DTN generados en RTaDAP contiene los siguientes datos, los paquetes utilizados en RTaDAP están encapsulados dentro de esta cabecera:

1. Type: Un byte que contiene el subtipo de paquete encapsulado. Los valores que puede tomar son:
   - Announce: Mensaje enviado por un nodo *custodian* cuando tiene paquetes en su cola para enviar.
   - Announce Response: Mensaje enviado por un nodo *candidate* como respuesta a la recepción de un *Announce*.
   - Data: Mensaje utilizado durante un intercambio de paquetes.
   - Ack: Mensaje utilizado para confirmar la recepción de un mensaje *Data*
2. Timestamp: Un entero de 64 bits que indica el momento en el que el paquete fue generado.

Mensajes *Announce* y *Announce Response* Estos mensajes contienen los siguientes datos y van siempre dirigidos a la dirección de broadcast:

1. ID: Dirección DTN del emisor única dentro de la red. Un nodo tan sólo puede tener una única dirección DTN, independientemente del número de interfaces disponibles.
2. Posición: Coordenadas X,Y con la posición del emisor.
3. Velocidad: Valores X,Y de la velocidad del nodo emisor.
4. DtnIndex: Valor de la función descrita en 1 para el nodo emisor.

Mensajes *Data* Estos mensajes van dirigidos de manera unicast al nodo destino y contienen los siguientes datos.

1. ID Destino: Dirección DTN del nodo al que se dirige el paquete.
2. ID Origen: Dirección DTN del nodo que transmite el paquete.

3. Message ID: Este campo identifica de manera unívoca un mensaje enviado por un determinado nodo.

4. N: Número de paquetes encapsulados en el mensaje. Para simplificar la implementación los paquetes tienen un tamaño fijo.

Mensajes ACK Estos mensajes se envían de manera unicast y contienen los siguientes campos:

1. ID Destino: Dirección DTN del nodo al que se dirige el paquete.

2. ID Origen: Dirección DTN del nodo que transmite el paquete.

3. Message ID: Este campo identifica de manera unívoca un mensaje enviado por un determinado nodo.

4. Posición: Coordenadas X,Y con la posición del emisor.

5. Velocidad: Valores X,Y de la velocidad del nodo emisor.

3.5.2. Funcionamiento de los nodos

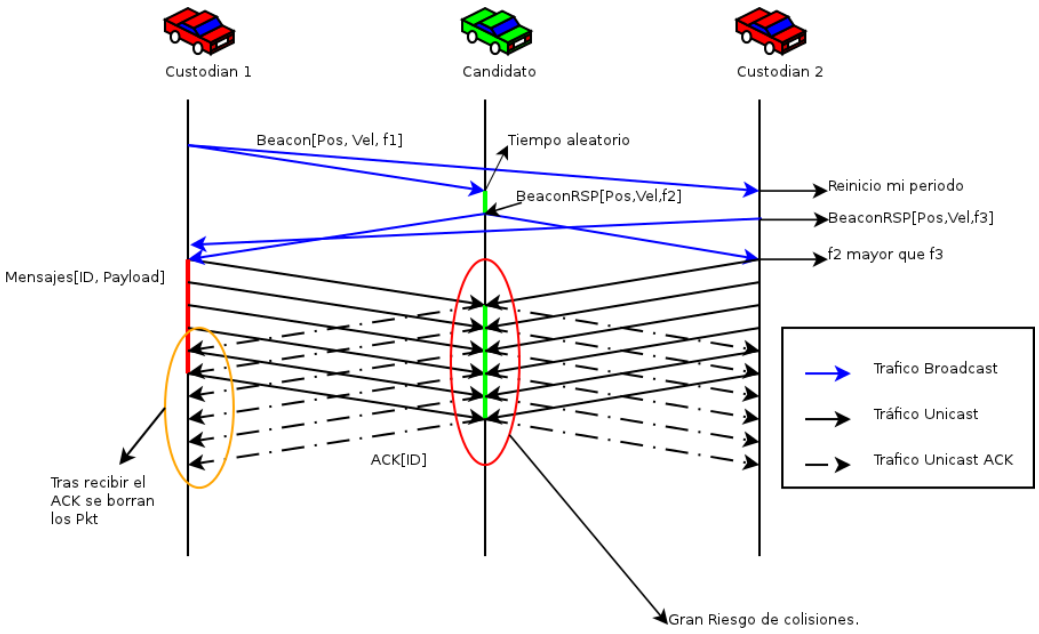


Figura 9: Esquema de comunicación entre nodos.

Para simplificar la definición del funcionamiento de los nodos lo hemos dividido en tres entidades diferentes, estas tres entidades son independientes entre sí. La primera de ellas es el generador de paquetes, esta entidad se encarga de calcular el número necesario de paquetes para enviar un mensaje, añadir los paquetes de redundancia necesarios, y

enviarlos a otra entidad llamada *custodian*. Un nodo *custodian* es aquel que contiene algún paquete en su cola, estos paquetes pueden haber sido generados por el generador de paquetes o por otra entidad llamada *candidate*. En RTaDAP todos los nodos son *candidate*, incluyendo aquellos que tienen paquetes almacenados en su cola(*custodians*). A continuación explicaremos de manera detallada el funcionamiento de estas tres entidades.

Generador de paquetes: El generador de paquetes es el más sencillo de todos, cuando se genera un mensaje calcula cuantos paquetes son necesarios para enviarlo, lo divide, y crea los paquetes de redundancia necesarios. Tras esto, envía todos los paquetes hacía el *custodian*.

***Custodian*:** Cuando un nodo recibe un paquete se convierte en *custodian*, tras este paso el funcionamiento del nodo es el siguiente:

1. Comienza a emitir periódicamente mensajes tipo *Announce*. Tras este anuncio espera la respuesta en forma de mensajes *Announce Resp*.
2. Cuando se recibe un *Announce Resp* se apunta el nodo *candidate* que envió el mensaje en una lista, y tras un tiempo de espera se inicia una transmisión con el mejor de los nodos *candidates* disponibles.
3. Para minimizar las transmisiones, si se recibe un *Announce* desde otro *custodian* considerado como vecino, y su $DtnIndex$ es peor que el del nodo, se omite el siguiente mensaje *Announce*.
4. Durante la fase de transmisión se envían paquetes encapsulados en el payload de mensajes *Data*, el tamaño de los mensajes *Data* esta limitado por el MTU de la red.
5. Cuando se recibe un *Data ACK* se procede a eliminar los paquetes confirmados del buffer.

La figura 10 muestra un flujograma con el funcionamiento del nodo de manera simplificada.

***Candidate*:** Los candidatos funcionan de la siguiente manera:

1. Se mantienen en modo pasivo escuchando el medio hasta recibir un mensaje *Announce* de un *custodian*.
2. Tras recibir un *Announce* esperan un tiempo aleatorio y siempre que el $DtnIndex$ local es mayor que el $DtnIndex$ del nodo emisor se responde con un mensaje *Announce Resp*.
3. Si otro *candidate* contesta al *Announce* antes, y en caso de que su $DtnIndex$ sea mejor que el local, se aborta la transmisión del *Announce Resp*.

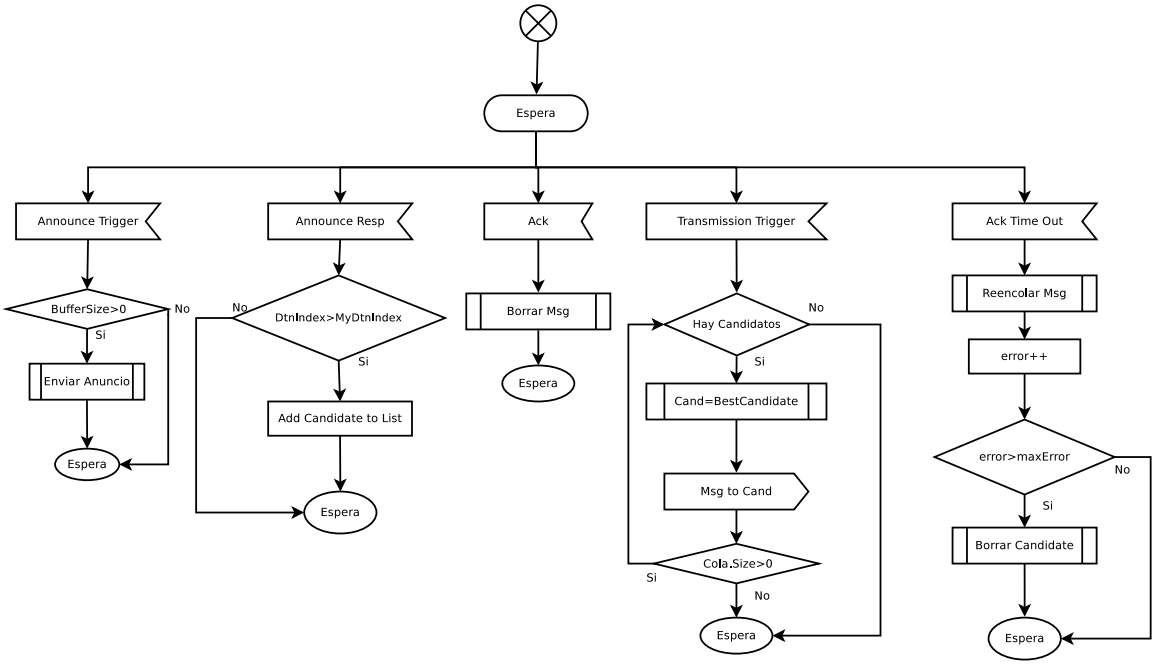


Figura 10: Funcionamiento de un nodo *custodian*

4. Si se recibe un mensaje *Data*, se desencapsulan los paquetes contenidos en el mismo y se añaden a la cola de salida, convirtiéndose el nodo en *custodian*.

5. Si los paquetes fueron encolados de manera satisfactoria se confirman con un *Data ACK*

la figura 11 muestra un flujograma con el funcionamiento del nodo de manera simplificada.

Sumideros: Además de estas tres entidades presentes en los nodos de la red, ésta está también formada por los *Sinks* o sumideros. Los sumideros funcionan de la siguiente manera:

1. Anuncian su presencia a la red mediante un *Sink Announce*.

2. Cuando reciben un *Data* confirman los paquetes contenidos en el mismo, para que los *custodians* puedan eliminarlos de la cola.

3. Tras recibir los paquetes los envían hacía el núcleo de la red cableada donde serán ensamblados para formar el paquete original.

La figura 9, ilustra un ejemplo de intercambio entre varios nodos. que sucede de la siguiente manera:

1. Primero el *Custodian 1* envía un *Announce* vía broadcast que es recibido por los otros 2 nodos, es importante recordar que los *custodians* también son *candidates*.

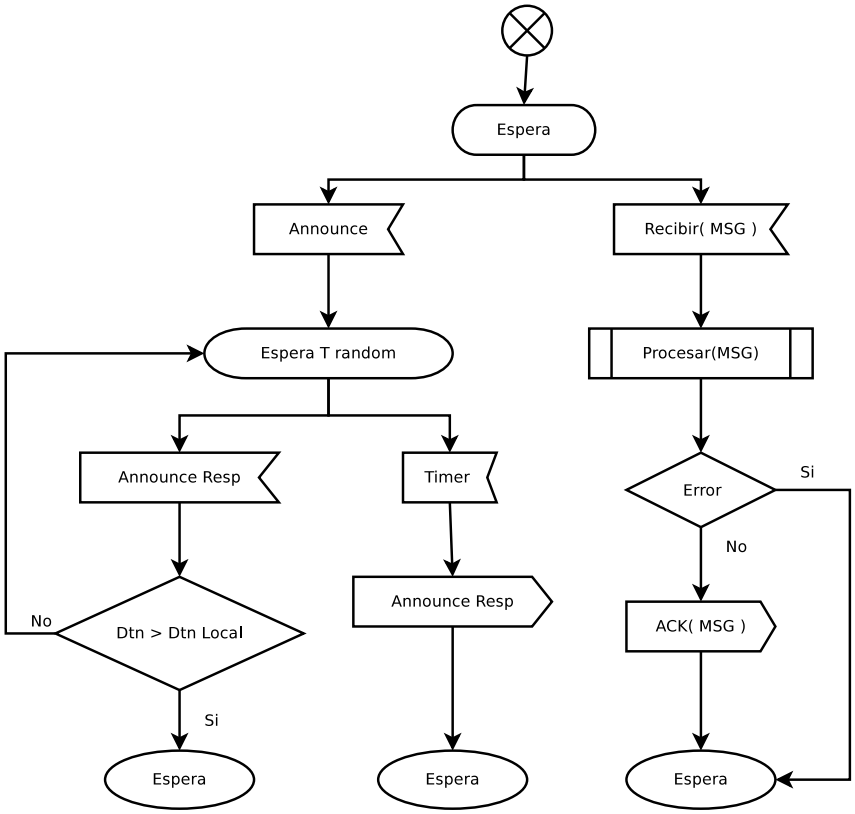


Figura 11: Funcionamiento de un nodo *candidate*

2. Tras recibir el *Announce* ambos nodos envían un *Announce Resp* mediante broadcast.

3. El *Custodian 2* también recibe el *Announce Resp* originado por el candidate.

4. Se inician 2 fases de transmisión entre ambos *custodians* y el *candidate*.

5. Tras recibir el ACK proveniente del *candidate* el *custodian* elimina el paquete de su buffer.

4. Verificación mediante simulaciones

Para comprobar la validez del protocolo diseñado y dada la gran dificultad en coste y tiempo para probarlo en un entorno real se decidió comprobar la validez del protocolo diseñado mediante simulaciones dirigidas por eventos. En esta sección describiremos la metodología adaptada y detallaremos los modelos implementados para al final exponer los resultados obtenidos. Como protocolo DTN de referencia con el que comparar nos hemos decidido por el protocolo *Epidemic* [22] que ya fue descrito en la sección anterior 3.1 ya que este protocolo proporciona una cota mínima para el retardo y una cota máxima para la probabilidad de entrega siempre que los recursos disponibles en la red sean disponibles.

4.1. Herramientas utilizadas

En este apartado describiremos brevemente el conjunto de herramientas utilizadas para llevar a cabo nuestras simulaciones.

4.1.1. Ns3

Ns3 es la nueva versión del ampliamente utilizado por la comunidad investigadora simulador dirigido por eventos Network Simulator 2 (Ns2). Ns3 como Ns2 es un simulador de redes dirigido por eventos [4]. Actualmente Ns3 se encuentra en desarrollo por lo que algunos de los modelos disponibles se encuentran en un estado aún embrionario que no permite su utilización de manera intensiva en la investigación. Sin embargo, a tenor de publicaciones anteriores, tanto los modelos de red referentes a 802.11* como los modelos de propagación están en un estado suficientemente avanzado como para considerar sus resultados válidos [5].

4.1.2. C4R

C4R es un software para la generación de trazas de movilidad sobre mapas reales. Esta herramienta fue desarrollada en el grupo [12], la figura 12 muestra una captura de pantalla de C4R. Para la generación de las trazas C4R se apoya en Simulation of Urban MObility (SUMO). SUMO es un simulador de tráfico microscópico, ie. el movimiento y la posición de los vehículos se calcula de manera detallada y no como un flujo o un todo [6].

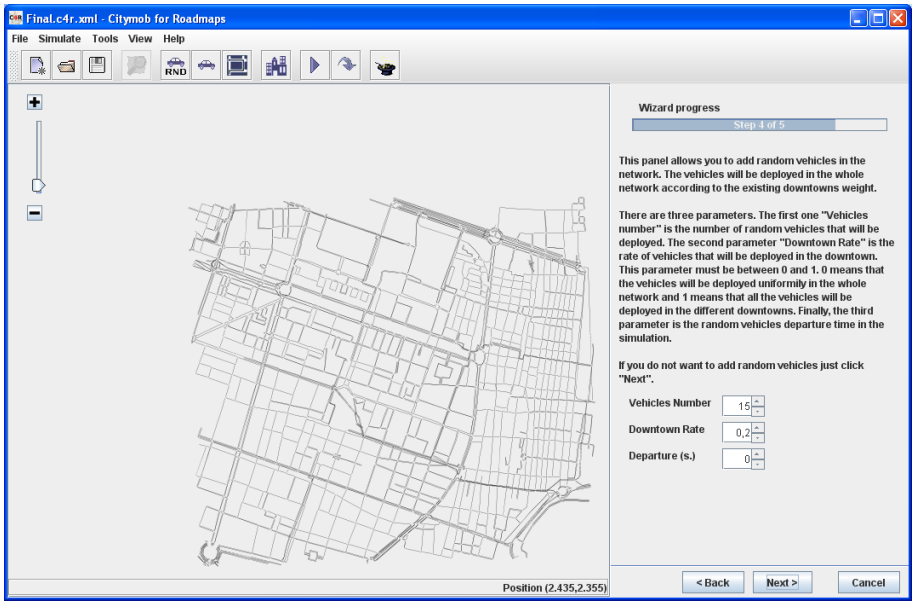


Figura 12: Captura de pantalla de C4R.

4.2. Escenario de simulación

En este apartado describiremos el escenario de simulación sobre el cuál hemos comprobado el rendimiento de nuestro protocolo. Dentro de las limitaciones de una simulación hemos intentado que esta se asemejase el máximo posible a las condiciones que experimenta un vehículo en una situación real. Con este objetivo en mente justificaremos cada uno de los parámetros elegidos.

4.2.1. Configuración de los nodos

En nuestro escenario de simulación cada nodo cuenta con 2 interfaces, la primera de ellas es una interfaz 802.11p sintonizada en la banda de los 5Ghz tal y como indica la normativa explicada anteriormente, la segunda de ellas es una interfaz 802.11g, esta interfaz se encuentra sintonizada en la banda de 2.4Ghz. Los parámetros correspondientes a los distintos protocolos situados en capas de red inferiores como UDP, IP, ARP, o los parámetros correspondientes a la capa MAC se han dejado configurados en su valor por defecto. La tabla 1 muestra los parámetros que consideramos más relevantes en nuestra simulación.

4.2.2. Red de carreteras

Con el ánimo de realizar una simulación lo más cercana a la realidad posible se decidió prescindir de escenarios sencillos utilizados en la validación de propuestas anteriores como el escenario *Manhattan* donde las calles forman una cuadrícula perfecta. En contraposición nosotros decidimos utilizar una porción del mapa de la ciudad de Valencia. La zona presentada en el mapa fue elegida por presentar un escenario que creemos bastante representativo, ya que se mezclan zonas de grandes avenidas con calles estrechas. La figura 13 muestra el mapa de calles utilizado, el tamaño de la zona es de unos $5km^2$.

4.2.3. Modelos de propagación

En la mayoría de los artículos donde se presentaron las propuestas analizadas en 3.2, se utilizan modelos de propagación demasiado básicos, en su mayoría deterministas, fijando un rango de cobertura para las transmisiones. Nosotros creemos que las simulaciones deben intentar ser lo más fieles posibles a la realidad. Por ello, dentro de las limitaciones de una simulación y de los modelos que proporciona Ns3 se decidió utilizar para la red 802.11p un modelo de propagación de 2 rayos determinista, al que se le añadió un *fading* aleatorio que sigue una distribución de Nakagami. Publicaciones anteriores han demostrado que dicha distribución se ajusta bastante bien a las variaciones que experimenta en el tiempo la potencia recibida por un nodo situado a una distancia dada [9].

Ya que nuestra investigación se centra en la comunicación entre los nodos y para simplificar la parte encargada de la predicción de oportunidades de transmisión con los puntos de acceso (sumideros), se decidió que el modelo de propagación utilizado en la red 802.11g tendría un alcance limitado a cierta distancia y totalmente determinista.

Figura 13: Mapa de la ciudad de Valencia utilizado en nuestras simulaciones.

4.2.4. Movilidad de los nodos

En nuestro escenario de simulación todos los vehículos se mueven siguiendo una ruta que fue generada de manera aleatoria utilizando C4R y SUMO. Para poder utilizar las trazas de movilidad generadas por SUMO en Ns3 hemos desarrollado un parser para los ficheros *xml* generados por SUMO, que utilizando el modelo de movilidad *Constant Velocity Mobility Model* incluido en Ns3, programa las variaciones de dicha movilidad en el tiempo.

La generación de trazas mediante C4R y SUMO presenta algunos problemas si lo comparamos con los mecanismos habitualmente utilizados, sin embargo estos problemas quedan claramente compensados con el aumento de fiabilidad de dichas simulaciones si lo comparamos con modelos de movilidad típicos. La mayoría de estas limitaciones son intrínsecas a unas rutas generadas aleatoriamente, a continuación enumeraremos algunas de ellas así como la solución aportada por nosotros:

- Duración de la movilidad indeterminada:Al tratarse de rutas aleatorias la longitud de las mismas es imposible de determinar a priori por lo que es muy posible que algunos nodos acaben su recorrido antes del final de la simulación. Esto supone un claro problema para nuestras simulaciones, ¿que hacer cuando un nodo alcanza su destino y se para? Nuestra solución ha sido asignar un valor a la coordinada

Parámetro	Valor
Potencia de transmisión if 802.11p	$33dBm \rightarrow 2W$
Potencia de transmisión if 802.11g	$20dBm \rightarrow 200mW$
Rango de cobertura AP	$250m$
Rango de cobertura if 802.11p	$\approx 750m$
Tiempo de inicio	Uniforme entre 0 y 5 seg
Tráfico generado	2000 Bytes cada 5 seg
Tiempo de ejecución del generador	100 seg
Ruta	Aleatoria de duración indeterminada

Tabla 1: Parámetros de los nodos.

Z de 10000m, con esto nos aseguramos que el nodo queda fuera de nuestra red, además deshabilitamos en el nodo cualquier transmisión de datos para no sesgar las estadísticas.

- Densidad media de los nodos desconocida: C4R nos permite generar rutas que se inician en determinados instantes de la simulación, sin embargo, ya que la duración de las mismas es desconocida, resulta imposible saber cuando debemos añadir más nodos a la red para intentar asegurar una densidad media. Para rodear este escollo se decidió fijar como parámetro de las simulaciones el número de nodos situados al inicio.

Como cabe de esperar los puntos de acceso o sumideros se han colocado de manera fija sobre el mapa. La localización de dichos puntos de acceso se corresponde con una ubicación sobre los puntos que tienden a concentrar más tráfico de vehículos en nuestras simulaciones. Creemos que esta ubicación es la que elegiría la empresa o entidad encargada del despliegue de la red por lo que nos parece mucho más acertada que una distribución aleatoria de los mismos. La figura 13 muestra la situación de los mismos sobre la red de carreteras, además hemos añadido la tabla 2 con la posición de los mismos en coordenadas X,Y.

4.2.5. Tráfico generado

En nuestras simulaciones hemos tratado de que el tráfico introducido en la red sea semejante al generado por una red de sensores que muestrea ciertos parámetros cada un tiempo t y obtiene un tamaño determinado de datos. Así cada nodo genera un mensaje de 2000 Bytes cada 5 segundos durante los 100 primeros segundos de la simulación, este mensaje es a su vez dividido en 10 partes a las que se añade un 20 % de redundancia, dando un total de 12 partes, cada una de estas partes es encapsulada en un nuevo paquete. La periodicidad y el tamaño de los paquetes generados por los sensores así como el resto de los valores utilizados dependen de la supuesta aplicación simulada, pero creemos que estos parámetros se ajustan bastante bien si asumimos que los datos recogidos son importantes pero no *vitales* (en el caso de que lo fueran quizás sería recomendable más redundancia

Punto de acceso	X	Y
1	1780	1939
2	1385	298
3	284	685
4	545	1598
5	1699	1215

Tabla 2: Posición de los puntos de acceso

o un menor intervalo de generación). Utilizando estos parámetros el total del tráfico introducido a la red es linealmente proporcional al número de nodos en la misma. Sin embargo, atendiendo a las limitaciones introducidas por el tipo de movilidad, no se puede garantizar que todos los nodos alcancen los 100 segundos de simulación dentro de la red, este hecho hace que el número de mensajes enviados varíe notablemente incluso entre las diferentes iteraciones dentro de una misma simulación.

4.2.6. Simulaciones y número de iteraciones

Con nuestras simulaciones hemos intentado determinar el impacto de la cantidad de nodos en nuestra red. Para ello hemos introducido al inicio de la simulación 15, 30, 63, 125, y 188 nodos, variando la densidad de nodos inicial de 3 $nodos/km^2$ a 37 $nodos/km^2$. Además, con el objetivo de obtener valores medios representativos y evitar los sesgos introducidos por la movilidad hemos realizado 10 iteraciones variando en cada una de ellas la movilidad.

5. Resultados

En esta sección se expondrán los resultados obtenidos. Como ya explicamos anteriormente cada simulación se ha repetido 10 veces variando tanto la movilidad como la semilla del generador de números aleatorios en cada una de las repeticiones. Cada parámetro seleccionado será discutido y evaluado en un apartado diferente. Es importante destacar que para cada uno de los protocolos analizados se han utilizado siempre las mismas trazas de movilidad, esto es, en la primera iteración de la simulación con 15 nodos y protocolo RTaDAP se ha utilizado la misma traza que para la primera iteración de la simulación con 15 nodos y protocolo *Epidemic*, esto permite que nuestros resultados con ambos protocolos sean comparables a pesar de la gran varianza que presentan algunos de ellos. Los parámetros que hemos analizado son los siguientes.

- Retardo: El primer parámetro que hemos creído importante evaluar es el retardo que sufren los paquetes antes de llegar a un punto de acceso y poder ser recogidos. Para dar una visión clara del mismo hemos elegido representar el tiempo en el cual se entrega cierto porcentaje de paquetes, obviamente se trata de una función

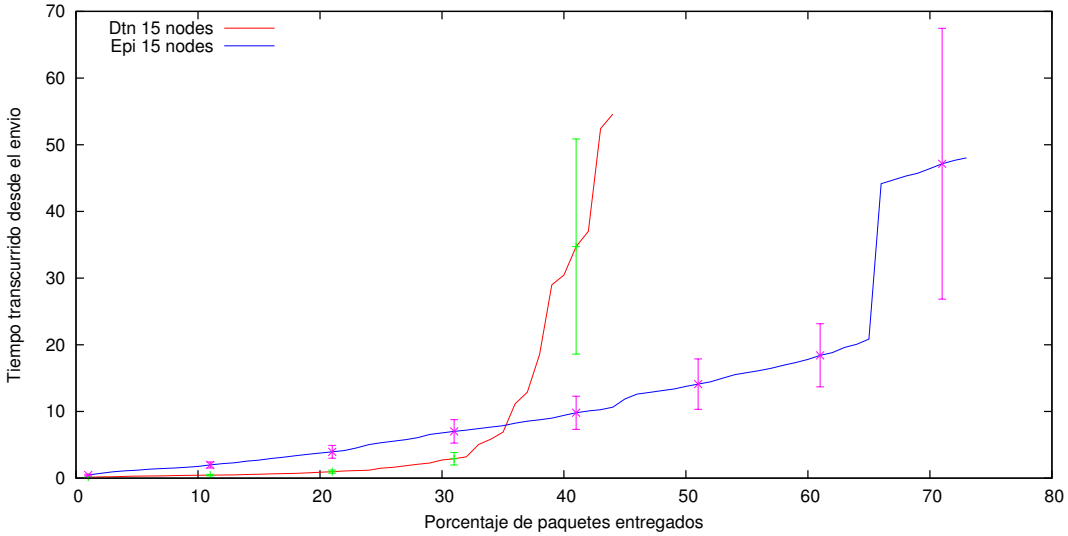


Figura 14: Función acumulativa del retardo sufrido por los paquetes.

acumulativa ya que si en un tiempo $t1$ se entregaron x paquetes, para un porcentaje de paquetes y mayor que x, $t2$ será siempre mayor o igual que $t1$.

- Porcentaje de mensajes entregados: Este parámetro indica que porcentaje de los mensajes generados en los 100 primeros segundos de nuestra simulación pudieron ser entregados antes del fin de la misma. Dado que se trata de redes DTN ningún paquete debe ser descartado por los nodos, sin embargo, se pueden producir pérdidas cuando un nodo es *inhabilitado* (llegan al final de su ruta) cuando aún contiene paquetes en su cola.

- Número de mensajes por mensaje enviado: Este parámetro contabiliza en media el número total de mensajes generados por cada uno de los mensajes originales, ie. antes de añadir redundancia, introducidos en la red. Por "número total" nos referimos a *todos* los tipos de mensajes especificados anteriormente en 3.5.1. Con este parámetro conseguimos evaluar los recursos utilizados por cada uno de los protocolos.

La figura 14 presenta los resultados que hemos obtenido con simulaciones de 15 nodos $\approx 3\ nodos/km^2$. La gráfica muestra también los intervalos de confianza obtenidos para el 95 % y 10 muestras, para una mayor claridad se muestran los intervalos tan solo cada 10 puntos. En dicha figura se puede apreciar como cuando la densidad de nodos es baja el protocolo *Epidemic* tiene un rendimiento superior a RTaDAP en cuanto al parámetro de retardo se refiere. La principal razón de este mayor rendimiento es la manera de actuar de ambos protocolos, mientras que el protocolo *Epidemic* intenta enviar una copia de cada mensaje a cada uno de los nodos con los que se establece contacto RTaDAP envía una única copia del mensaje tan solo a aquellos nodos cuyo *DtnIndex* es superior al *DtnIndex* del nodo actual. Esta manera de actuar hace que cuando el *DtnIndex* del nodo actual

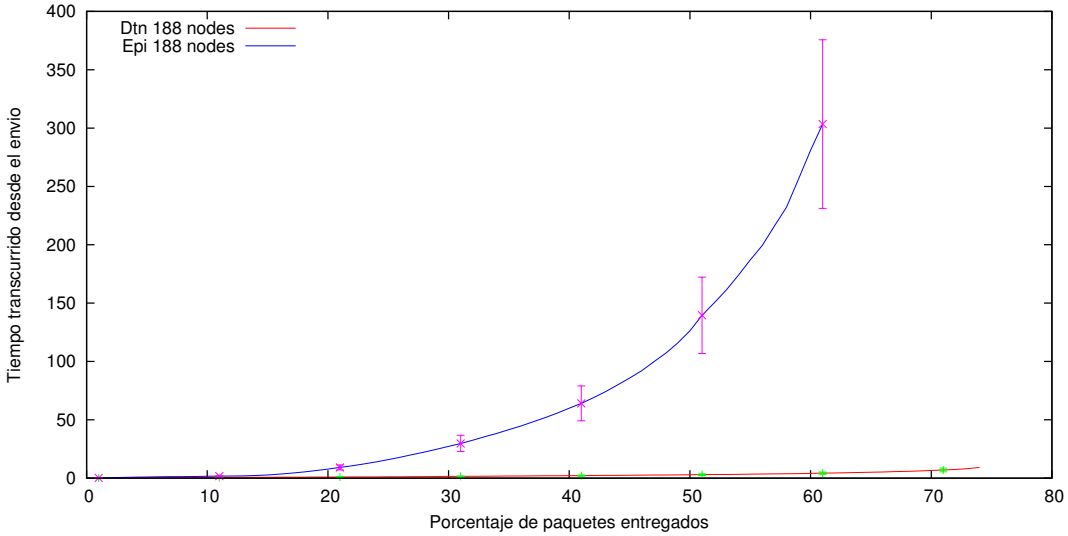


Figura 15: Función acumulativa del retardo sufrido por los paquetes.

es igual a 0 y el *DtnIndex* de todos sus vecinos es también 0 no se produce intercambio alguno de mensajes, desperdiciando así los posibles contactos futuros de los vecinos con otros nodos cuyo *DtnIndex* sea mayor que el local. Por el contrario el protocolo *Epidemic* al enviar copias a todos los vehículos aprovecha mucho mejor las futuras oportunidades no previstas. El gran tamaño de los intervalos de confianza para un porcentaje de paquetes entregados alto se debe a que el retardo depende en gran medida de la movilidad, y dada la baja densidad de nodos la variedad de rutas que pueden tomar éstos es muy amplia, lo que deriva en una gran varianza en los valores de retardo. Por otro lado, la figura 17 nos muestra la probabilidad de recepción según el número de nodos presentes en la red, si atendemos al valor obtenido para 15 nodos vemos como el protocolo *Epidemic* se comporta mucho mejor que RTaDAP cuando existen pocos nodos. Por último, la figura 18 muestra el número de transmisiones por mensaje enviado, como era de esperar, aún cuando la densidad de los nodos es baja *Epidemic* consume muchísimos más recursos que RTaDAP.

En el otro extremo la figura 15 presenta los resultados que hemos obtenido con simulaciones de 188 nodos $\approx 37{,}6\ nodos/km^2$. Al contrario que en el caso anterior en esta gráfica observamos como el tiempo necesario para entregar cierta cantidad de mensajes utilizando el protocolo *Epidemic* crece de manera exponencial, mientras que en el caso de RTaDAP crece de manera aproximadamente lineal. El gran descenso en las prestaciones de *Epidemic* se debe sobre todo al aumento de la congestión cuando aumenta el número de nodos. En este caso cabe recordar que al aumentar el número de nodos aumenta también el número de mensajes introducidos en la red. Si observamos de nuevo la figura 17 vemos como en este caso la probabilidad de entrega en RTaDAP es alrededor de un 15 % mejor que la de *Epidemic*. Por otro lado en la figura 18 observamos como para una densidad de 188 nodos el número total de mensajes enviados por mensaje transmitido es menor que para simulaciones con menos nodos, esto, que puede parecer una anomalía, se pro-

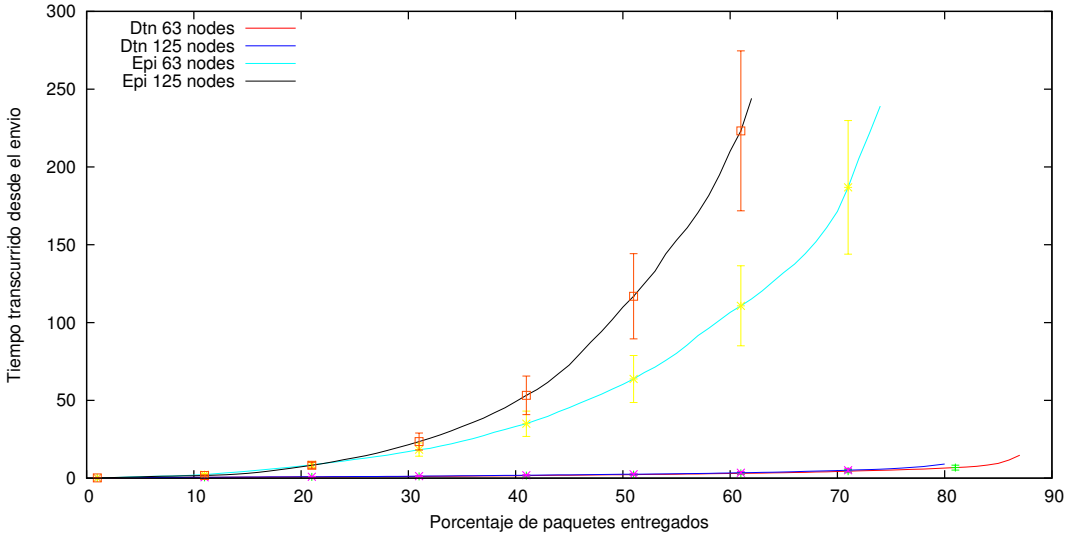


Figura 16: Función acumulativa del retardo sufrido por los paquetes.

duce debido a que la congestión es tal que resulta imposible para los nodos establecer conexiones y aprovechar las distintas oportunidades de transmisión, generando así menos tráfico total.

Finalmente la figura 16 nos da una visión más general de como evoluciona el retardo sufrido por los mensajes para cada uno de los protocolos. El primer punto que llama la atención es la gran mejora que presenta RTaDAP frente a *Epidemic* pero sobre todo, lo que nosotros consideramos más importante es que en RTaDAP el retardo sufrido por los paquetes no varía de manera notable al variar el número de nodos en la red. Analizando los valores centrales de la figura 17 vemos como la probabilidad de entrega cuando se utiliza RTaDAP alcanza su máximo en la columna correspondiente a 63 nodos para posteriormente decrecer aunque nunca desciende por debajo del 75 %, comparando esta figura con la correspondiente al retardo podemos concluir que el aumento del número de nodos conlleva a su vez un aumento en una proporción mayor de los paquetes que quedan en el buffer de los nodos al ser estos extraídos de la red. Por último, atendiendo a la figura 18 vemos como el uso de los recursos es muchísimo mayor cuando se utiliza *Epidecmic* que en el caso de utilizar RTaDAP. Aún más importante, para valores de densidad de nodos relativamente bajos (15 y 30) vemos como al doblar el número de nodos en la red, doblando así el tráfico introducido en la misma, el protocolo *Epidemic* genera más del doble de mensajes totales por mensaje enviado, mientras que en el caso de RTaDAP el aumento de este valor es mínimo y parece seguir una progresión lineal. Como ya se expuso anteriormente, la razón para que el total de mensajes generados por mensaje enviado cuando se utiliza *Epidemic* disminuye cuando existe saturación en la red se debe a no utilizar todas las oportunidades de transmisión.

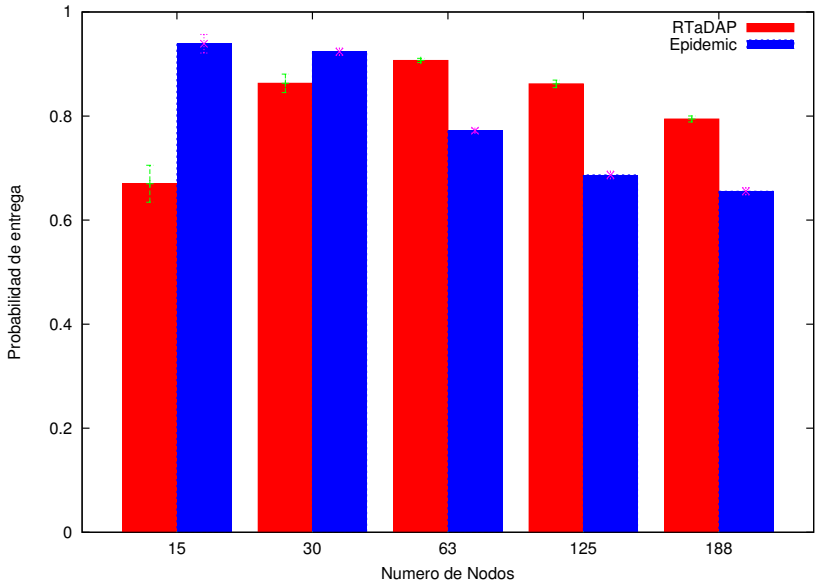


Figura 17: Probabilidad media de entrega para cada simulación.

6. Conclusiones

A lo largo de esta tesina y a tenor de los resultados expuestos en el capítulo anterior creemos que ha quedado suficientemente demostrado como un protocolo basado en la predicción de la movilidad de los nodos tendrá generalmente un mejor rendimiento que un protocolo puramente oportunístico que no utiliza más información que la disponible en el momento del intercambio, disminuyendo además la utilización de recursos de la red. Éste último punto nos parece el más importante ya que, aunque en las redes vehiculares no suele haber restricciones de energía, potencia o almacenamiento, pueden existir otras aplicaciones con diferentes fines coexistentes con la red DTN por lo que conviene minimizar los recursos utilizados. Creemos también haber demostrado como una única función puede combinar varios factores para dar un único parámetro de decisión.

Por otro lado somos conscientes de que RTaDAP presenta algunos problemas, como el bajo rendimiento en redes muy poco pobladas, y debemos de trabajar en su mejora.

7. Trabajo Futuro

Esta tesina forma parte de un trabajo más amplio con finalidad de desembocar en una tesis doctoral. Dentro de ese marco las siguientes tareas que tenemos pendientes desarrollar serían las siguientes:

- Mejora de las herramientas de simulación: Una de las primeras acciones que pretendemos llevar a cabo es la mejora de C4R para permitir mayor control en la generación de rutas aleatorias.

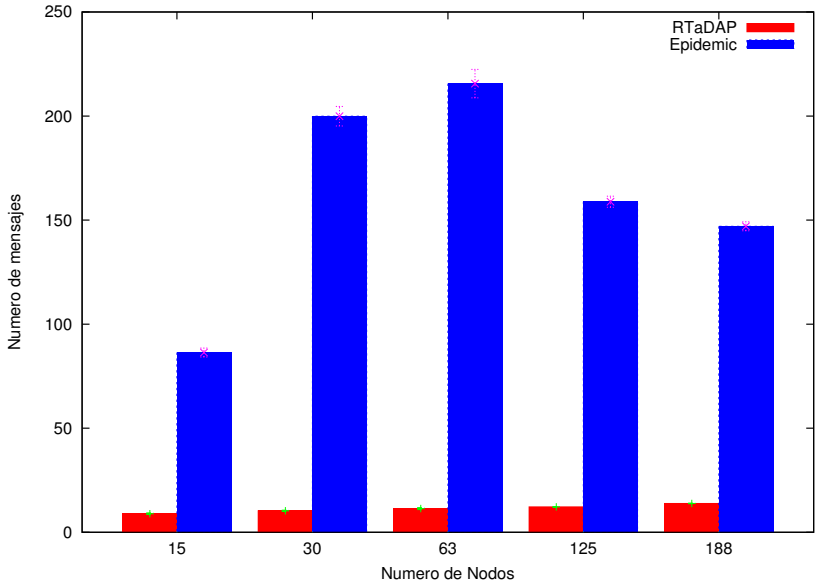


Figura 18: Número medio de mensajes totales enviados en la red por cada mensaje generado.

- Explorar otras propuestas: En estos momentos ya nos encontramos inmersos en la implementación de protocolos DTN diferentes con el fin de poder comparar RTaDAP con el objetivo de obtener nuevas ideas para su mejora.
- Adaptación de RTaDAP para aplicaciones específicas: En el futuro exploraremos las posibilidades de RTaDAP en su utilización en aplicaciones específicas como puede ser eXtended Floating Car Data (xFCD) [20].

8. Agradecimientos

El trabajo expuesto en esta tesina ha sido financiado con fondos pertenecientes al proyecto “SEISCIENTO: provisión de servicios ubicuos adaptables en entornos vehiculares” con número TIN2008-06441-C02-01 financiado por el Ministerio de Educación y Ciencia.

Referencias

[1] Supplement to IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks - Specific Requirements. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: High-Speed Physical Layer in the 5 GHz Band. Technical report, 1999.

[2] IEEE Standard for Information technology–Telecommunications and information exchange between systems–Local and metropolitan area networks–Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 6: Wireless Access in Vehicular Environments. Technical report, July 2010.

[3] IEEE Standard for Information Technology–Telecommunications and information exchange between systems–Local and metropolitan area networks–Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment 10: Mesh Networking. Technical report, 2011.

[4] Ns3 website. http://www.nsnam.org/, December 2011.

[5] Nicola Baldo, Manuel Requena, Jose Nunez, Marc Portoles, Jaume Nin, Paolo Dini, and Josep Mangues. Validation of the ns-3 ieee 802.11 model using the extreme testbed. In *Proceedings of SIMUTools Conference, 2010*, March 2010.

[6] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz. Sumo - simulation of urban mobility: An overview. In *SIMUL 2011, The Third International Conference on Advances in System Simulation*, pages 63–68, Barcelona, Spain, October 2011.

[7] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss. Delay-Tolerant Networking Architecture. Number RFC 4838. IETF, April 2007.

[8] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, E. Travis, and H. Weiss. Delay-Tolerant Network Architecture:The Evolving Interplanetary Internet. IPN Research Group, 2002.

[9] Qi Chen, Felix S. Eisenlohr, Daniel Jiang, Marc T. Moreno, Luca Delgrossi, and Hannes Hartenstein. Overhaul of ieee 802.11 modeling and simulation in ns-2. In *Proceedings of the 10th ACM Symposium on Modeling, analysis, and simulation of wireless and mobile systems*, MSWiM '07, pages 159–168, New York, NY, USA, 2007. ACM.

[10] Wai Chen, R. Guha, J. Chennikara-Varghese, M. Pang, R. Vuyyuru, and J. Fukuyama. Context-driven disruption tolerant networking for vehicular applications. In *Vehicular Networking Conference (VNC), 2010 IEEE*, pages 33–40. IEEE, December 2010.

[11] Electronic Communications Committee. ECC recommendation (08)01, use of the band 5855-5875 MHz for intelligent transport systems (ITS), 2008.

[12] Manuel Fogue, Piedad Garrido, Francisco J. Martinez, Juan-Carlos Cano, Carlos T. Calafate, and Pietro Manzoni. Using roadmap profiling to enhance the warning message dissemination in vehicular environments. In *36th IEEE Conference on Local Computer Networks (LCN 2011)*, Bonn, Germany, October 2011.

[13] R. Frank, E. Giordano, P. Cataldi, and M. Gerla. TrafRoute: A different approach to routing in vehicular networks. In *Wireless and Mobile Computing, Networking and Communications (WiMob), 2010 IEEE 6th International Conference on*, pages 521–528. IEEE, October 2010.

[14] Ting-Kai Huang, Chia-Keng Lee, and Ling-Jyh Chen. PRoPHET+: An Adaptive PRoPHET-Based Routing Protocol for Opportunistic Network. In *Advanced Information Networking and Applications (AINA), 2010 24th IEEE International Conference on*, pages 112–119. IEEE, April 2010.

[15] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot. Optimized link state routing protocol for ad hoc networks. In *Multi Topic Conference, 2001. IEEE INMIC 2001. Technology for the 21st Century. Proceedings. IEEE International*, pages 62–68. IEEE, August 2001.

[16] K. C. Lee and M. Gerla. Opportunistic vehicular routing. In *Wireless Conference (EW), 2010 European*, pages 873–880. IEEE, April 2010.

[17] Ze Li and Haiying Shen. A Direction Based Geographic Routing Scheme for Intermittently Connected Mobile Networks. In *Embedded and Ubiquitous Computing, 2008. EUC '08. IEEE/IFIP International Conference on*, volume 1, pages 359–365. IEEE, December 2008.

[18] Anders Lindgren, Avri Doria, and Olov Schelén. Probabilistic routing in intermittently connected networks. In *SIGMOBILE Mob. Comput. Commun. Rev.*, volume 7, pages 19–20, New York, NY, USA, July 2003. ACM.

[19] C. E. Perkins and E. M. Royer. Ad-hoc on-demand distance vector routing. In *Mobile Computing Systems and Applications, 1999. Proceedings. WMCSA '99. Second IEEE Workshop on*, pages 90–100, 1999.

[20] R. Quintero, A. Llamazares, D. F. Llorca, M. A. Sotelo, L. E. Bellot, O. Marcos, I. G. Daza, and C. Fernandez. Extended Floating Car Data system - experimental study. In *Intelligent Vehicles Symposium (IV), 2011 IEEE*, pages 631–636. IEEE, June 2011.

[21] Thrasyvoulos Spyropoulos, Konstantinos Psounis, and Cauligi S. Raghavendra. Spray and wait: an efficient routing scheme for intermittently connected mobile networks. In *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, WDTN '05, pages 252–259, New York, NY, USA, 2005. ACM.

[22] A. Vahdat and D. Becker. Epidemic Routing for Partially Connected Ad Hoc Networks. In *Technical Report CS-200006*, April 2000.

[23] Jingfeng Xue, Xiumei Fan, Yuanda Cao, Ji Fang, and Jiansheng Li. Spray and Wait Routing Based on Average Delivery Probability in Delay Tolerant Network. In *Networks Security, Wireless Communications and Trusted Computing, 2009. NSWCTC '09. International Conference on*, volume 2, pages 500–502. IEEE, April 2009.

[24] Danlei Yu and Young-Bae Ko. FFRDV: Fastest-Ferry Routing in DTN-enabled Vehicular Ad Hoc Networks. In *Advanced Communication Technology, 2009. ICACT 2009. 11th International Conference on*, volume 02, pages 1410–1414. IEEE, February 2009.