



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



Departamento de Informática de Sistemas y Computadores

Universitat Politècnica de València

# **Análisis y Rendimiento del Protocolo Bundle con la Implementación DTN2, en Escenarios Oportunistas**

TRABAJO FIN DE MASTER

Máster en Ingeniería de Computadores y Redes

*Autor*

Leonardo Chancay García

*Director*

Dr. Pietro Manzoni

September 4, 2015

### ***Agradecimientos***

*A mi Familia por su apoyo incondicional, a los Profesores  
y compañeros del Departamento de Informática de Sistemas y Computadores.*

*Al Gobierno de la República del Ecuador y a su programa  
de Becas a través de la Secretaria de Educación Superior,  
Ciencia, Tecnología e Innovación - SENESCYT*

## Resumen

En las últimas décadas, las Redes de Computadores han permitido acortar las distancias facilitando el envío de información de un sitio a otro. Existen varios tipos de redes, desde las más convencionales, las cableadas, pasando a las Redes Inalámbricas, hasta aquellas llamadas *Redes Tolerantes a Retardos (DTN)*

Las DTN se han ido implementando en los últimos años en todo el planeta incluso en el espacio, logrando comunicar grandes distancias sin necesidad de una infraestructura permanente, con nodos en movimiento semi constante.

Este tipo de redes son muy prácticas para asegurar el envío y la recepción de los mensajes aún cuando las condiciones de transmisión no sean las mejores. Éstas han llegado a ser usadas en entornos genéricos como en un ambiente urbano, en los sistemas de transporte público, llevando mensajes de la región de una ciudad a otra. Un claro ejemplo de esto es el uso que se les está dando dentro del nuevo paradigma de la computación móvil y las ciudades inteligentes.

El *Protocolo Bundle (RFC 5050)* es el protocolo de comunicación estándar en DTNs. En los últimos años, DTN ha recibido una gran cantidad de interés por parte de la comunidad de redes. Aun así DTN sigue siendo un área de investigación joven y sus especificaciones, así como sus implementaciones aún no han alcanzado el mismo estado de madurez que el de otras áreas de internet.

Existen además muchos protocolos de encaminamiento en DTN, que tienen como objetivo aumentar la probabilidad de encontrar un camino que pueda asegurarnos la entrega de información. Para descubrir este camino se utilizan una variedad de mecanismos, incluyendo la estimación de probabilidades de encontrar

el mejor nodo que asegure la entrega. Además, en DTN es vital tener en cuenta aspectos como el consumo de energía y la memoria que un nodo es capaz de requerir.

El objetivo de este trabajo es exponer el funcionamiento del Protocolo Bundle y demostrar cómo el retardo en la entrega y la pérdida de paquetes puede resultar en la reducción del rendimiento en una red de computadores utilizando una implementación DTN2.

Dentro del estudio analizamos algunos aspectos claves: mecanismo de almacenamiento y reenvío, transferencia bajo custodia, características de los nodos, contactos entre nodos y algoritmos de encaminamiento. También se analizan las futuras tendencias en el desarrollo de este tipo de redes.

Al concluir los experimentos, se constató que el rendimiento es mayor en una red que puede sufrir retardos o pérdidas si usamos DTN2 que si usamos transferencias de TCP puro, por lo que es factible el uso del Protocolo Bundle y las tecnologías que aquí se estudian, en distintas áreas tecnológicas y escenarios oportunistas.

*Palabras Claves:* Redes Oportunistas, Redes Tolerantes a Retardos, DTN2, Protocolo Bundle

## **Abstract**

In past few decades, Computer Networks have led to shortening of distances allowing the transmission of information from one place to another. There are several types of networks, from the conventional wired to wireless networks, including *Delay Tolerant Networking (DTN)*.

DTN has been implemented in recent years around the planet even in space, managing to communicate long distances without permanent infrastructure, with non-static nodes.

Such networks are very practical to ensure the sending and receiving of messages even when transmission conditions are not the best. These are used in generic environments such as urban scenarios including public transport systems that can carry messages from one part of the city to another. A clear example of its use can be observed within the paradigm of mobile computing and smart cities.

*The Bundle Protocol (RFC 5050)* is the standard communication protocol used in DTN. In the recent years, DTN has received a lot of interest from networking community. Being relatively new, the specifications and implementations of DTN have not yet reached the same stage of maturity like other research areas related to networking.

There are also many DTN routing protocols, that aim to increase the probability of finding a route to ensure the delivery of information. To discover this route, a variety of mechanisms are used, including estimation of the odds of finding the best node that would ensure delivery. Furthermore, in DTN, it is vital to consider aspects such as power consumption, and the memory that a node has to offer.

The aim of this master thesis is to present the operation of Bundle Protocol, and using an actual implementation of DTN2 demonstrate how the delay in delivery and packet loss can result in the reduction of performance of a computer network.

Within the study we analyzed some key aspects: store and forward mechanism, custody transfer, characteristics of the nodes, contacts between the nodes, and routing algorithms; trends related to the possible future developments in such networks have also been analyzed.

As a conclusion from the conducted experiments, it was found that the performance is higher using DTN2 on a network that can suffer delays or losses, making it feasible to use the Bundle Protocol and technologies studied here in different technological areas and opportunistic scenarios.

*Keywords:* Opportunistic Networks, Delay Tolerant Networking, DTN2, Bundle Protocol

# Índice general

<b>1</b>	<b>Introducción</b>	<b>1</b>
1.1	Motivación . . . . .	2
1.2	Objetivos . . . . .	3
1.3	Metodología . . . . .	3
1.4	Estructura . . . . .	4
<b>2</b>	<b>Trabajo relacionado</b>	<b>5</b>
2.1	Redes Tolerantes a Retardos . . . . .	5
2.2	Protocolo Bundle . . . . .	8
2.3	Protocolos de Enrutamiento o Encaminamiento . . . . .	12
2.4	DTN2 . . . . .	14
2.4.1	Implementación . . . . .	15
2.4.2	Capacidades . . . . .	15
<b>3</b>	<b>Descripción del Experimento</b>	<b>18</b>
3.1	Equipos Utilizados . . . . .	19
3.1.1	Dummysnet . . . . .	19
3.1.2	Servidor y Cliente . . . . .	20
3.2	Descripción del Experimento . . . . .	21

<b>4</b>	<b>Evaluación</b>	<b>23</b>
4.1	Resultados . . . . .	24
4.1.1	Resultados TCP/IP . . . . .	25
4.1.2	Resultados DTN2 . . . . .	27
4.1.3	Resultados DTN2 con Perdidas . . . . .	30
4.1.4	Resultados DTN2 con Retardos . . . . .	30
4.1.5	Resultados DTN2 con Perdidas y Retardos en la red . . . . .	33
<b>5</b>	<b>Conclusiones</b>	<b>36</b>
5.1	Conclusiones . . . . .	36
5.2	Trabajo futuro . . . . .	37

# Índice de figuras

2.1	Esquema Red DTN en la Tierra [1] . . . . .	5
2.2	Esquema Red DTN en el Espacio Exterior [1] . . . . .	6
2.3	Red intermitente [1] . . . . .	7
2.4	Capas de TCP y Bundle [1] . . . . .	8
2.5	Reenvío de la información en TCP y DTN [1] . . . . .	9
2.6	Segmentación de la información en TCP/IP y P. Bundles [1] . . . .	10
2.7	Almacenamiento persistente [1] . . . . .	11
2.8	Cronología de los Protocolos de encaminamiento [2] . . . . .	13
3.1	Esquema de Red . . . . .	19
4.1	Bandwidth en TCP/IP inyectando Retardos y Perdidas . . . . .	23
4.2	DTN2 Bundles - Goodput . . . . .	26
4.3	Bundles - Goodput con Perdidas del 5 % . . . . .	28
4.4	Bundles - Goodput con Perdidas del 10 % . . . . .	28
4.5	Bundles - Goodput con Perdidas del 15 % . . . . .	29
4.6	Bundles - Goodput con Perdidas del 20 % . . . . .	29
4.7	Bundles - Goodput con Perdidas del 25 % . . . . .	29
4.8	Bundles - Goodput con Retardo de 10ms . . . . .	31
4.9	Bundles - Goodput con Retardo de 50ms . . . . .	31
4.10	Bundles - Goodput con Retardo de 100ms . . . . .	31

4.11 Bundles - Goodput con Retardo de 150ms . . . . .	32
4.12 Bundles - Goodput con Retardo de 200ms . . . . .	32
4.13 Goodput - Throughput para Retardos de 100ms con Perdidas . . .	33
4.14 Goodput - Throughput para Retardos de 150ms con Perdidas . . .	34
4.15 Goodput - Throughput para Retardos de 200ms con Perdidas . . .	35

# Índice de tablas

4.1	Bandwidth con Perdidas en TCP . . . . .	24
4.2	Bandwidth con Retardos en TCP . . . . .	24
4.3	Bundles y Goodput en DTN2 . . . . .	26

# Capítulo 1

## Introducción

A partir de que se creó el lenguaje el hombre siempre ha tratado de comunicarse, desde una simple figura pintada en una cueva hasta nuestros tiempos a través de líneas de código, tratando de compartir siempre sus ideas y transferir el conocimiento hacia las nuevas generaciones. Por este motivo el ser humano creó el correo convencional, el telégrafo, el teléfono, por nombrar algunas de las creaciones mas importantes.

Las redes de computadores llevan implementándose desde más de cincuenta años en nuestro planeta, al inicio conectando diferentes organismos gubernamentales, pasando después a gobiernos y países enteros.

El uso de las redes inalámbricas es un poco mas joven. Involucrando diferente tipos de tecnologías en las conexiones, actualmente podemos conectar la mayoría de dispositivos electrónicos con los que contamos en casa.

Existen diferentes tecnologías inalámbricas entre las que se destacan: Wireless, Bluetooth y Celular. Estas tecnologías se despliegan utilizando algún tipo de infraestructura para la transmisión de datos. Paralelamente existen otras redes inalámbricas, las cuales no requieren de una infraestructura que esté siempre conectada o disponible. Por el contrario podríamos decir que el éxito de estas redes

se basa en la movilidad que presentan algunos de sus nodos sobre todo en lugares donde no existe infraestructura para la transmisión de datos, como se hace referencia en [3] y [4].

A este tipo de redes se les da el nombre de *Redes Tolerantes a Retardos* o DTN en ingles, *Delay Tolerant Networking*, las cuales se sirven de distintos medios para asegurar la entrega de los mensajes o paquetes, aun cuando una conexión puede tomar minutos, horas o días en ocurrir.

El Protocolo Bundle definido en RFC 5050 [5], llegando a ser un protocolo estándar, del cual se pueden encontrar varias implementaciones dependiendo del lugar donde se vayan a utilizar y la aplicación para la cual se necesite.

En este trabajo expondremos varios conceptos, para posteriormente realizar las mediciones de la implementación DTN2 de código abierto para el sistema operativo Linux.

En algunos de los trabajos el rendimiento de las DTN's se analiza solo de manera teórica y el uso de simuladores especializados como es el caso de The ONE [6] donde se obtienen medidas aproximadas y los resultados no podrían estar en línea con los de una implementación real.

Adicionalmente hemos investigado el rendimiento sistemático sobre la capa de convergencia TCP y cual es el comportamiento habitual en estas redes oportunistas.

## **1.1 Motivación**

Debido al imparable campo tecnológico en el que las redes de computadores se encuentran desde hace años, la línea de investigación que podría resultar novedosa en este campo para varias personas es la de las Redes DTN.

Para hacer un análisis lo más cercano a la realidad se procedió al envío de

datos entre dos nodos en dos redes diferentes, variando el estado de la conexión de red, durante diferentes periodos de tiempo, para medir la cantidad de paquetes o bundles que se recibían, además de medir el *goodput* en cada una de las condiciones.

La investigación sobre el Protocolo Bundle y DTN2 es enriquecedora, debido a la versatilidad de los campos de aplicaciones en las que se usa en la actualidad y en los que podrían llegar a usarse, además de otras nuevas que se podrían plantear, ya que la transmisión de los mensajes se vería beneficiada sobre todo en aquellos lugares donde la infraestructura de red es escasa o no existe, mientras que los retardos o pérdidas que se presentan podrían ser muy elevados, siendo más que un reto un logro para los investigadores desarrollar aplicaciones para este tipo de redes.

## **1.2 Objetivos**

Los objetivos de este Trabajo de Fin de Máster son los siguientes:

- Estudiar el Protocolo Bundle y cual es su comportamiento tomando como referencia al modelo TCP/IP.
- Realizar pruebas sobre la implementación DTN2 con diferentes escenarios.
- Analizar los resultados obtenidos en este tipo de red DTN, basados en los puntos anteriores.

## **1.3 Metodología**

El presente trabajo se ha realizado en base a las propuestas y estudios anteriores realizados sobre redes oportunistas, la identificación de sus puntos flacos y la

validación del protocolo en una implementación real.

Para la obtención de los resultados se han simulado diferentes condiciones que podrían presentarse, configurando los parámetros en el nodo que comunica a ambas redes, formando lo que sería una red oportunista.

Una vez que se obtuvieron los datos, se evaluaron y se procedió a formular las conclusiones para un trabajo futuro.

## **1.4 Estructura**

Este Trabajo de Fin de Máster se encuentra dividido en cinco capítulos. El capítulo 1 en el cual nos encontramos ofrece una visión previa de las redes y del trabajo que hemos realizado. En el capítulo 2 se presentan algunos detalles de las investigaciones previas, así como conceptos esenciales sobre las redes oportunistas, el Protocolo Bundle y DTN2. En el capítulo 3 se realiza la descripción de cómo se llevaron a cabo los experimentos. En el capítulo 4 se realiza la evaluación sobre los resultados obtenidos. El capítulo 5 presenta las conclusiones de toda la investigación además se hace una referencia a un trabajo futuro sobre esta línea investigadora.

# Capítulo 2

## Trabajo relacionado

### 2.1 Redes Tolerantes a Retardos

Las Redes Oportunistas llevan siendo motivo de estudio en varios trabajos de investigación y desarrollo los cuales se centran en los campos y ramas para cubrir un determinado problema, dependiendo de hacia donde va orientado. En base al trabajo presentado en [1] explicaremos el funcionamiento de las DTN, donde se podrá comparar con el modelo TCP/IP (Transfer Control Protocol / Internet Protocol).

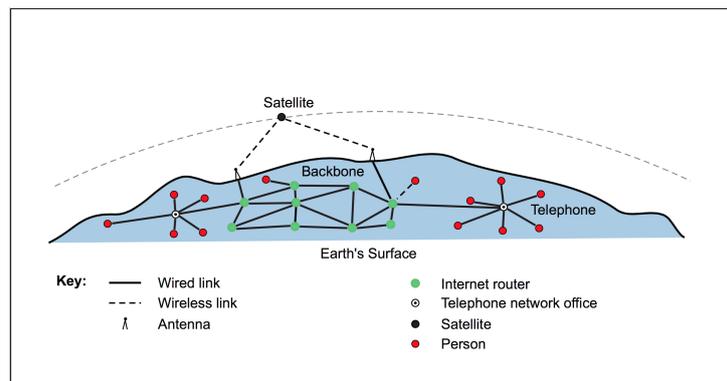


Figura 2.1: Esquema Red DTN en la Tierra [1]

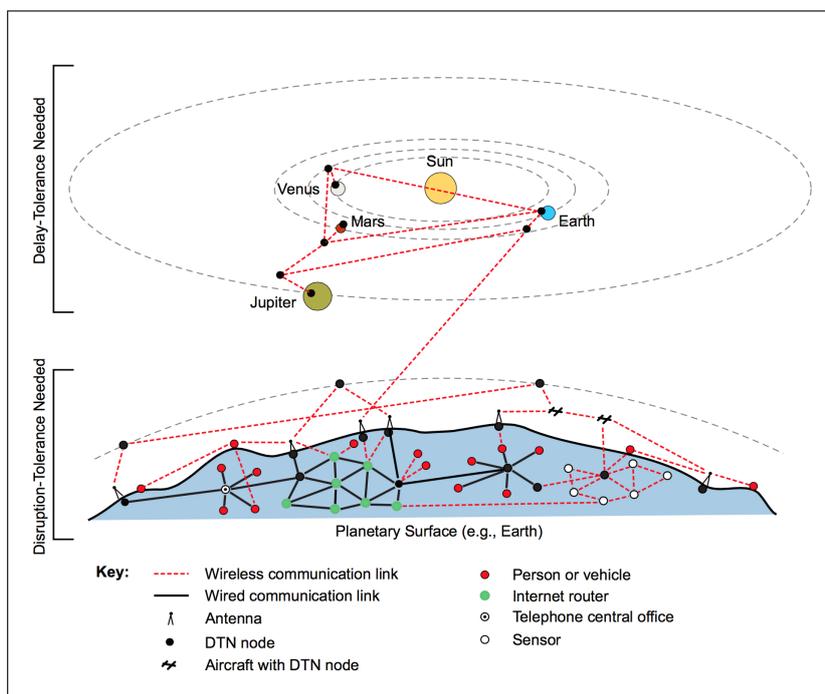


Figura 2.2: Esquema Red DTN en el Espacio Exterior [1]

Las DTN son promovidas por el grupo *Internet Research Task Force* siendo parte del *Delay Tolerant Networking Research Group* (DTNRG) [5], más información se puede encontrar en el portal <http://www.dtnrg.org>.

Esta tecnología fue propuesta por el *Interplanetary Internet Special Interest Group* (IPNSIG, <http://www.ipnsig.org>), se desarrolló al comienzo para ser utilizada en el espacio exterior, estableciendo comunicación interplanetaria, para luego ser utilizada en otras aplicaciones gubernamentales, civiles y militares.

En [7] sus autores se refieren a ellas como Redes Tolerantes a Retardo [8], debido a la forma como se almacena la información en los nodos y los envíos redundantes, que se suelen hacer para asegurar la entrega de la información a su destino, aún cuando una posible ruta no pueda darse.

La figura 2.1 muestra como se establece la comunicación entre distintos puntos del planeta Tierra hasta llegar a comunicarse con un satélite en órbita geoestacio-

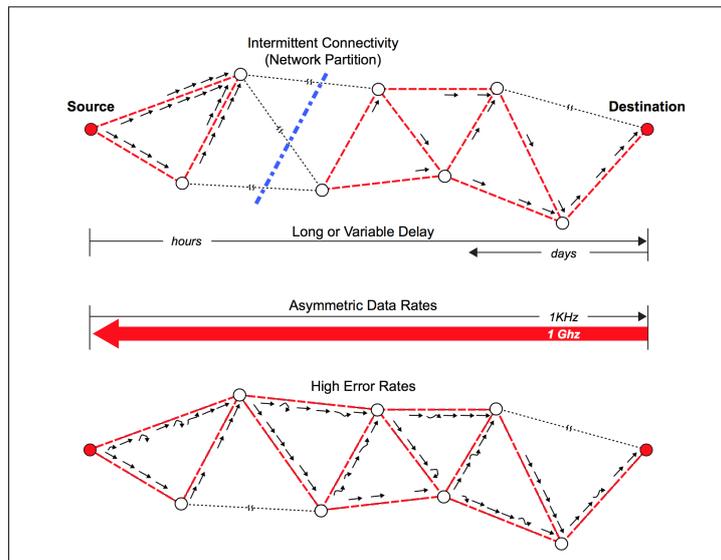


Figura 2.3: Red intermitente [1]

naria, mientras que la figura 2.2 muestra adicionalmente la comunicación que se puede dar entre distintos satélites ubicados en otros planetas del sistema solar.

En una red con varios nodos los cuales pueden transmitir la información es posible que uno de ellos falle o no este en una posición cercana, arriba de la figura 2.3 se muestra cómo una ruta podría no estar disponible en un determinado período de tiempo, mientras en la parte baja de la imagen se aprecia que durante otro lapso o período de tiempo el nodo podrá estar disponible, o haciendo uso de una ruta diferente.

Los protocolos de enrutamiento pueden garantizar que un mensaje llegue a su destino, siempre que exista una ruta disponible y que se use el protocolo de enrutamiento adecuado.

Los autores de [2] han realizado un estudio sobre varios de estos protocolos, los cuales pueden ser aplicados a las redes vehiculares.

En [9] se expone cual es el funcionamiento del protocolo y cómo se establecen los distintos procesos de comunicación y almacenamiento en una red DTN;

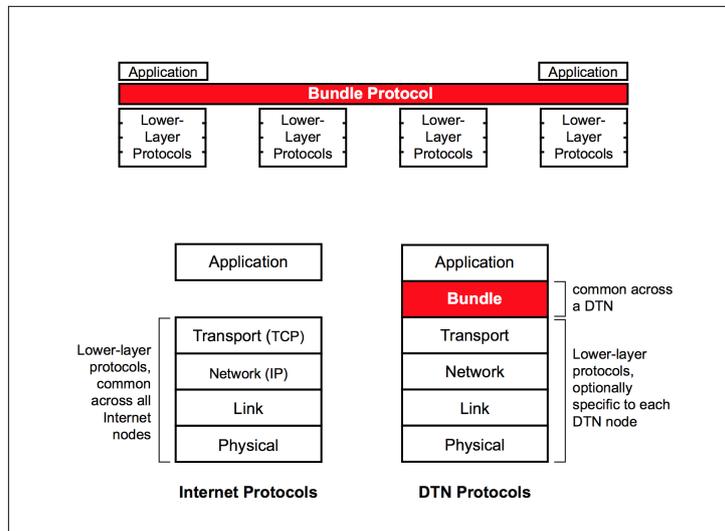


Figura 2.4: Capas de TCP y Bundle [1]

cual es el comportamiento en las distintas capas y módulos, y cómo influyen en el rendimiento de la red los distintos procesos como la fragmentación y encaminamiento.

Las DTN se pueden implementar en diferentes tecnologías inalámbricas como: WiFi, Bluetooth, incluyendo también las redes ópticas y las acústicas (acuáticas, debido a que las señales inalámbricas no se desplazan igual debajo del agua). Esto ha dado lugar a diferentes trabajos para diseñar e implementar aplicaciones con esta infraestructura de telecomunicaciones.

## 2.2 Protocolo Bundle

El modelo de transmisión de datos en internet se basa en la conmutación interactiva de paquetes, el cual demanda la existencia de un canal bidireccional, que debe ser continuo e ininterrumpido, desde el punto inicial al punto final.

Generalmente estos enlaces cuentan con bajas tasas de errores en la entrega de información, mientras que en DTN la transmisión de datos se basa en paquetes o

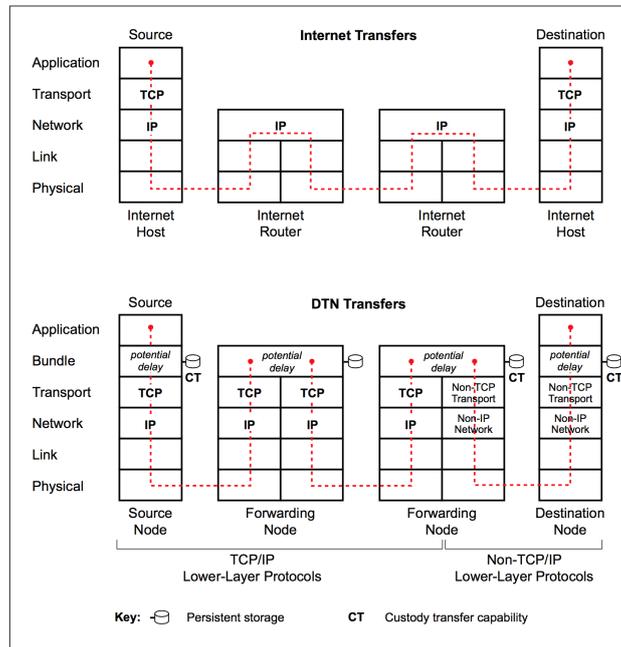


Figura 2.5: Reenvío de la información en TCP y DTN [1]

*bundles*, los cuales son recibidos y reenviados por los nodos.

El *Protocolo Bundle* se ejecuta bajo el método conocido como *Store and Forward*, el cual almacena en cada uno de los nodos el mensaje y luego lo reenvía, por lo que no es necesario garantizar que exista una conexión de principio a fin. El tiempo de entrega de la información se puede tornar algo impreciso, debido a que los canales de comunicación y los enlaces de datos son intermitentes por la movilidad que pueda existir entre los nodos.

La figura 2.3 muestra el motivo por el cual existe un retardo en la entrega, el que puede durar varios segundos, minutos o hasta días, dependiendo de la intermitencia de los enlaces o hasta que un nodo este disponible, por lo que no necesariamente el ancho de banda tiene que ser simétrico. A este proceso se le puede comparar con el correo convencional, solo ocurre cuando los nodos tienen alguna oportunidad de establecer la comunicación.

Basándonos en el modelo del Protocolo TCP/IP, Bundle funciona un nivel por

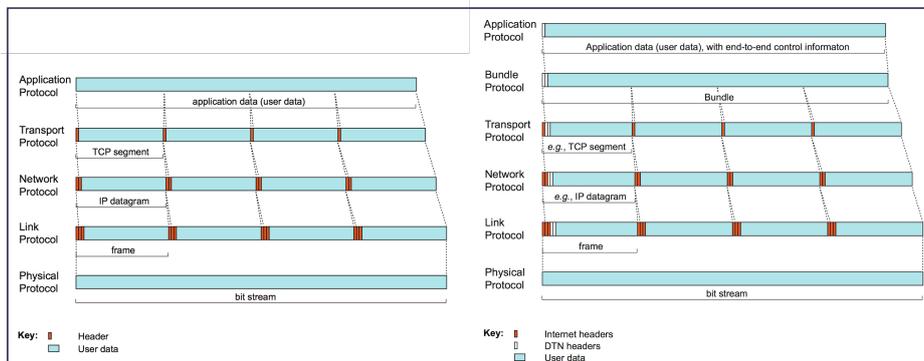


Figura 2.6: Segmentación de la información en TCP/IP y P. Bundles [1]

debajo de la capa de aplicación como se expone en [10], trabajando en conjunto con las capas inferiores, haciendo que la aplicación se pueda comunicar a través de las capas mas bajas, como se muestra en la figura 2.4.

La figura 2.5 es un esquema que explica el funcionamiento de como se realiza la transmisión y retransmisión de los datos en los protocolos TCP/IP y DTN, podemos apreciar los recursos usados para el reenvío de información en el modelo DTN puede existir el uso del almacenamiento persistente, mientras que en el modelo TCP/IP no es requerido.

Bundle se implementa mediante el uso de varios agentes los cuales se encargan del almacenamiento y reenvío de los mensajes entre los nodos. Así mismo como en el protocolo TCP donde los datos son segmentados en paquetes mas pequeños ocurre de manera similar en DTN, ver figura 2.6. Un agente puede fragmentar los datos en *bultos* o *bundles* los cuales serán re-ensamblados por este mismo agente cuando arriben a su destino como se explica en [11] y además se trata el tema de la compresión y de la identificación de los bundles. En el trabajo presentado en [12] se ofrece una referencia sobre la primera versión DTN.

Parte del rendimiento y éxito del protocolo se debe a lo que denominamos *almacenamiento o memoria persistente* que debe existir en los nodos involucrados

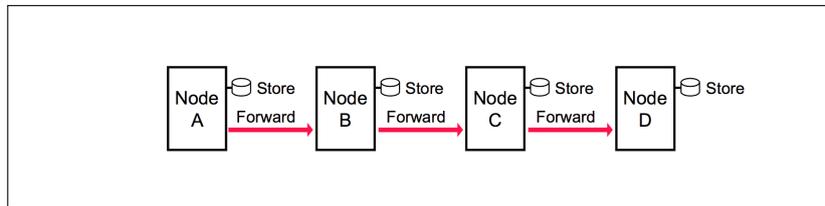


Figura 2.7: Almacenamiento persistente [1]

en este tipo de redes, observar la figura 2.7, mientras que en los encaminadores usados convencionalmente en el internet se usan chips de baja capacidad de almacenamiento.

El contacto en las redes DTN puede darse de dos maneras:

- Programada, en la que los nodos saben cuando y donde se producirá el contacto, por que disponen de información sobre la velocidad de desplazamiento y posición, un ejemplo de esto pueden ser los trenes de transporte publico, en todo momento un indicador nos muestra cuantos minutos faltan para que llegue el siguiente tren, así mismo ocurre con los aviones, buses entre otros.
- Oportunista, el contacto puede surgir en un momento y lugar no determinado, dependiendo del movimiento, alcance de la señal y de los nodos que puedan existir, formando de esta manera las *Redes Inalámbricas Oportunistas*.

En [13] los autores presentan un control de congestión, basado en estados de la red, además las propiedades de la red van disminuyendo a medida que estos retardos aumentan.

El reto para los investigadores esta en diseñar e implementar protocolos para considerar entornos donde exista poca o casi nada de infraestructura de transmisión de datos, ya que el proceso de comunicación en las redes oportunistas esta caracterizado por la intermitencia de los enlaces y la corta duración de estos con-

tactos.

## 2.3 Protocolos de Enrutamiento o Encaminamiento

A la hora de transmitir un mensaje por una red con distintos nodos, dispersos en distintas zonas geográficas, existen varios métodos para seleccionar cual será el mejor camino para que un mensaje llegue a su destino.

En DTN el método mas usado para compartir información en las redes oportunistas esta basado en la *diseminación epidémica* de los mensajes.

En el trabajo presentado en [14] los autores han realizado un estudio detallado del Enrutamiento Epidémico en redes inalámbricas en el que muestran distintos aspectos como el tamaño del *buffer*, el número de saltos hasta que el mensaje llegue a su destino o la distancia que hay entre los nodos, para poder así garantizar la entrega de los mensajes, minimizar la latencia, reducir los recursos consumidos y la seguridad en este tipo de redes.

Muchos de los trabajos sobre enrutamiento se basan en la movilidad de sus nodos, los que podrían ser coches como en [15] donde sus autores presentan algunos de estos protocolos de enrutamiento en las *redes vehiculares* tambien llamadas *VDTN*.

Los protocolos de encaminamiento en este tipo de redes, están habilitados para almacenar y reenviar la información entre dispositivos, esto define el modo en que los mensajes pueden ser distribuidos entre los nodos por ejemplo las distancias y tiempos de entrega, expuesto en [16] un trabajo que mide la polución del aire y las distintas trazas de trafico en el ambiente que se desarrolla.

Dependiendo del criterio y uso que le puede dar cada autor pueden existir varias clasificaciones de los protocolos de enrutameinto, los autores de [2] en su trabajo sobre los sistemas de transportes inteligentes han detallado la evolución de

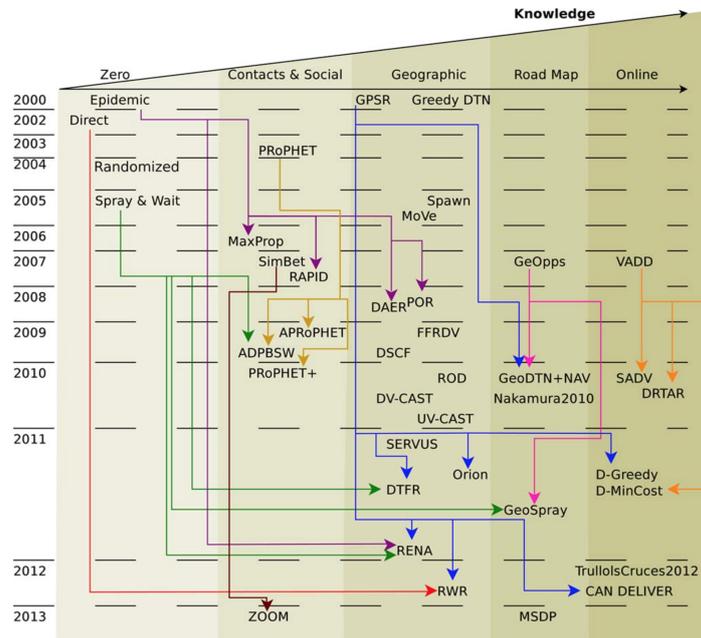


Figura 2.8: Cronología de los Protocolos de encaminamiento [2]

estos protocolos, ver la figura 2.8, además, han realizado experimentos y analizado las tasas de respuesta y envío de estos protocolos de enrutamiento.

En [17] los autores proponen una taxonomía basada en el reenvío de la información. Mientras que en [18] se plantea una clasificación de acuerdo al uso o no de la infraestructura de comunicaciones, además de hacer un estudio de casos reales como *The Hagggle Project* (<http://www.hagggleproject.org>).

Adicionalmente en [19] los autores han realizado un estudio sobre el uso de estos en redes con un ambiente vehicular en ciudades, además de plantear ciertas mejoras de encaminamiento y seguridad.

En [20] y [21] los autores resaltan la importancia de la privacidad en las redes inalámbricas oportunistas a la hora de descubrir nodos y reenviar información a los nodos correctos sobre todo si la información es de carácter confidencial. En [22] se realiza un estudio detallado de dos nuevos métodos, en los que se han

aplicado modelos matemáticos y reales teniendo en cuenta el tiempo y el espacio que pueden recorrer los nodos.

El proyecto ZebraNet analizado en [23] es un estudio detallado sobre como se pueden implementar las redes DTN y los protocolos de encaminamiento en un ambiente de vida salvaje, en este trabajo tambien se tiene en cuenta la energía que pueden consumir estos nodos, al ser portables y necesitar de una batería con un tiempo de carga determinado.

Los trabajos expuestos en esta sección ofrecen un punto de vista clave sobre los distintos temas que involucran este tipo de tecnología, desde la movilidad, rendimiento y seguridad que debe existir en los nodos y los enlaces, además, la latencia e intermitencia que caracterizan estas redes, siendo realmente el punto fuerte que tratamos en nuestro estudio.

## **2.4 DTN2**

En algunos trabajos se han realizado estudios sobre las implementaciones de DTN como en [9] donde se hace referencia a varios aspectos esenciales a la hora de transmitir los datos.

La implementación DTN2 proporciona un marco flexible para la realización de experimentos, puede ser configurado y manejado mediante archivos de configuración y líneas de comando / TCL.

Las extensiones para el enrutamiento, almacenamiento y las capas de convergencia son fácilmente configurables a través de interfaces XML. Proporciona un marco robusto y flexible para la experimentación, extensión, y el despliegue en el mundo real.

Cuenta con dos módulos diferentes, uno para el almacenamiento en memoria y otro para el almacenamiento en disco basado en *Berkeley DB Library*. El código

está equipado con registro, para efectos operativos como depuración.

En [24] se analizan los periodos de tiempo entre las conexiones y las tasas de entregas con la primer versión de DTN. En [25] se han analizado diferentes implementaciones del protocolo bundle, incluido DTN2, donde se puede apreciar el rendimiento que es posible obtener de él, IBR-DTN e ION, otras dos implementaciones muy populares.

### **2.4.1 Implementación**

La implementación de DTN2 esta escrita en C++, como requisito previo necesita de *Oasys* que esta diseñada para proporcionar una interfaz uniforme al código DTN2 [26]. Algunas de las funcionalidades de *Oasys* pueden resultar algo complejas de entender debido a sus intentos de ocultar los detalles específicos del sistema.

El Agente del Protocolo Bundle y todo su código de soporte se implementa como un demonio de usuario llamado *dtnd*. El demonio tiene una interfaz de configuración y control, la cual se puede ejecutar de forma remota a través de una conexión TCP cuando el demonio está en ejecución. La interfaz DTN para las aplicaciones de *dtnd* a través de la API son un mecanismo de llamada a procedimiento remoto.

### **2.4.2 Capacidades**

Esta aplicación tiene un conjunto bastante completo de las distintas funcionalidades de DTN, incluyendo la API de la aplicación, custodia de mensaje, apoyo inicial de algunos protocolos de seguridad DTN, acuses de recibo, varias capas de convergencia incluyendo TCP, Bluetooth y LTP (Protocolo de Transmisión Lic-klider), además una extensa colección de protocolos de enrutamiento, entre los

que se incluye:

- Enrutamiento Estático,
- Encaminamiento Epidemico - Inundación,
- Enrutamiento PRoPHET.

En [27] se describe el funcionamiento de la capa de convergencia LTP, el cual fue diseñado para proporcionar fiabilidad en la retransmisión de datos para DTN, especialmente para aquellos mensajes que en periodos de tiempo de ida y vuelta puedan resultar algo largos, en caso de que los enlaces tuvieran interrupciones frecuentes en su conectividad.

DTN2 utiliza almacenamiento persistente para mantener un estado previo en caso de que el *daemon* se detenga, para que los paquetes y alguna que otra información se pueden volver a cargar al reinicio. El envío de información se basa en una carga de trabajo util o *payload* la cual se envía al agente antes de pasar a la capa inferior donde se realiza la segmentación, en [28] se explica como estas cargas influyen en los envíos tal como se ve en la figura 2.6.

La arquitectura del protocolo DTN es presentada en [29], sus características estándar, como se realiza el almacenamiento de los datos y su funcionamiento.

En DTN2 se encuentran algunos mecanismos de almacenamiento, entre los que se incluyen:

- Unico Sistema de archivos,
- Berkeley DB (mecanismo básico, clave y valor)
- Memoria, rápida pero no persistente
- Interfaz ODBC para soporte de bases de datos SQL

- SQLite versión 3
- MySQL la versión 5.x

En la implementación se adjuntan algunos ejemplos de aplicaciones, como *ping* y *dtncp / dtncpd*, un cliente dtn para transferencia de archivos y su servidor.

Hay una interfaz para el bucle principal DTN2, que permite a los diversos componentes externos que se ejecuten en procesos separados de comunicación, con los principales procesos dtnd utilizando un protocolo basado en XML.

Se han implementado varios proyectos entre los que se pueden incluir componentes de router externo, como en este par de proyectos: RAPID (Resource Allocation Protocol for Intentional DTN) y HBSD (History Based Scheduling and Drop).

## Capítulo 3

### Descripción del Experimento

La presente sección esboza las características principales para la implementación y los puntos notables del diseño, resaltando algunas propiedades del demonio relevantes para nuestra evaluación, para poder evaluar el Protocolo Bundle en su implementación de código abierto DTN2, utilizando el Sistema Operativo Linux, en la cual se pueden centran diferentes aplicaciones.

En las redes con conexiones intermitentes el rendimiento es especialmente importante, el objetivo es tomar los beneficios de las conexiones transitorias tanto como sea posible, nos hemos centrado en el rendimiento que se puede lograr en diferentes escenarios oportunistas.

Hemos evaluado el rendimiento bruto utilizando una conexión Gigabit Ethernet, ancho de banda limitado, el rendimiento posible que se puede lograr cuándo se utiliza un emisor y un receptor, de que manera influye el tamaño de la carga útil a la hora de hacer las transferencias, con valores de pérdidas y retardos que pueden existir en la red.

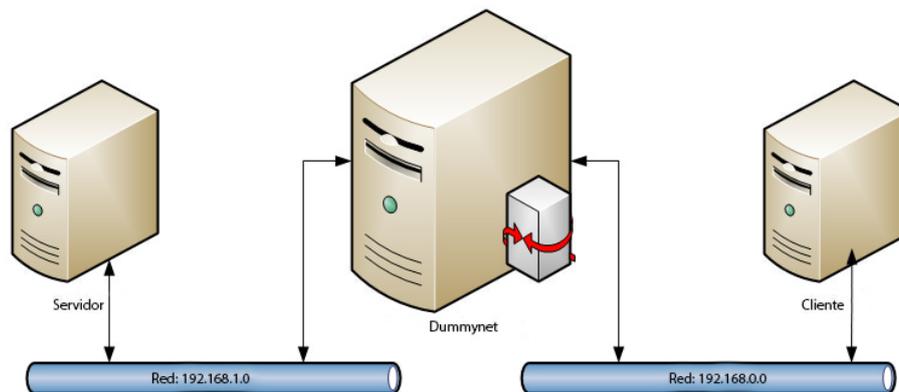


Figura 3.1: Esquema de Red

### 3.1 Equipos Utilizados

Para la implementación de las máquinas virtuales se ha utilizado una máquina huésped con las siguientes características:

- Procesador Intel Core i7-4870HQ CPU 2.5GHz
- 16 GB de memoria RAM
- Disco Duro de estado sólido - SSD
- OS X Yosemite Versión 10.10.5
- Virtual Box Versión 4.3.30

En la implementación se requirió dos redes, un nodo en cada red y un punto intermedio para comunicar ambas redes, ver figura 3.1. El nodo intermedio hace las labores de enrutamiento y de dummynet, como se explica en las líneas siguientes.

#### 3.1.1 Dummynet

Dummynet es una herramienta de emulación de red [30], diseñada originalmente para pruebas de protocolos (TCP y UDP). Dummynet automatiza lo que antes

se utilizaba con una serie de comandos para controlar herramientas tales como LARTC y Netem que gestionaban el ancho de banda, pérdida de paquetes, retardo en la entrega, y creación de colas de entrada y salida [31].

En nuestro experimento se ha instalado Dummynet en el nodo intermedio para hacer un puente entre los nodos de cada red y es donde se simularán los distintos parámetros de las mismas. Las especificaciones detalladas se pueden encontrar en <http://info.iet.unipi.it/luigi/dummynet/>.

Esta maquina posee las siguientes características:

- 1 GByte de memoria RAM
- 8 GBytes en Disco Duro
- SO Ubuntu Linux 12.04 LTS para 32 bits
- Dos adaptadores Intel PRO/1000 MT Desktop

### 3.1.2 Servidor y Cliente

**Cliente** Es el nodo encargado de enviar la información, donde se han configurado distintos tamaños de paquetes, además de distintos parametros para hacer las pruebas.

**Servidor** Será el nodo que recibirá los datos del cliente.

Ambos nodos tienen características similares a la del dummynet, pero nos hemos visto en la necesidad de aumentar la capacidad del disco duro, al hacer la transferencia de paquetes se solía llenar rápidamente. Las características se detallan a continuación:

- 1 GByte de memoria RAM

- 20 GBytes de Disco Duro
- SO Ubuntu Linux 14.04 LTS para 32 bits
- Un Adaptador Intel PRO/1000 MT Desktop

Para poder instalar la implementación DTN2 como se dijo en el capítulo anterior se necesita de algunos paquetes adicionales e instalar *Oasis* la versión 1.6.0. para posteriormente instalar *DTN2* en su versión 2.9.0, este proceso se realizó tanto para el servidor como para el cliente.

## 3.2 Descripción del Experimento

Antes de cada prueba se configuraron los parámetros que se requerían en la red, luego se iniciaba el demonio tanto en el cliente como servidor, comprobando que existiera conectividad con el demonio de la otra maquina, para posteriormente configurar en el dummynet las tramas perdidas, el retardo en la entrega y el ancho de banda.

Es muy importante verificar que exista comunicación entre el cliente y el servidor antes de hacer la configuración de los parámetros, para evitar malos entendidos en caso que no se llegarán a recibir paquetes o este tardando un poco en llegar el mensaje.

Posteriormente con la aplicación *dnperf* se procedió al envío de diferentes cargas de trabajo o *payload* que van desde los 10 KBytes a 50 Mbytes. Se realizaron diez pruebas para cada uno de los envíos, con el fin de obtener un valor más realista y a su vez permite establece un intervalo de confianza.

En todas y cada una de las pruebas se reinicio el demonio con la finalidad de que no existiera un caída sucesiva del rendimiento, en algunas pruebas anteriores los resultados decaían poco a poco si se realizaban pruebas consecutivas.

Se modificaron valores en el dummynet con diferentes tasas de pérdidas o *packet loss rate* que van desde el 5 % hasta el 25 %. Además, se cambiaron los parámetros de retardo o *delay* con valores de 10 milisegundos hasta los 200 ms.

Se utilizaron estas configuraciones con valores algo elevados, ya que la tasa de aciertos para entregar un mensaje dependerá del número de copias que se envíen, la cantidad de nodos que se encuentren en la red y permitan asegurar la entrega al nodo correcto, adicionalmente el retardo que puede surgir hasta que la entrega se haga efectiva suele ser elevado, aunque el enlace sea punto a punto, pueda llevar un poco de tiempo a que el nodo se encuentre en posición, por ejemplo el caso de un satélite si se tratase de una red interplanetaria.

La información con los datos obtenidos se guardó en un archivo para luego realizar el análisis pertinente.

# Capítulo 4

## Evaluación

En esta sección se evalúan los resultados obtenidos en el capítulo anterior, una vez realizadas las pruebas, antes de continuar es necesario acotar lo siguiente: se denomina *goodput* a la cantidad de información útil entregada por la red a un destino determinado, en unidad de tiempo. En la cantidad de datos que se consideran no se incluyen los gastos de protocolo ni los paquetes de datos retransmitidos.

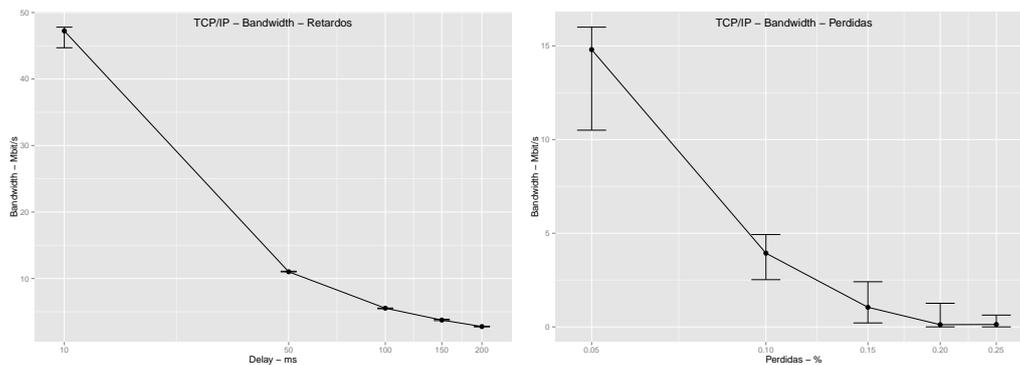


Figura 4.1: Bandwidth en TCP/IP inyectando Retardos y Perdidas

Perdida (%)	Max. Bandwidth (Mbit/s)	Bandwidth (Mbit/s)	Min. Bandwidth (Mbit/s)
5	16.00	14.80	10.50
10	4.93	3.93	2.52
15	2.41	1.05	0.21
20	1.26	0.12	0
25	0.63	0.12	0

Tabla 4.1: Bandwidth con Perdidas en TCP

Delay (ms)	Max. Bandwidth (Mbit/s)	Bandwidth (Mbit/s)	Min. Bandwidth (Mbit/s)
10	47.80	47.23	44.70
50	11.10	11.02	11.00
100	5.56	5.54	5.45
150	3.88	3.76	3.67
200	2.83	2.78	2.73

Tabla 4.2: Bandwidth con Retardos en TCP

## 4.1 Resultados

Después de haber realizado las pruebas de bundles enviados y el goodput, se guardaron los resultados en un archivo .csv para ser organizados, estudiados y graficados.

Esta información esta representada de la siguiente manera: en el eje  $x$  se encuentra el tamaño de la carga útil o payload debido a que el protocolo tiene un comportamiento muy importante basado en el tamaño de estas cargas, en el eje  $y$  los valores correspondientes a los bundles enviados o goodput obtenido.

Las imágenes se encuentran organizadas de tal manera que paralelamente se vean dos, del lado izquierdo se muestran el número de bundles enviados en cada sesión de pruebas, y del otro lado el goodput.

En las siguientes secciones se muestran los resultados del protocolo TCP, posteriormente las imágenes correspondientes a los resultados DTN con sus variaciones de pérdidas y retardos, finalmente hacemos unas pruebas mezclando configuraciones de pérdidas y retardos.

#### **4.1.1 Resultados TCP/IP**

Antes de mostrar los valores con el protocolo bundle es muy importante tener una idea como los retardos y las pérdidas influyen negativamente en la red. En las mediciones realizadas se pudo obtener un valor máximo de ancho de banda para el modelo TCP de *990 Mbit/sec*, teniendo en cuenta que nuestra implementación utiliza para su comunicación una red GigaBit Ethernet, estaríamos haciendo uso del 99 % del ancho de banda o *bandwidth*.

La figura 4.1 muestra el ancho de banda que se tiene en nuestra red con el modelo TCP aplicando diferentes parámetros de pérdidas y de retardos. Podemos observar que entre más grandes sean los valores de las tasas de pérdidas y los milisegundos de retardos, el rendimiento de la red será influenciado negativamente, y el ancho de banda decrecerá poco a poco, disminuyendo de esta manera las prestaciones de nuestros enlaces.

La tabla 4.1 muestra los valores promedios del bandwidth cuando se producen pérdidas en la red, sus máximos y mínimos obtenidos en esta prueba. Mientras que en la tabla 4.2 se aprecian los valores del ancho de banda cuando existen retardos en la red.

De acuerdo a estos resultados podemos acotar que cuando tenemos una pérdida del 5 % de los datos lo máximo que podría llegar a transmitirse en la red

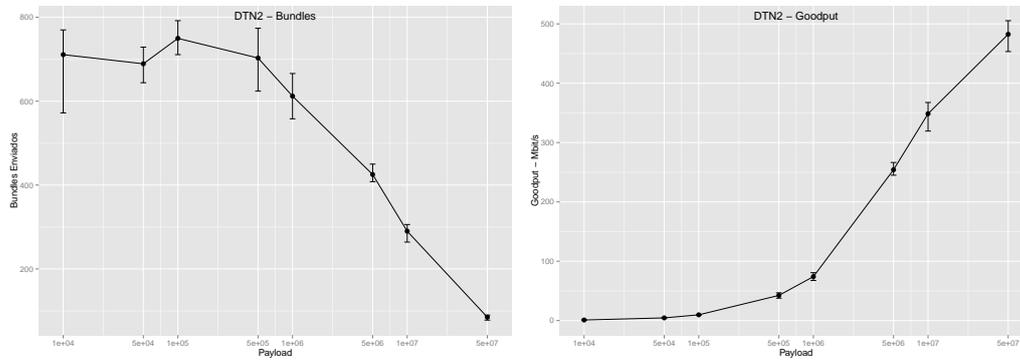


Figura 4.2: DTN2 Bundles - Goodput

Payload (Bytes)	Max B. (n°)	Bundles (n°)	Min B. (n°)	Max G. (Mbit/s)	Goodput (Mbit/s)	Min G. (Mbit/s)
1e+04	770	711	572	1.01	0.93	0.75
5e+04	729	690	644	4.76	4.51	4.21
1e+05	792	750	711	9.89	9.67	9.32
5e+05	774	703	624	46.73	42.39	37.68
1e+06	666	612	558	80.69	74.03	67.61
5e+06	450	425	408	266.40	254.20	244.95
1e+07	306	290	264	367.55	348.62	319.34
5e+07	90	85	78	505.42	482.50	453.36

Tabla 4.3: Bundles y Goodput en DTN2

es de 16 Mbit/s, mientras que si las pérdidas llegaran al 25 % se podrían llegar a transmitir hasta 630 Kbit/s. En el caso de los valores extremos de estas pruebas (25 % de pérdidas y los 200 milisegundos de retardos) existe poco ancho de banda, la transmisión de datos en estos casos extremos es muy poca, pero veremos a continuación que usando DTN en las siguientes pruebas los resultados mejorarán notablemente.

#### **4.1.2 Resultados DTN2**

En los resultados siguientes tenemos una idea del comportamiento del protocolo bundle, las pruebas fueron realizadas sin aplicar ningún retardo o pérdida en la red, la figura 4.2 muestra a su derecha como a un número más alto de payload o carga útil automáticamente se ve reflejado en un aumento del goodput en la red, debido a que entre más grande sea el payload va a existir menos trabajo a la hora de hacer el segmentado, reenvío y ACK como aspectos principalmente influyentes; mientras que a la izquierda se muestra como los bundles disminuyen, esto es normal, básicamente entre mayor sea el tamaño de la carga útil menos bundles van a ser necesarios para el envío, además como se dijo en líneas anteriores el trabajo de fragmentación y envío llevará un poco de tiempo.

La tabla 4.3 contiene los bundles y goodput obtenidos, así como los valores máximos y mínimos comprobamos que a medida que los paquetes o bundles son más grandes aumenta significativamente el goodput, lógicamente entre mayor sea la carga son más los datos enviados y se minimiza el trabajo que ocurre por debajo en la capa del protocolo bundle.

Como ya se ha explicado anteriormente en cada una de las pruebas se detuvo el demonio y se volvió a iniciar para asegurar que no existieran rastros de las pruebas anteriores, aun así algunos valores pueden resultar algo imprecisos como ha ocurrido en los valores de los payload de 10KB, 50KB, 100KB y 150KB de los

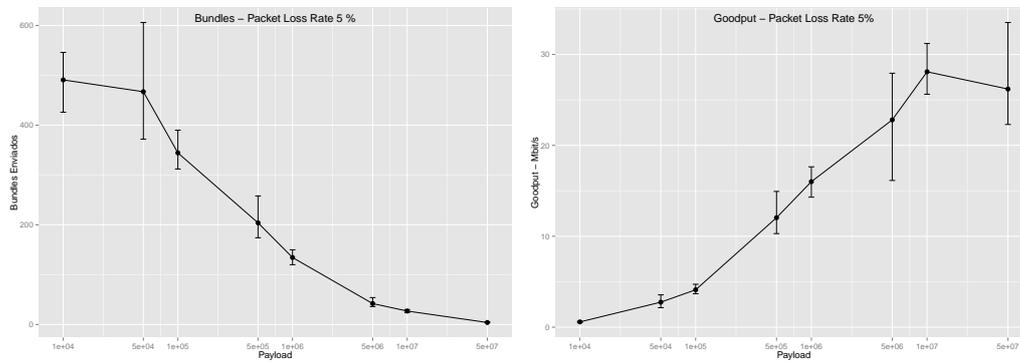


Figura 4.3: Bundles - Goodput con Perdidas del 5 %

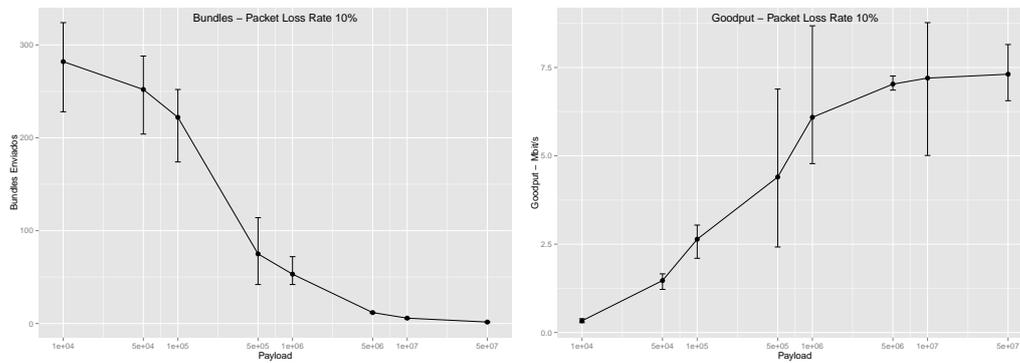


Figura 4.4: Bundles - Goodput con Perdidas del 10 %

bundles enviados como muestra la figura 4.2 donde se puede ver como la línea cae un poco y luego vuelve a subir, los valores promedio para estas cargas son 711, 690, 750 y 703 bundles enviados respectivamente, lo cual al menos en estas cargas de trabajo más pequeñas se puede dar, la razón radica en el proceso de dividir y segmentar estos paquetes para poder ser enviados, desemboca en una carga extra de trabajo antes de que se pueda realizar el envío.

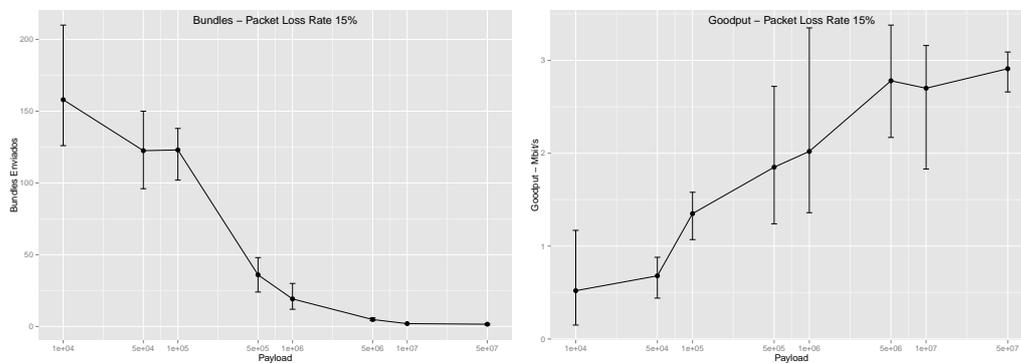


Figura 4.5: Bundles - Goodput con Perdidas del 15 %

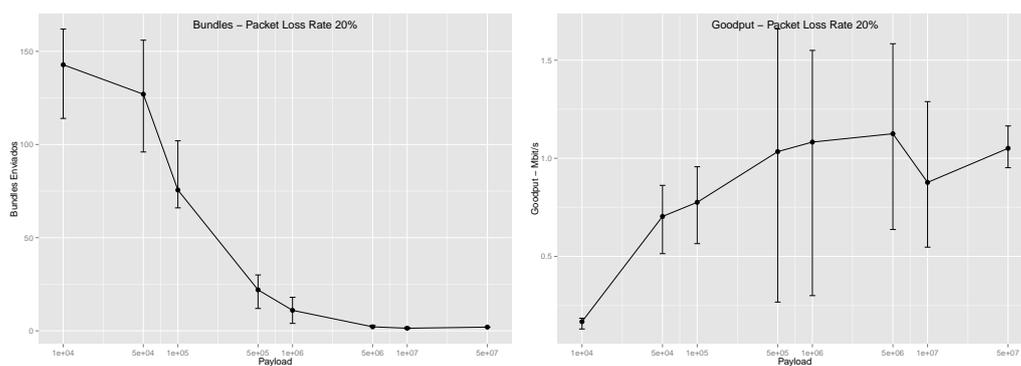


Figura 4.6: Bundles - Goodput con Perdidas del 20 %

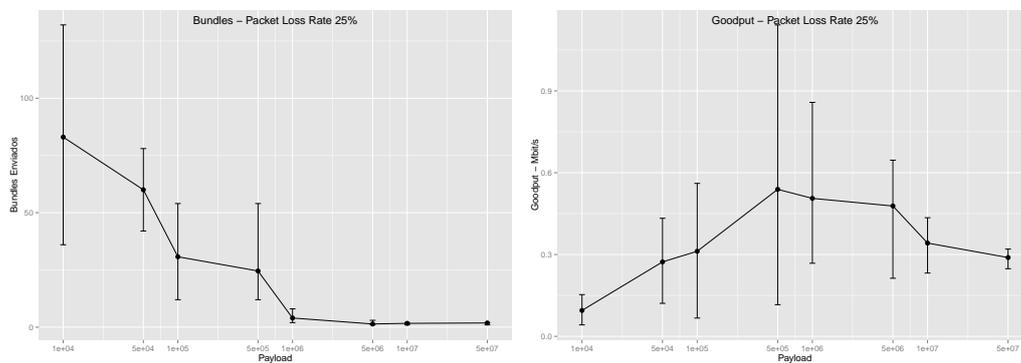


Figura 4.7: Bundles - Goodput con Perdidas del 25 %

### **4.1.3 Resultados DTN2 con Perdidas**

En las figuras 4.3 a la 4.7 se puede apreciar el comportamiento del protocolo DTN2 cuando existen perdidas en la red. Se puede comprobar que a mayor porcentaje de perdidas las cualidades de la red disminuyen notablemente, pero no en gran medida como en las pruebas realizadas bajo el protocolo TCP, DTN2 mejoró los resultados, algo que ya se expuso en los párrafos anteriores en los casos extremos en TCP se obtenían ceros o resultados muy bajos, mientras que con DTN2 se obtuvieron resultados en los que si había transmisión y entrega de datos.

Algo que tambien se puede apreciar es que en los resultados de payload intermedios los valores máximos y mínimos tienen un rango muy amplio, además llegado a ese punto las prestaciones comienzan a decaer.

Debido a esto para las pruebas en las que mezclamos perdidas y reatrdos se realizaron con los valores de carga de trabajo de 100KB, 500KB y 1MB que a parte de ser los valores intermedios de las pruebas, y ser los que presentan mayor variabilidad, tambien son un punto medio para los posibles datos que se enviarán en las transmisiones de una aplicación real.

### **4.1.4 Resultados DTN2 con Retardos**

En estas pruebas los resultados fueron mucho mejores que en la prueba anterior, en las figuras 4.8 hasta la 4.12 se puede apreciar que cuando existen retardos en la red las cualidades de la red disminuyen, esto se debe básicamente a que del mismo modo que ocurre en TCP hay que esperar una respuesta o un *ACK* en este caso un *ARQ* en base a la implementación LTP, que nos asegura la entrega del mensaje, implementando el almacenamiento persistente los resultados mejorarían, como nuestra red solo consta de dos puntos y al no existir nodos o caminos adicionales el resultado puede no quedar tan claro.

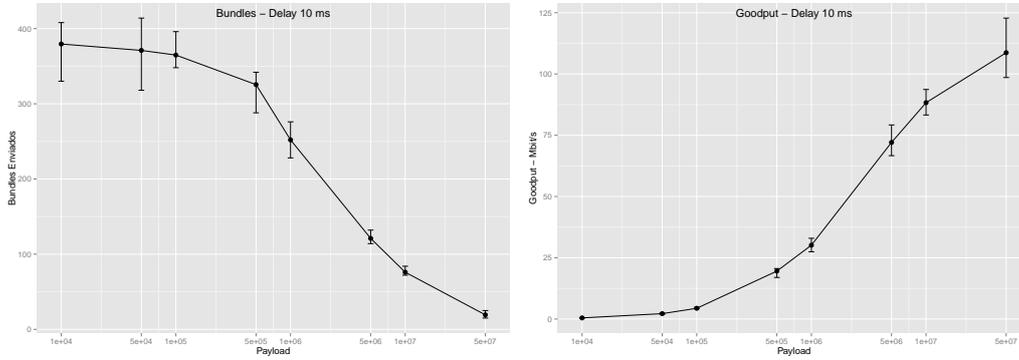


Figura 4.8: Bundles - Goodput con Retardo de 10ms

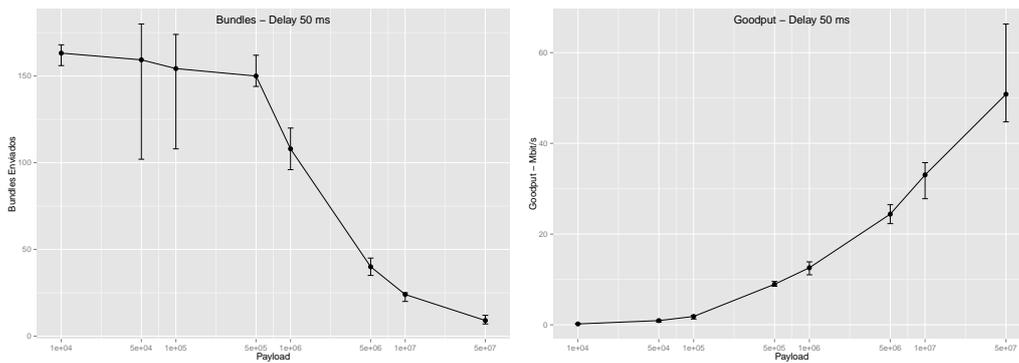


Figura 4.9: Bundles - Goodput con Retardo de 50ms

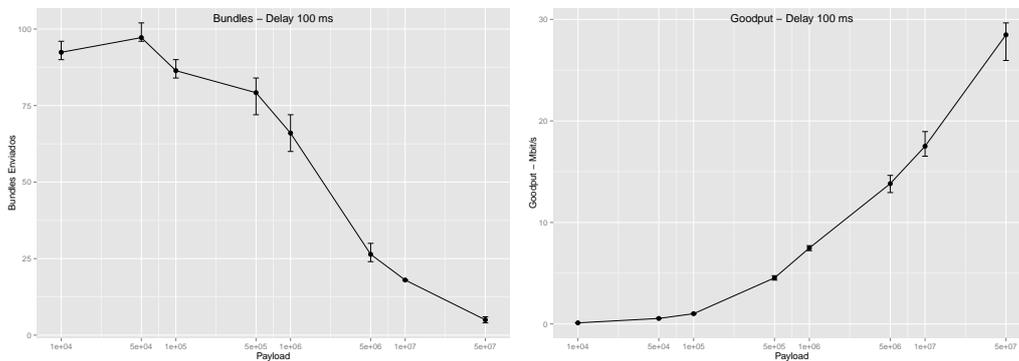


Figura 4.10: Bundles - Goodput con Retardo de 100ms

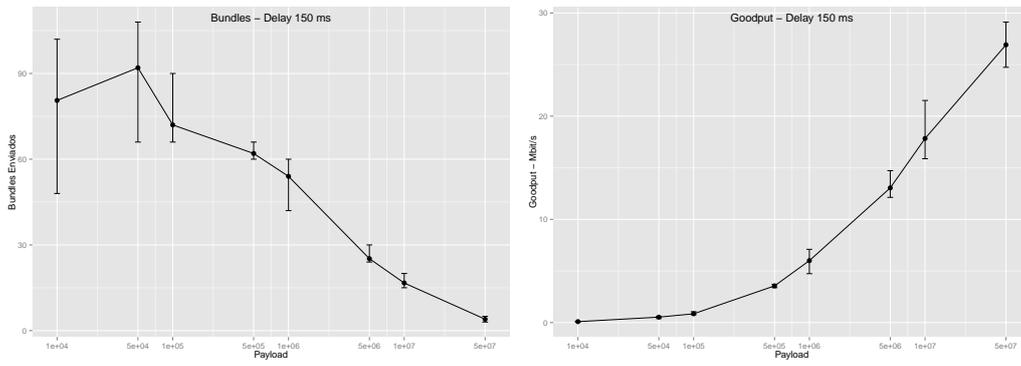


Figura 4.11: Bundles - Goodput con Retardo de 150ms

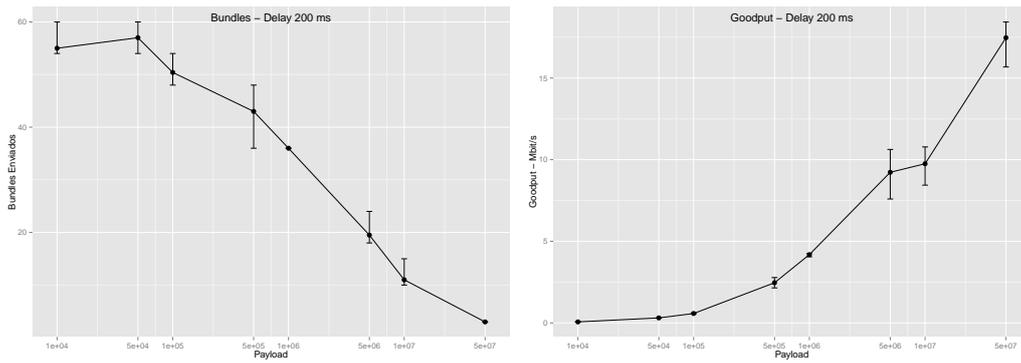


Figura 4.12: Bundles - Goodput con Retardo de 200ms

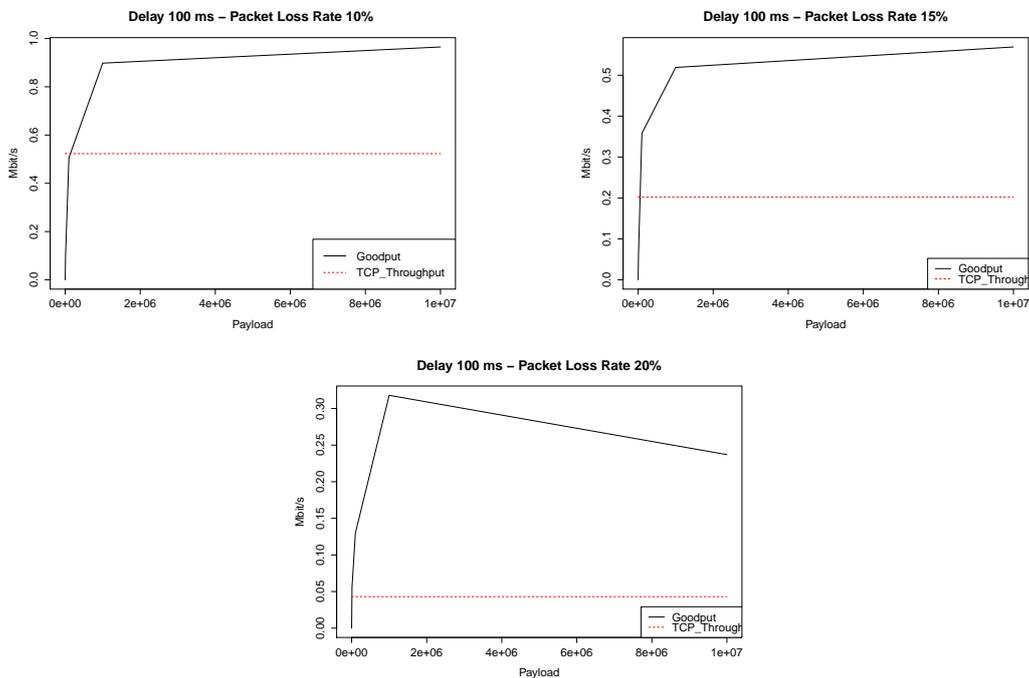


Figura 4.13: Goodput - Throughput para Retardos de 100ms con Perdidas

En los resultados de la figura 4.11 y 4.12 no ocurre lo mismo que en las pérdidas, el rango de los bundles enviados o el goodput podría resultar demasiado amplio, al contrario se ven relativamente uniformes si los comparamos con las figuras anteriores.

#### 4.1.5 Resultados DTN2 con Perdidas y Retardos en la red

Después de haber analizado el comportamiento de la red de manera individual para las pérdidas y los retardos, se procedió a realizar pruebas más reales la cual puede sufrir pérdidas y retardos al mismo tiempo, que es lo que suele ocurrir en un ambiente oportunista.

Las figuras 4.13, 4.14 y 4.15 se muestra cual es el comportamiento de la red para los retardos de 100, 150 y 200 milisegundos respectivamente, en cada una

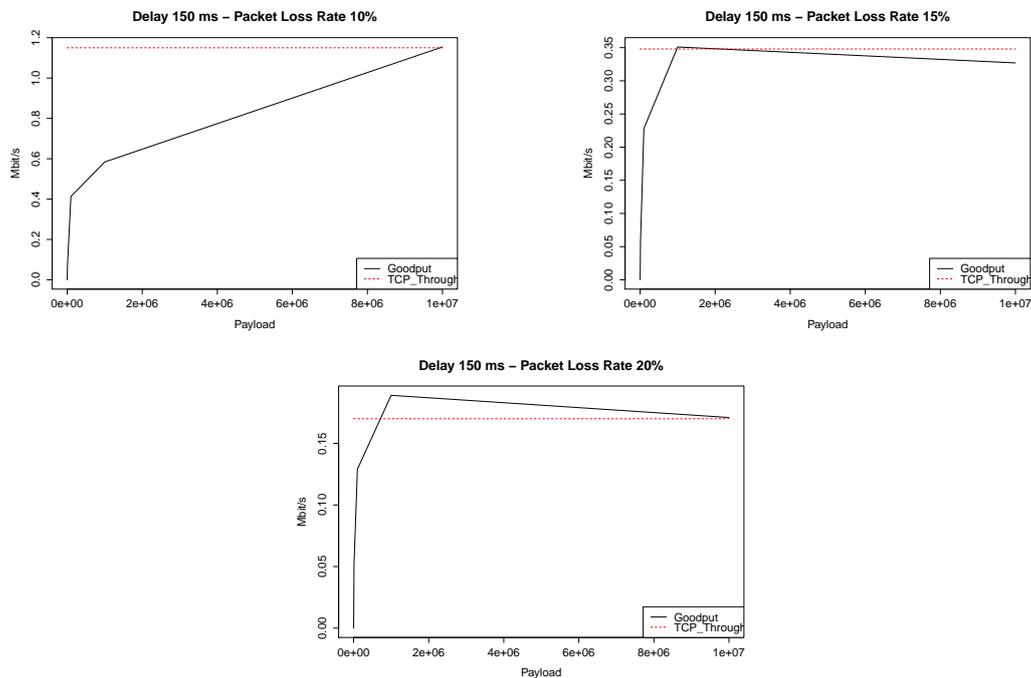


Figura 4.14: Goodput - Throughput para Retardos de 150ms con Perdidas

de ellas se encuentran tres imágenes de las pérdidas de 10, 15 y 20 por ciento, hemos escogido estas configuraciones para hacer pruebas adicionales al ser las que tienen mejores resultados de forma general en pruebas anteriores, además mostraban variabilidad en el intervalo de confianza.

Los resultados de DTN2 se muestran con una línea de color negro mientras los de TCP con la línea segmentada de color rojo la cual se pueden ver totalmente horizontal, debido a que el eje  $x$  esta en función de las cargas útiles o payload, por lo que los resultados de TCP no aplican para esta configuración, pero se ha colocado para analizar el comportamiento con los resultados de DTN.

En varios de los experimentos DTN2 supero los valores de TCP, pero en otras no es así, este comportamiento no se puede explicar con claridad ya que no existe un patrón que nos permita identificar a que se debe, por ejemplo en la figura 4.14 muestra los resultados de 150 ms de retardo, TCP supera en casi todas las

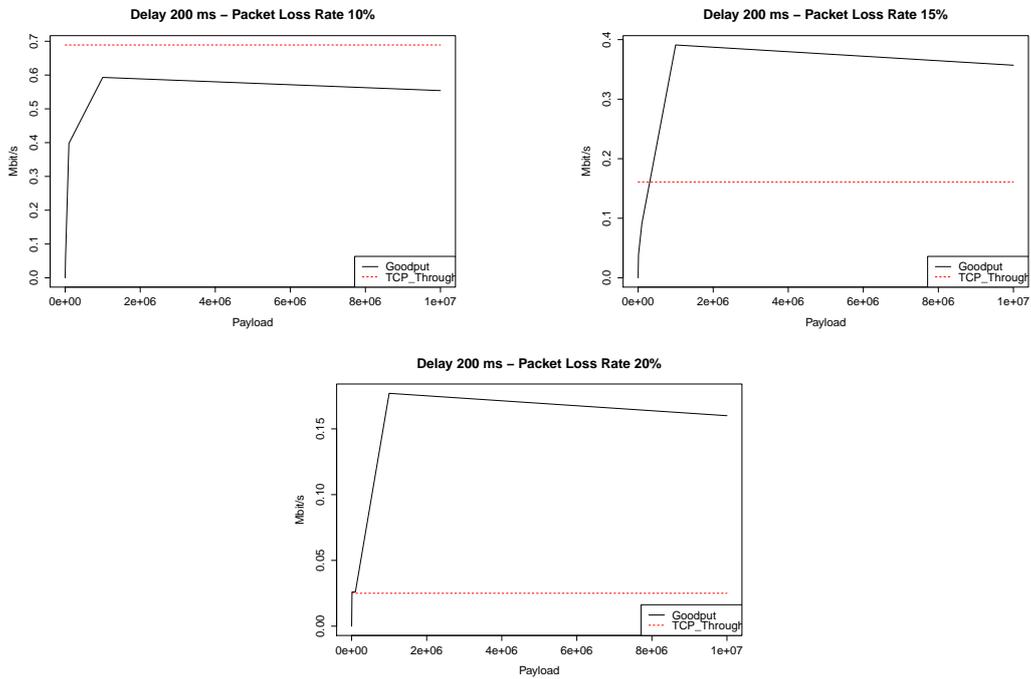


Figura 4.15: Goodput - Throughput para Retardos de 200ms con Perdidas

configuraciones de perdidas, pero en las otras dos figuras no es así.

Para terminar esta sección podemos decir que dependiendo del escenario en el que vamos a usar nuestra aplicación sobre DTN2, buscar la configuración que mas se amolde a nuestros requerimientos o los valores de retardos y perdidas que puedan existir en nuestra red.

# Capítulo 5

## Conclusiones

### 5.1 Conclusiones

El principal objetivo de este trabajo ha sido realizar un amplio estudio del Protocolo Bundle e implementar DTN2 para posteriormente efectuar experimentos para medir el rendimiento en las transferencias llevadas a cabo en escenarios oportunistas.

Después de haber inyectado retardos y pérdidas en la red, hemos sido capaces de darnos cuenta de la factibilidad del uso de este tipo de implementaciones, sobre todo en aquellos escenarios en los cuales el ambiente, el desplazamiento de los nodos o la no existencia de una infraestructura fija, puedan hacer que las características de la red no sean las mas óptimas para la transmisión de los datos.

Las aplicaciones actuales en las que se puede hacer uso de este tipo de implementaciones son muchas: sitios donde la infraestructura sea escasa o casi nula, pudiendo ir desde zonas rurales hasta el espacio exterior.

Adicionalmente se pueden usar bajo distintas tecnologías de transmisión de datos, por lo que aumenta significativamente el campo de investigación y desarrollo de las Redes Oportunistas. Los usuarios de DTN quieren elegir su aplicación

en función del escenario de uso, mientras que los desarrolladores quieren lograr la mejor facilidad para un escenario específico.

## **5.2 Trabajo futuro**

Esta tesis formará parte de un trabajo más amplio con la finalidad de desembocar en un trabajo doctoral, por lo que a mediano y largo plazo nuestro objetivo principal realizando este tipo de análisis es poder proponer mejoras en el protocolo bundle y la implementación DTN2, además de llegar a desarrollar aplicaciones y optimizar las ya existentes, sin que existan pérdidas en sus funcionalidades.

Dentro de ese marco las siguientes tareas que tenemos pendientes desarrollar serían las siguientes: mejora de las herramientas de DTN2, una de las primeras acciones que se pretende llevar a cabo es la mejora en los algoritmos de encaminamiento en escenarios oportunistas. Explorar otras propuestas, sería muy importante comparar estos resultados con otras implementaciones del Protocolo Bundle. Adaptación del protocolo para aplicaciones específicas, exploraremos las posibilidades en la utilización de DTN2 en aplicaciones realmente prácticas de acuerdo al campo en que pretendan desarrollarse.

Diseñar e implementar aplicaciones eficientes en las que exista colaboración entre los distintos dispositivos, entre los que se podrían incluir los teléfonos móviles y las Raspberry Pi para compartir información de diversa índole dependiendo del campo tecnológico en el que se despliegue.

# Bibliografía

- [1] F. Warthman *et al.*, “Delay-and disruption-tolerant networks (dtns),” *A Tutorial. V 2.0, The InterPlaNetary (IPN) Internet Project. Interplanetary Internet Special Interest Group*, 2012.
- [2] S. Tornell, C. T. Calafate, J. C. Cano, and P. Manzoni, “Dtn protocols for vehicular networks: an application oriented overview,” *IEEE Communications Surveys & Tutorials*, pp. 1–10, 2015.
- [3] S. Ferretti, “Shaping opportunistic networks,” *Computer Communications*, vol. 36, no. 5, pp. 481–503, 2013.
- [4] C.-M. Huang, K.-c. Lan, and C.-Z. Tsai, “A survey of opportunistic networks,” in *Advanced Information Networking and Applications-Workshops, 2008. AINAW 2008. 22nd International Conference on.* IEEE, 2008, pp. 1672–1677.
- [5] K. L. Scott and S. Burleigh, “Bundle protocol specification,” *RFC 5050 (Experimental)*, 2007.
- [6] A. Keränen, T. Kärkkäinen, and J. Ott, “Simulating mobility and dtns with the one,” *Journal of Communications*, vol. 5, no. 2, pp. 92–105, 2010.
- [7] J. Niu, J. Guo, Q. Cai, N. Sadeh, and S. Guo, “Predict and spread: An efficient routing algorithm for opportunistic networking,” in *Wireless Commu-*

- nications and Networking Conference (WCNC), 2011 IEEE.* IEEE, 2011, pp. 498–503.
- [8] G. S. Thakur, U. Kumar, A. Helmy, and W.-J. Hsu, “On the efficacy of mobility modeling for dtn evaluation: Analysis of encounter statistics and spatio-temporal preferences,” in *Wireless Communications and Mobile Computing Conference (IWCMC), 2011 7th International.* IEEE, 2011, pp. 510–515.
- [9] M. Demmer, E. Brewer, K. Fall, M. H. Jain, and R. Patra, “Implementing delay tolerant networking,” *Technical report, IRB-TR-04-020*, Dec. 2004.
- [10] J. Ott, M. Demmer, and S. Perreault, “Delay-tolerant networking tcp convergence-layer protocol,” *IEFT Draft*, Nov. 2014.
- [11] S. Burleigh, “Compressed bundle header encoding (cbhe),” *IETF Draft*, Feb. 2011.
- [12] S. Burleigh, A. Hooke, L. Torgerson, K. Fall, V. Cerf, B. Durst, K. Scott, and H. Weiss, “Delay-tolerant networking: an approach to interplanetary internet,” *Communications Magazine, IEEE*, vol. 41, no. 6, pp. 128–136, 2003.
- [13] S. Burleigh, E. Jennings, and J. Schoolcraft, *Autonomous congestion control in delay-tolerant networks.* Pasadena, CA: Jet Propulsion Laboratory, National Aeronautics and Space Administration, 2006.
- [14] A. Vahdat, D. Becker *et al.*, “Epidemic routing for partially connected ad hoc networks,” Duke University, Tech. Rep. CS-200006, 2000.
- [15] N. Benamar, K. D. Singh, M. Benamar, D. El Ouadghiri, and J.-M. Bonnin, “Routing protocols in vehicular delay tolerant networks: A comprehensive survey,” *Computer Communications*, vol. 48, pp. 141–158, 2014.

- [16] S. Lahde, M. Doering, W.-B. Pöttner, G. Lammert, and L. Wolf, “A practical analysis of communication characteristics for mobile and distributed pollution measurements on the road,” *Wireless Communications and Mobile Computing*, vol. 7, no. 10, pp. 1209–1218, 2007.
- [17] B. Poonguzharselvi and V. Vetriselvi, “Survey on routing algorithms in opportunistic networks,” in *Computer Communication and Informatics (ICCCI), 2013 International Conference on.* IEEE, 2013, pp. 1–5.
- [18] L. Pelusi, A. Passarella, and M. Conti, “Opportunistic networking: data forwarding in disconnected mobile ad hoc networks,” *Communications Magazine, IEEE*, vol. 44, no. 11, pp. 134–141, 2006.
- [19] S. M. Tornell, C. T. Calafate, J.-C. Cano, and P. Manzoni, “Assessing the effectiveness of dtn techniques under realistic urban environments,” in *Local Computer Networks (LCN), 2013 IEEE 38th Conference on.* IEEE, 2013, pp. 573–580.
- [20] E. M. Goyal and E. M. Chaudhary, “Ensuring privacy in opportunistic network,” *International Journal of Computer Applications*, vol. 76, no. 4, pp. 11–17, 2013.
- [21] L. Lilien, Z. H. Kamal, V. Bhuse, and A. Gupta, “Opportunistic networks: the concept and research challenges in privacy and security,” *Proc. of the WSPWN*, pp. 134–147, 2006.
- [22] E. Hernández-Orallo, J.-C. Cano, C. T. Calafate, and P. Manzoni, “A representative and accurate characterization of inter-contact times in mobile opportunistic networks,” in *Proceedings of the 16th ACM international conference on Modeling, analysis & simulation of wireless and mobile systems-MSWiM’13.* ACM, 2013, pp. 309–316.

- [23] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. S. Peh, and D. Rubenstein, “Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with zebranet,” *ACM Sigplan Notices*, vol. 37, no. 10, pp. 96–107, 2002.
- [24] E. Oliver and H. Falaki, “Performance evaluation and analysis of delay tolerant networking,” in *Proceedings of the 1st international workshop on System evaluation for mobile platforms, MobiEval’07*. ACM, 2007, pp. 1–6.
- [25] W.-B. Pöttner, J. Morgenroth, S. Schildt, and L. Wolf, “Performance comparison of dtn bundle protocol implementations,” in *Proceedings of the 6th ACM workshop on Challenged networks*. ACM, 2011, pp. 61–64.
- [26] O. O. Ajayi, “Dynamic trust negotiation for decentralised e-health collaborations,” Ph.D. dissertation, University of Glasgow, 2009.
- [27] M. Ramadas, S. Burleigh *et al.*, “Licklider transmission protocol-specification,” *RFC 5326(Experimental)*, Sep. 2008.
- [28] G. Papastergiou, I. Alexiadis, S. Burleigh, and V. Tsaoussidis, “Delay tolerant payload conditioning protocol,” *Computer Networks*, vol. 59, pp. 244–263, 2014.
- [29] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss, “Delay-tolerant networking architecture,” Tech. Rep., April 2007.
- [30] L. Rizzo, “Dummynet: a simple approach to the evaluation of network protocols,” *ACM SIGCOMM Computer Communication Review*, vol. 27, no. 1, pp. 31–41, 1997.

- [31] J. Luzuriaga *et al.*, “Descarga de computación de dispositivos móviles a ambientes cloud computing en un caso en concreto, el reconocimiento facial,” Valencia, 2012.