Delay Tolerant Network para la red interplanetaria

Carlos Lozano García
Grau en Enginyeria de Tecnologies i Serveis de Telecomunicació
11.625 Administració de xarxes i sistemes operatius

Consultor: Mario Prieto Vega
Profesor: Jordi Serra Ruiz

6 de junio de 2021

Aquesta obra està subjecta a una llicència de Reconeixement-NoComercial-SenseObraDerivada 3.0 Espanya de Creative Commons

FICHA DEL TRABAJO FINAL

Título del trabajo:	*Delay Tolerant Network para la red interplanetaria*
Nombre del Autor:	*Carlos Lozano García*
Nombre del consultor:	*Mario Prieto Vega*
Nombre del PRA:	*Jordi Serra Ruiz*
Fecha de entrega:	*06/2021*
Titulación:	Grau en Enginyeria de Tecnologies i Serveis de Telecomunicació
Área del Trabajo Final:	*Administració de xarxes i sistemes operatius*
Idioma del trabajo:	*Castellà*
Palabras clave	*DTN (Delay/Disruption Tolerant Network)* *IPN (Interplanetary Network)* *SSI (Solar System Internet)* *Space Networking*

Resumen del Trabajo:

La industria espacial es un sector que está experimentando una importante evolución, impulsada por nuevos desafíos y la creciente participación del sector privado. Es un escenario que genera nuevas necesidades en las redes de telecomunicaciones, y requiere soluciones para problemas inmediatos y futuros. Extender el concepto de Internet para idear una red interplanetaria, promueve nuevas tecnologías que se adapten a las características de las comunicaciones espaciales, donde Delay/Distruption Tolerant Networking (DTN) es la propuesta para estas redes en el Sistema Solar. En este proyecto, se estudia el proceso de propuesta de esta arquitectura, y el recorrido trazado a lo largo de los años hasta convertirse en una tecnología viable. En esta primera parte, se sigue una metodología secuencial siguendo los pasos cronológicos. En una segunda parte, se trabaja en un entorno práctico, profundizando de forma iterativa e incremental en el uso del software experimental. De esta forma, se investigan las implementaciones y sus características, además de sus usos en proyectos de pruebas en entornos reales. Finalmente, se experimenta con ION -la implementación DTN desarrollada por JPL/NASA-, y se presenta un escenario de pruebas y de emulación, junto con varias herramientas de análisis. En el trabajo se han realizado diferentes pruebas de uso y configuración. Como resultado del proyecto, se pretende transmitir las particularidades de la red DTN y su procedimiento de uso en un escenario de red interplanetaria.

Abstract:

The space industry is a sector that is undergoing a growing evolution, driven by new challenges and the increasing participation of the private sector. It is a scenario that generates further needs in telecommunications networks and requires solutions for immediate and future problems. Extending the concept of the Internet to devise an interplanetary network promotes new technologies that adapt to the characteristics of space communications, where Delay / Disruption Tolerant Networking (DTN) is the proposal for these networks in the Solar System. In this project, the proposal process of this architecture is detailed, and the route traced over the years until it becomes a viable technology. First, a sequential methodology is followed by understanding the chronological steps. In the second part, it works in a practical environment. Afterwards, the work deepening iteratively and incrementally in experimental software frameworks. Implementations and their characteristics are research, as well as their uses in test projects in real-life environments. Finally, ION - the DTN implementation developed by JPL / NASA - is experimented with, and it shows a testbed and emulation scenario, along with various analysis tools. The example has performed several testing and configurations. As a result of the project, it aims to convey the particularities of the DTN network and its use procedure in an interplanetary network scenario.

A mi esposa por su apoyo incondicional.

A mis hijas por sus ánimos y sus alentadores “tu puedes papá!”.

A mis padres y mis hermanos que siempre han creído en mi.

A mis amigos y mi socio por ayudarme a superar los baches del camino.

Agradecer a ELETE y todo su equipo por los medios cedidos durante el proyecto. (http://elete.es).

Agradecer a todos los profesores, consultores y a mi tutor de la Univesitat Oberta de Catalunya, por acompañarme durante todos estos años de aprendizaje y esfuerzo.

Índice de contenidos

Tabla de contenido

ÍNDICE DE CONTENIDOS **7**

ACRÓNIMOS **9**

LISTA DE FIGURAS **11**

LISTA DE TABLAS **12**

1. INTRODUCCIÓN **13**

1.1. ***IPN* COMO RESPUESTA A LOS DESAFÍOS *NEW SPACE*** 13
1.2. **DESCRIPCIÓN DEL TRABAJO REALIZADO** 14
1.3. **CAMINO RECORRIDO EN EL DESARROLLO DE UNA TECNOLOGÍA** 14
1.4. **MÉTODO PARA LA INVESTIGACIÓN TEÓRICA Y APLICADA** 15
1.4.1. Modelo Waterfall para teoría *15*
1.4.2. Modelo Agile para la investigación aplicada *16*
1.5. **DIAGRAMA DE GANTT Y ESTIMACIÓN DE HORAS** 16
1.5.1. Diagrama de Gantt *17*
1.5.2. Estimación de horas *18*
1.5.3. Comparativa entre planificación y horas realizadas *19*
1.6. **HARDWARE Y SOFTWARE UTILIZADO EN EL PROYECTO** 19
1.7. **CONTENIDO** 20

2. TECNOLOGÍAS PARA LA RED INTERPLANETARIA **21**

2.1. **ESTANDARIZACIÓN Y ESPECIFICACIONES** 21
2.1.1. Actores en la IPN *21*
2.1.2. Publicaciones de interés de IRTF (DTNRG) *24*
2.1.3. Publicaciones CCSDS (SIS-DTN) *25*
2.1.4. Publicaciones IETF (DNTWG) *26*
2.2. **DELAY-DISRUPTION TOLERANT NETWORK (DTN)** 27
2.2.1. Arquitectura de Internet *27*
2.2.2. Características de la red DTN *27*
2.2.3. Arquitectura DTN *31*
2.2.4. Bundle Protocol *43*
2.2.5. Capas de Convergencia *51*
2.2.6. Protocolos de enrutamiento *53*
2.3. **IMPLEMENTACIONES** 58
2.3.1. Adaptación del estándar a la implementación *58*
2.3.2. Implementaciones para la versión 6 de Bundle Protocol *59*
2.3.3. Implementaciones para la versión 7 de Bundle Protocol *60*
2.4. **EJEMPLOS Y BENEFICIOS DE USO DE DTN** 61

3. DISEÑO DEL BANCO DE PRUEBAS Y SIMULACIÓN **65**

3.1. **REQUERIMIENTOS** 65
3.2. **DIAGRAMA DE BLOQUES DEL SISTEMA** 66
3.3. **ENTORNO DE LABORATORIO** 67
3.4. **HERRAMIENTAS PARA EL BANCO DE PRUEBAS** 69
3.5. **HERRAMIENTAS PARA LA SIMULACIÓN** 70
3.6. **HERRAMIENTAS DE ANÁLISIS Y TELEMETRÍA** 71
3.7. **TOPOLOGÍA DE RED INTERPLANETARIA PROPUESTO** 71

4. ESCENARIO DE RED IPN 1: BANCO DE PRUEBAS 75

4.1. ESCENARIO PARA EL SISTEMA TIERRA-MARTE 75
4.1.1. Nodos de la red en el sistema Tierra-Marte 75
4.1.2. Enlaces entre los nodos de la red 76
4.2. PLAN DE CONTACTOS 76
4.3. ADAPTACIÓN DEL ESCENARIO AL BANCO DE PRUEBAS 77
4.4. NODOS Y RED DTN 79
4.5. PRUEBAS DE APLICACIONES DTN 80
4.5.1. bpecho y bping 80
4.5.2. Messages (chat) 80
4.5.3. Transferencia de archivos: bprecvfile bpsendfile 81
4.5.4. DTNperf3 81

5. ESCENARIO DE RED IPN 2: EMULADOR. 82

5.1. TOPOLOGÍA Y ESCENARIO DE LA RED. ENLACE CON ESCENARIO DE RED IPN 1 82
5.2. PLAN DE CONTACTOS 83
5.3. NODOS, RED DTN Y PRUEBAS DE TRANSMISIÓN 83

6. VIABILIDAD TÉCNICA 85

6.1. PUNTOS FUERTES 85
6.2. PUNTOS DÉBILES 85

7. CONCLUSIONES 86

7.1. LECCIONES APRENDIDAS 87
7.2. TIEMPO DEDICADA A CADA PAC Y ANÁLISIS DE PLANIFICACIÓN 87
7.3. OBJETIVOS ALCANZADOS 88
7.4. AMPLIACIÓN DEL TRABAJO 88

ANEXOS I

ANEXO 1: TABLAS COMPARATIVAS BP6 Y BP7 I
ANEXO 2: CÁLCULO DE CONTACTOS IV
ANEXO 3: NETEM Y USO DE IPERF3 XIII
ANEXO 4: CONFIGURACIÓN DE UN NODO DTN XVI
ANEXO 5: GRAYLOG XXIII
ANEXO 6: INFLUXDB Y TELEGRAF XXV
ANEXO 7: DTNPERF3 XXVI
ANEXO 8: PLAN DE CONTACTOS XXXIII

BIBLIOGRAFÍA XLV

Acrónimos

#

μPCN
Micro Planetary Communication Network . 60

A

AA
Application Agent 44
ADU
Application Data Unit 33
AU
Astronomical Unit 76

B

BDT
Best Delivery Time 58
BER
Bit Error Rate 20
BP
Bundle Protocol 13
BPA
Bundle Protocol Agent 44
BPSec
Bundle Protocol Security 42
BSR
Bundle Status Report 43

C

CBOR
Concise Binary Object Representation 46
CCSDS
Consultative Committee for Space Data Systems 23
CGR
Contact Graph Routing 53
CLA
Convergence Layer Adapter 44
CORE
Common Open Research Emulator 70
COTS
Commercial Of-The-Shelf 13
CRC
Cyclic Redundancy Check 44

D

DAG
Directed Acyclic Graph 57
DDoS
Distributed Denial of Service 40
DINET
Deep Impact Network Experimient 61
DNS
Domain Name System 23
DoS
Denial of Service 39
DSN
Deep Space Network 75
DTN
Delay-Disruption Tolerant Network 13
DTNRG
DTN Research Group 22

E

EID
Endpoint Identifier 33
EMANE
Extendable Mobile Ad-hoc Network Emulator 70
ESA
European Space Agency 75

F

FIB
Forwarding Information Base 36

I

IAB
Internet Architecture Board 22
IANA
Internet Assigned Numbers Authority 23
IAP
IPN Access Provider 72
IAU
IPN Access User 72
ICANN
Internet Corporation for Assigned Names and Numbers 23
I-D
Internet Drafts 23
IETF
Internet Egineering Task Force 22
IGS
IPN Ground Station 73
IGW
IPN Gateway 73
ISS
International Space Station 61
ION
Interplanetary Overlay Network 60
IPN
Inter Planetary Network 13
IPNSIG
Interplanetary Networking Special Interest Group 23

IRTF
Internet Research Task Force 22
ISL
Inter-Satellite Link 74
ISOC
Internet Society 22
ISP
Internet Service Provider 71

J

JSON
JavaScript Object Notation 46

L

LADEE
Lunar Athmosphere and Dust Environment Explorer 61
LTP
Licklider Transmission Protocol 13

M

MLS
Mars Science Laboratory 75
MRG
Minimum Reception Group 33
MRO
Mars Reconnaissance Orbiter 75
MSB
Most Significant Bit 45

N

NAIF
Navigation and Ancillary Information Facility 77
NGF
Next Generation Firewall 40

O

OWLT
One-Way Light Time..... 54

P

PDU
Protocol Data Unit 33
PKI
Public Key Infraestructure 39
POSIX
Portable Operating System Interface UniX 61
PRoPHET
Probabilistic Routing Protocol for Intermittently Connected Networks 53

R

RFC
Request for Comments 22
RIB
Routing Information Base 36
RTT
Round Trip Time 20

S

SDNV
Self Delimiting Numeric Values 45
SMD
Science Mission Directorate 62
SSI
Solar System Internet 13
SSP
Scheme-Specific Part 33

T

TEG
Time Evolving Graph 56
TTL
Time to Live..... 54

U

URI
Uniform Resource Identifier 33

V

VM
Virtual Machine 68

W

WGC
WebGeocalc..... 77

Lista de figuras

Figura 1: Fases de desarrollo Agile. 16
Figura 2: Soluciones para la red interplanetaria 21
Figura 3 Actores en el desarrollo de la arquitectura de red interplanetaria. 22
Figura 4: Comparación de capas de protocolo IP y DTN. 28
Figura 5: Comparación de envío de paquetes en una red IP y una red DTN. 29
Figura 6: Regiones y gateways. 32
Figura 7: Bundle Protocol. 32
Figura 8: Fragmentación de un bundle. 35
Figura 9: Proceso DTN desde la aplicación hasta la capa de convergencia. 38
Figura 10: Primary block y payload block para la versión 6. 50
Figura 11: Proceso de enrutamiento. 55
Figura 12: Plan de contactos. 56
Figura 13: Representaciones de plan de contactos. 56
Figura 14: Grafo de Contacto. 57
Figura 15: Cálculo de rutas. 58
Figura 16: NASA SCaN network DTN infusion plan. 62
Figura 17: Sistema de virtualización para banco de pruebas y emulador. 68
Figura 18: Modelo IPN-IAU. 72
Figura 19: Sistema de pruebas y emulador para red Luna-Tierra-Marte. 74
Figura 20: Mapa de red de escenario Luna-Tierra-Marte para banco de pruebas y emulador. 79
Figura 21: Entorno de red obtenido del emulador CORE. 82
Figura 22: parámetros WGC para los contactos MSL-MRO. IV
Figura 23: contactos MSL-MRO. IV
Figura 24: parámetros WGC para los contactos MSL-Odyssey. V
Figura 25: contactos MSL-Odyssey. V
Figura 26: MRO-DSS25(Goldstone). VI
Figura 27: parámetros WGC para los contactos MRO-DSS-55 (Madrid). VI
Figura 28: contactos MRO-DSS-55 (Madrid). VII
Figura 29: parámetros WGC para los contactos MRO-DSS-25 (Goldstone). VII
Figura 30: contactos MRO-DSS-25 (antena de Goldstone). VII
Figura 31: parámetros WGC para los contactos MRO-DSS-35 (Canberra). VIII
Figura 32: contactos MRO-DSS-35 (Canberra). VIII
Figura 33: parámetros WGC para los contactos Odyssey-DSS-55 (Madrid). IX
Figura 34: contactos Odyssey-DSS-55 (Madrid). IX
Figura 35: parámetros WGC para los contactos Odyssey-DSS-25 (Goldstone). X
Figura 36: contactos Odyssey-DSS-25 (Goldstone). X
Figura 37: parámetros WGC para los contactos Odyssey-DSS-35 (Canberra). XI
Figura 38: contactos Odyssey-DSS-35 (Canberra). XI
Figura 39: Captura de pantalla del servidor graylog. XXIV
Figura 40: Captura de pantalla de InfluxDB Network Interface Performance. XXV

Lista de tablas

TABLA 1: PUBLICACIONES DE IRTF RELACIONADOS CON DTN. 24
TABLA 2: PUBLICACIONES CCSDS RELACIONADAS CON DTN. 25
TABLA 3: BORRADORES ACTIVOS DE DTNWG. 26
TABLA 4: TIPOS DE CONTACTOS SEGÚN RFC 4838. 30
TABLA 5: BLOQUES Y CIFRADO BUNDLE PROTOCOL 6. 41
TABLA 6: BLOQUES Y CIFRADO BUNDLE PROTOCOL 7. 42
TABLA 7: TIPOS DE REGISTROS. 43
TABLA 8: TIPOS DE DATOS CBOR Y CARACTERÍSTICAS. 47
TABLA 9: BENEFICIOS DEL USO DE DTN EN MISIONES ESPACIALES. 63
TABLA 10: CUESTIONES QUE DEBEN ABORDARSE PARA EL USO DE DTN EN MISIONES ESPACIALES. 64
TABLA 11: VLANS PARA LA RED DEL ENTORNO DE PRUEBAS. 69
TABLA 12: ROLES Y TIPOS DE ELEMENTOS DE LA RED IPN. 74
TABLA 13: VALORES DE LOS ENLACES ENTRE LOS NODOS DEL ESCENARIO PROPUESTO PARA EL SISTEMA TIERRA-MARTE. 76
TABLA 14: CONTACTOS DSN MADRID DÍAS 27 A 30 DE ABRIL. 77
TABLA 15: PLAN DE CONTACTOS REDIMENSIONADO PARA UN PERIODO DE PRUEBA MÁXIMO DE 24 MINUTOS. 78
TABLA 16: VALORES DE LOS ENLACES ENTRE LOS NODOS DEL ESCENARIO PROPUESTO PARA EL SISTEMA TIERRA-LUNA. 83
TABLA 17: TIPOS DE BLOQUES PARA VERSIONES 6 Y 7. I
TABLA 18: FLAGS DE CONTROL DE PROCESOS DE BUNDLES PARA VERSIONES 6 Y 7. I
TABLA 19: FLAGS DE CONTROL DE PROCESOS DE BLOQUES PARA VERSIONES 6 Y 7. II
TABLA 20: CÓDIGOS DE NOTIFICACIONES DE ESTADO DEL BUNDLE PARA VERSIONES 6 Y 7. III
TABLA 21: CONTACTOS MSL-MRO. V
TABLA 22: CONTACTOS MSL-ODYSSEY V
TABLA 23: CONTACTOS MRO-DSS-55 (MADRID). VII
TABLA 24: CONTACTOS MSL-DSS-25 (GOLDSTONE). VII
TABLA 25: CONTACTOS MRO-DSS-35 (CANBERRA). VIII
TABLA 26: CONTACTOS MRO-DSN PARA EL DÍA 1 DE MAYO DE 2020. IX
TABLA 27: CONTACTOS ODYSSEY-DSS-55 (MADRID). X
TABLA 28: CONTACTOS ODYSSEY-DSS-25 (GOLDSTONE). X
TABLA 29: CONTACTOS ODYSSEY-DSS-35 (CANBERRA). XI
TABLA 30: CONTACTOS ODYSSEY-DSN PARA EL DÍA 1 DE MAYO DE 2020. XII

1. Introducción

1.1. *IPN* como respuesta a los desafíos *New Space*

Contexto y justificación del trabajo

Hasta hace pocos años, las **misiones espaciales** únicamente estaban al alcance de pocos gobiernos y empresas de telecomunicaciones. La estandarización del concepto **CubeSat** [1][2]**, así como los nuevos desafíos para la exploración del Sistema Solar** han abierto la puerta a una escalada sin precedentes en el desarrollo de misiones espaciales. Este hecho, ha permitido acelerar la cooperación entre Universidades, Agencias Espaciales y empresas.

Además, ha nacido un nuevo paradigma -***New Space*** *con un volumen de negocio de 366B $ en 2019* [3] - donde diferentes equipos de investigación y empresas privadas están ideando y proyectando el uso de satélites de órbita baja como redes masivas de dispositivos para usos científicos y comerciales, así como nuevas misiones de exploración más allá de la órbita terrestre.

En los últimos años, se ha demostrado una sorprendente **capacidad de transmisión de datos** en los nanosatélites y sistemas de comunicación, así como de **procesamiento** [4]. Este hecho, -sumado a la **reducción de costes por el uso de COTS**[1]- ha iniciado una nueva carrera espacial que nos hace imaginar un **entorno donde desarrollar servicios y aplicaciones**[2]; incluso replantear el estado actual de la industria.

El incremento de la industria espacial ha puesto sobre la mesa la **necesidad de una red global**, que al igual que la Internet terrestre, ofrezca un mecanismos flexible, confiable y escalable para la **red de datos en el Sistema Solar o *Solar System Internet* (SSI)** [5][6]**.** Las propuestas teóricas se impulsan bajo las tecnologías de la **Red Interplanetaria o *Inter-Planetary Networking* (IPN)** [7][8], para materializarse en propuestas de estandarización e implementaciones de agencias espaciales y empresas privadas.

Entre las nuevas soluciones, se está trabajando en sistemas para que distintos elementos - satélites, sondas, estaciones, rovers, etc.- puedan realizar **transferencias de información mucho más eficientes**. **Las redes Delay-Disruption Tolerant Network (DTN)** [9] **son la propuesta actual para esta función**. Las soluciones DTN, que veremos en los próximos capítulos, se materializan principalmente en las especificaciones Bundle Protocol (BP) y Licklider Transmission Protocol (LTP) de las cuales existen diferentes implementaciones.

---

[1] Producto de caja o *Commercial Of-The-Shelf* (COTS) es un término, que en tecnologías de la información, hace referencia a un producto fabricado en serie para la distribución comercial, en contrapartida a un producto diseñado y construido a medida para un uso exclusivo. La utilización de productos diseñados para su fabricación en serie, reduce drásticamente el coste de los proyectos.

[2] https://oscw.space.

El proyecto, se centra en el análisis y despliegue de las implementaciones para redes **DTN en un contexto de red interplanetaria**.

1.2. Descripción del trabajo realizado

El trabajo realizado en un primer bloque, consiste en el estudio y descripción de una arquitectura de red, protocolos y especificaciones desde su propuesta hasta su borrador de estándar.

También se han identificado diferentes implementaciones y sus características para seleccionar una de las implementaciones para los capítulos prácticos.

En la segunda parte, con un contenido práctico, se ha implementado un escenario de pruebas y se han estudiado los diferentes escenarios y su caracterización para después adaptarlo a simulaciones, realizando además diferentes modificaciones en las características del escenario y comprobando funcionalidad y comportamiento. El diseño de un entorno de pruebas y el uso de diferentes herramientas, forma parte del proyecto y el trabajo realizado.

1.3. Camino recorrido en el desarrollo de una tecnología

Objetivos

El objetivo principal consiste en estudiar el desarrollo de una tecnología en sus primeros pasos. Es un momento en el cual todavía no se ha consolidado, pero sin embargo es el más emocionante e interesante. Y es atractivo porque podemos ver las nuevas revisiones de las especificaciones y actualizaciones de software, además de un goteo constante de publicaciones o *papers*. Incluso, tenemos la opción de participar de una u otra forma.

Desde un punto de vista transversal, los objetivos de fondo del proyecto son los siguientes:

a) Estudio de las **características de una arquitectura de red**.
b) **Análisis del código de las implementaciones** de un protocolo de red.
c) Uso de entorno para la **simulación de un sistema de red** de datos.
d) **Análisis de herramientas de administración y control de un protocolo de red**.
e) Estudio cronológico del desarrollo de un **proyecto tecnológico** emergente (RFC4838, RFC5050, RFC5326, borradores de propuesta de estándar).

f) **Análisis de una solución de red** de datos en el Sistema Solar, donde se extenderán las capacidades de la red de Internet terrestre.

1.4. Método para la investigación teórica y aplicada

Enfoque y método

El proyecto es una investigación teórica y aplicada en igual medida. Previo a los puntos del proyecto que están plasmados en la memoria, existe un trabajo de búsqueda de información en la temática elegida, con el objetivo de crear un marco de diseño y planificación, que está reflejado en la presentación o propuesta de TFG. El siguiente paso, La elaboración del proyecto, sigue una estructura y un modelo de desarrollo coherente con el plan de trabajo.

Una vez definido el objeto y los objetivos del proyecto, el trabajo queda dividido en dos partes: una teórica y otra aplicada o práctica. La parte teórica se refiere a un estudio del estado del arte y síntesis de información proveniente de las fuentes consultadas, para tomar decisiones y realizar una selección adecuada para concretar qué forma tendrá la parte práctica. La fase empírica se refiere a modelar un escenario de prueba y análisis.

1.4.1. Modelo Waterfall para teoría

El modelo Waterfall o el modelo tradicional es el usado en la parte teórica. El motivo del uso de este modelo tiene que ver con definir unos pasos y seguirlos secuencialmente para llegar a una conclusión final. En este caso, no es necesario replantear o profundizar más en ninguna sección ya estudiada. Una vez en la conclusión final, la parte teórica queda cerrada y sigue la parte práctica.

Las redes interplanetarias (IPN) son un ámbito de investigación y en vías de desarrollo y consolidación. En esta situación, el espacio temporal en el que se mueve esta tecnología es relativamente corto. Entonces, el enfoque consiste en recorrer un camino desde los orígenes más teóricos, hasta encontrar las herramientas actuales con las que cuentan los ingenieros e investigadores. Dicho recorrido corresponde a la siguiente secuencia de preguntas planteadas:

a. ¿Qué actores participan en la industria y desarrollan esta tecnología? (quién)
b. ¿Existe una estandarización internacional oficial o de facto? (cómo)
c. ¿Por qué DTN? (qué y por qué)
d. ¿Se ha materializado esta tecnología? ¿hay una aplicación práctica “en producción”? (dónde)

e. ¿Podemos trabajar con esta tecnología hoy?: Conclusiones. La IPN en acción pasando a la siguiente fase en un banco de pruebas.

1.4.2. Modelo Agile para la investigación aplicada

La parte práctica está basada en una metodología de iteraciones (figura 1). Por ese motivo, tiene más sentido utilizar el modelo Agile con un enfoque incrementar y adaptativo. Los pasos en la metodología serán los siguientes:

1. Requerimientos: definir el escenario de red IPN.
2. Plan: pasos a realizar para modelar un escenario propuesto.
3. Diseño: diseñar el modelo según las aplicaciones disponibles.
4. Desarrollo: instalar y configurar software y aplicaciones.
5. Publicación: probar aplicaciones y transmisión de datos.
6. Telemetría y monitorización: administrar la red, análisis y telemetría.

Figura 1: Fases de desarrollo Agile.

Hay varias interacciones donde cada una de ellas se podrían desplegar en diferentes *sprints* o tareas objetivo. Cada una de ellas corresponde a diferentes fases de la parte práctica:

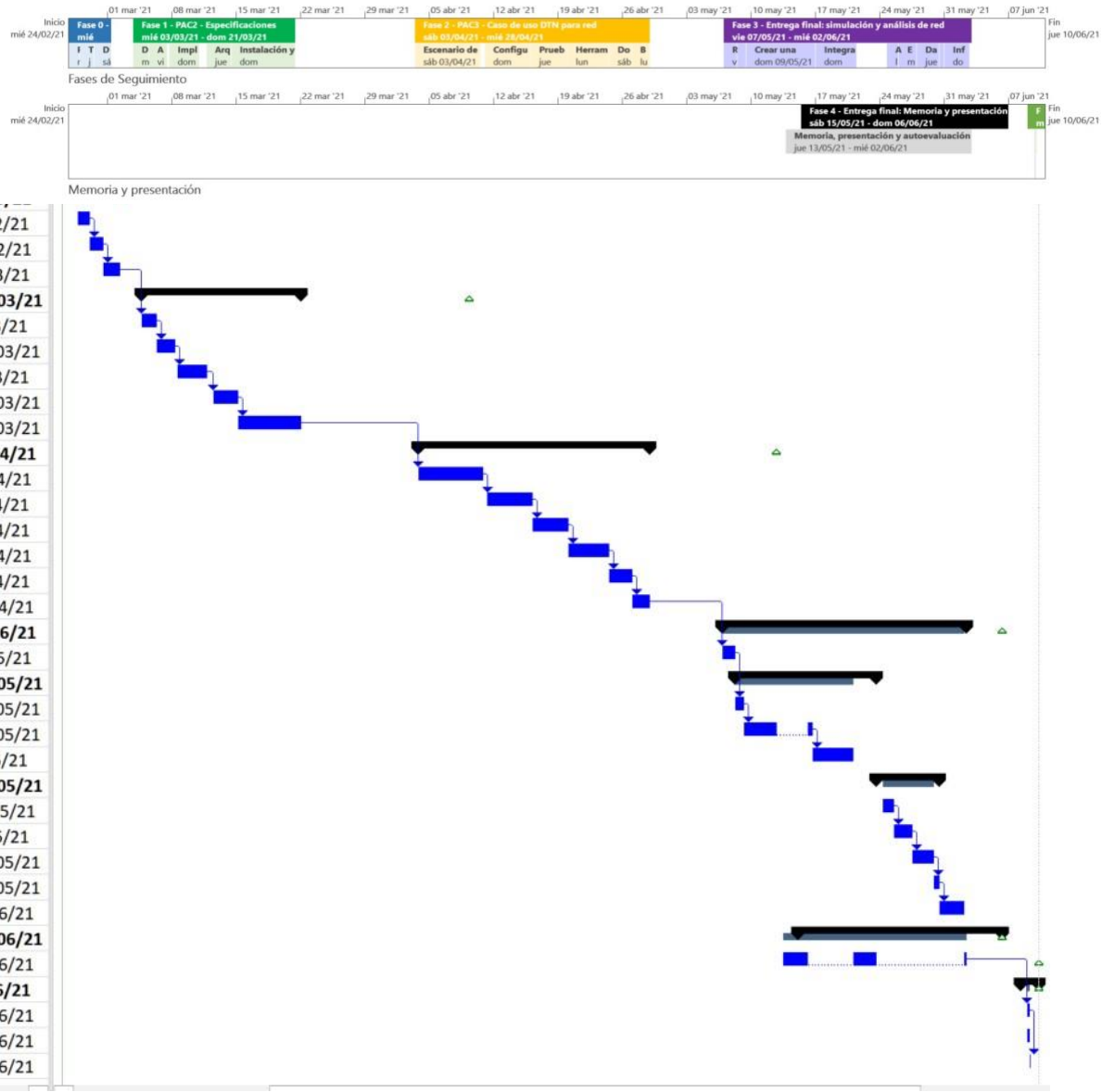
I. Entorno básico para caso de uso de una red DTN utilizando máquinas virtuales.
II. Ampliación del caso de uso con una herramienta de simulación.
III. Profundizar en herramientas de administración y gestión, así como en análisis y telemetría.

1.5. Diagrama de Gantt y estimación de horas

Planificación del trabajo

1.5.1. Diagrama de Gantt

	EDT	Nombre de tarea	Comienzo	Fin
0	**P-**	**UOC - Carlos Lozano - TFG**	**mié 24/02/21**	**jue 10/0**
1	**P-A**	**Fase 0 - PAC 1 - Plan de trabajo**	**mié 24/02/21**	**lun 01/0**
2	P-A.1	Redacción de plan de trabajo	mié 24/02/21	jue 25/02/21
3	P-A.2	Tareas y diagrama de Gantt	jue 25/02/21	sáb 27/02/21
4	P-A.3	Diagrama de red y sistema	sáb 27/02/21	lun 01/03/21
5	**P-B**	**Fase 1 - PAC2 - Especificaciones DTN e implementaciones**	**mié 03/03/21**	**dom 21/03/21**
6	P-B.1	DTN: Conceptos y objetivos	mié 03/03/21	vie 05/03/21
7	P-B.2	Análisis y cronología de las especificaciones (RFCs)	vie 05/03/21	dom 07/03/21
8	P-B.3	Implementaciones DTN existentes: Características principales	dom 07/03/21	jue 11/03/21
9	P-B.4	Arquitectura y funcionalidades de las principales implementacio	jue 11/03/21	dom 14/03/21
10	P-B.5	Instalación y configuración de una implementación DTN	dom 14/03/21	dom 21/03/21
11	**P-C**	**Fase 2 - PAC3 - Caso de uso DTN para red interplanetaria**	**sáb 03/04/21**	**mié 28/04/21**
12	P-C.1	Escenario de red DTN: Instalación de VMs y software	sáb 03/04/21	sáb 10/04/21
13	P-C.2	Configuración protocolos para el escenario propuesto	dom 11/04/21	jue 15/04/21
14	P-C.3	Pruebas de transmisión con aplicaciones DTN	jue 15/04/21	lun 19/04/21
15	P-C.4	Herramientas de administración para la red DTN	lun 19/04/21	sáb 24/04/21
16	P-C.5	Documentación e informe de seguimiento	sáb 24/04/21	lun 26/04/21
17	P-C.6	Borrador de la memoria	lun 26/04/21	mié 28/04/21
18	**P-D**	**Fase 3 - Entrega final: simulación y análisis de red**	**vie 07/05/21**	**mié 02/06/21**
19	P-D.1	Revisión funcionalidades y resultados	vie 07/05/21	sáb 08/05/21
20	**P-D.2**	**Simulación**	**sáb 08/05/21**	**dom 23/05/21**
21	P-D.2.	Instalación de VM y software de simulación	sáb 08/05/21	dom 09/05/21
22	P-D.2.	Crear una simulación	dom 09/05/21	dom 16/05/21
23	P-D.2.	Integrar escenario de la Fase 2 en la simulación	dom 16/05/21	vie 21/05/21
24	**P-D.3**	**Análisis y telemetría**	**lun 24/05/21**	**dom 30/05/21**
25	P-D.3.	Análisis de tráfico de red de entorno y simulación	lun 24/05/21	mar 25/05/21
26	P-D.3.	Envío de información del sistema de red a BBDD	mar 25/05/21	jue 27/05/21
27	P-D.3.	Dashboard con datos de la BBDD	jue 27/05/21	dom 30/05/21
28	P-D.4	Preparación de entrega y presentación: Conclusiones finales	dom 30/05/21	dom 30/05/21
29	P-D.5	Informe de entrega.	dom 30/05/21	mié 02/06/21
30	**P-E**	**Fase 4 - Entrega final: Memoria y presentación**	**sáb 15/05/21**	**dom 06/06/21**
31	P-E.1	Memoria, presentación y autoevaluación	jue 13/05/21	mié 02/06/21
32	**P-F**	**Fase 5 - Presentación y defensa**	**mié 09/06/21**	**jue 10/06/21**
33	P-F.1	Presentación	mié 09/06/21	mié 09/06/21
34	P-F.2	Demostración	mié 09/06/21	mié 09/06/21
35	P-F.3	Defensa	mié 09/06/21	mié 09/06/21



1.5.2. Estimación de horas

La presente estimación tiene relación con el punto anterior -planificación general-, sobre el que se calculan las horas aproximadas invertidas en todo el proceso. No se han calculado las horas invertidas en todo el proceso de investigación previa a la presente propuesta. **Las horas aproximadas necesarias son 20 horas semanales con un total de 320 horas.**

P-A Fase 0 (Plan de trabajo)	
P-A.1 Redacción del plan de trabajo (varios puntos)	5h
P-A.2 Tareas y diagrama de Gantt	5h
P-A.3 Diagrama de red y sistema	10h
Total horas P-A	**20h**
P- B Fase 1 (PAC 2) – Especificaciones DTN e implementaciones	
P-B.1 DTN: Conceptos y objetivos	10h
P-B.2 Análisis y cronología de las especificaciones (RFCs)	10h
P-B.3 Implementaciones DTN existentes. Características principales	15h
P-B.4 Arquitectura y funcionalidades de las principales implementaciones	15h
P-B.5 Instalación y configuración de una implementación DTN	30h
Total horas P-B	**80h**
P- C Fase 2 (PAC 3) – Caso de uso DTN para red interplanetaria	
P-C.1 Escenario de red DTN: Instalación de VMs y software	15h
P-C.2 Configuración de protocolos para el escenario propuesto	20h
P-C.3 Pruebas de transmisión con aplicaciones DTN	20h
P-C.4 Herramientas de administración para la red DTN	20h
P-C.5 Documentación e informe de seguimiento	10h
P-C.6 Borrador de la memoria	10h
Total horas P-C	**95h**
P-D Fase 3 (Entrega Final 1 de 2) – simulación y análisis de red	
P-D.1 Revisión funcionalidades y resultados	
5h	
P-D.2 Simulación	
P-D.2.1 Instalación de VM y software de simulación	5h
P-D.2.2 Crear una simulación	20h
P-D.2.3 Integrar escenario de la Fase 2 en la simulación	20h
Total horas P-D.2	45h
P-D.3 Análisis y telemetría	
P-D.3.1 Análisis de tráfico de red de entorno y simulación	5h
P-D.3.2 Envío de información del sistema de red a BBDD	10h
P-D.3.2 Dashboard con datos de la BBDD	10h
Total horas P-D.3	25h
P-D.4 Preparación de entorno para entrega y presentación: conclusiones	5h
P-D.5 Informe de entrega	10h
Total horas P-D	**90h**
P-E Fase 4 (Entrega Final 2 de 2) – Memoria y presentación	
P-E.1 Memoria, presentación y autoevaluación	30h
Total horas P-E	**30h**
P-F Fase 5 (Presentación, demostración y defensa)	
P-F.1 Presentación	-
P-F.2 Demostración	-

P-F.3 Defensa	-
Total horas P-F	**5h**
TOTAL A+B+C+D+E+F	**320 h**

1.5.3. Comparativa entre planificación y horas realizadas

Entre la planificación y las horas realizadas se han producido diferencias que han incidido más en la profundidad de desarrollo de las diferentes líneas del proyecto. En este sentido, se ha dado más importancia al avance progresivo y al cierre de cada tarea con los recursos y resultados más viables, que a la obtención de resultados concretos que podían ser inviables. Si que se ha producido un aumento de las horas dedicadas en general en cada uno de los apartados, considerando que el proyecto puede haber requerido un 10% más de las horas proyectadas.

En el apartado 7.3 se detalla el tiempo dedicado y el análisis de la planificación.

1.6. Hardware y software utilizado en el proyecto

Recursos empleados

Los recursos empleados para el desarrollo del proyecto se dividen en dos grupos. El primero tiene que ver con los medios para la gestión de proyecto, gestión de la información y redacción de memoria y presentación. En segundo lugar, los recursos técnicos necesarios para la ejecución práctica del proyecto, que se detallan en el capítulo 3.

El primer grupo de recursos empleados han sido:

Portátil MacBook Pro
Microsoft Office para Mac
Mendeley Desktop
Uso: documentación.

Portátil Sony Vaio
Microsoft Windows 10 Pro
Microsoft Visio Profesional
Oracle VM VirtualBox (máquinas virtuales Ubuntu Server y Ubuntu Desktop en las versiones 18.04 y 20.04)
Uso: elaboración de esquemas y planos. Máquinas virtuales en local previo instalación en infraestructura de virtualización remota en ELETE.

1.7. Contenido

Descripción de los capítulos de la memoria

El **capítulo 2** está dedicado al concepto, origen y objetivos de las redes DTN orientadas a la red interplanetaria. Contiene una descripción de las **especificaciones y las implementaciones**. Es en este punto donde se presenta una comparativa del software y sus versiones existentes hasta la fecha, para valorar cual o cuales utilizar para diseñar un escenario de pruebas.

El **capítulo 3** trata la arquitectura de un **escenario de red IPN**. Se incluye el diagrama de bloques final del sistema y las **opciones para simulación y análisis** utilizadas en los siguientes capítulos. También, describe las características técnicas de sistema utilizado para el **entorno de pruebas**.

El **capítulo 4** incluye en primer lugar la metodología utilizada para **caracterizar una red DTN**, describiendo los diferentes puntos que se deben tener en cuenta, como órbitas, propagación de la señal, calidad del medio, etc. En segundo lugar, analiza la viabilidad de trasladar el escenario al **entorno de pruebas**. En tercer lugar, comprueba la **funcionalidad con pruebas de transmisión,** utilizando aplicaciones disponibles. Hay que tener en cuenta que en este capítulo se trabaja con simulaciones que ayudan a caracterizar con una precisión limitada una red de estas características, por lo que habrá situaciones para un escenario IPN que puede no sean reproducibles con exactitud. Estas características pueden ser: la comunicación asimétrica, el retardo -recordemos que estamos trabajando en transmisiones interplanetarias, que en un sistema Tierra-Marte pueden ser de 14 minutos en tiempo de ida y vuelta o *Round Trip Time* (RTT)-, el ajuste de tasa de transmisión y *Bit Error Rate* (BER).

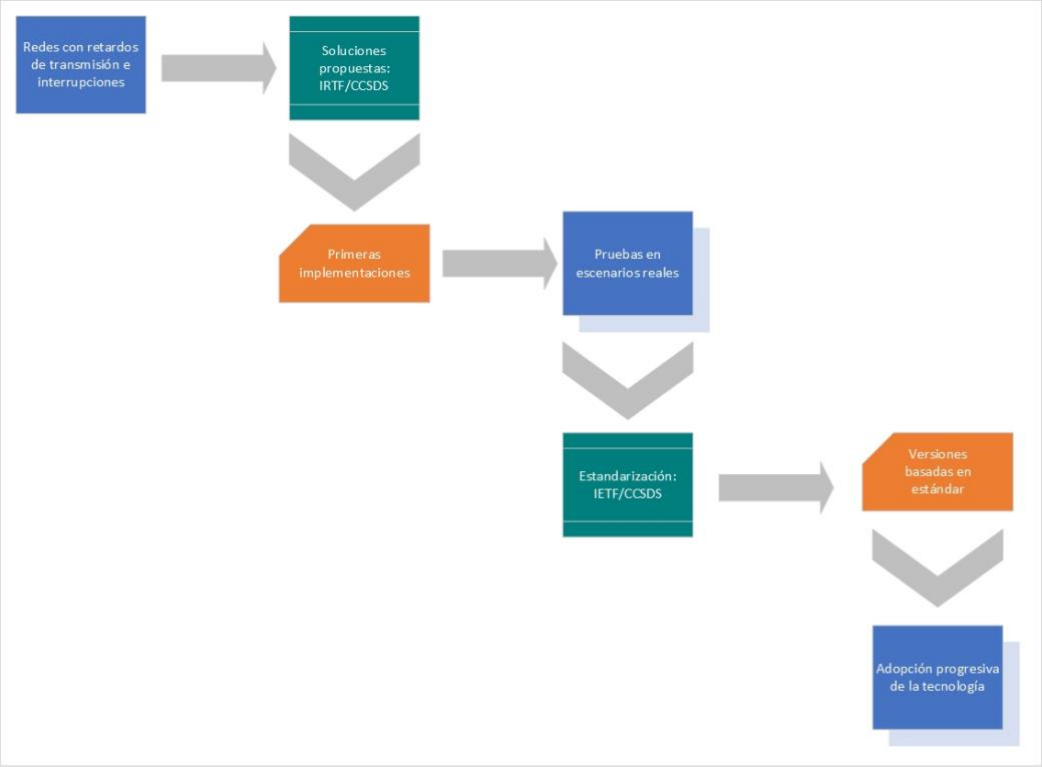
El **capítulo 5** está dedicado a un **emulador** donde completar escenario de red propuesto. Este sistema queda conectado a la red DTN trabajada en el capítulo 4.

Los **capítulos 6 y 7** están dedicados a la **viabilidad** de la tecnología discutida en el proyecto y a las **conclusiones** respectivamente.

2. Tecnologías para la red interplanetaria

La red interplanetaria es un concepto que genera necesidades operativas. Estas necesidades se convierten en requerimientos que no son solucionados de forma óptima con las tecnologías actuales para Internet. En este capítulo se analiza el proceso de generación de soluciones (figura 2).

Figura 2: Soluciones para la red interplanetaria

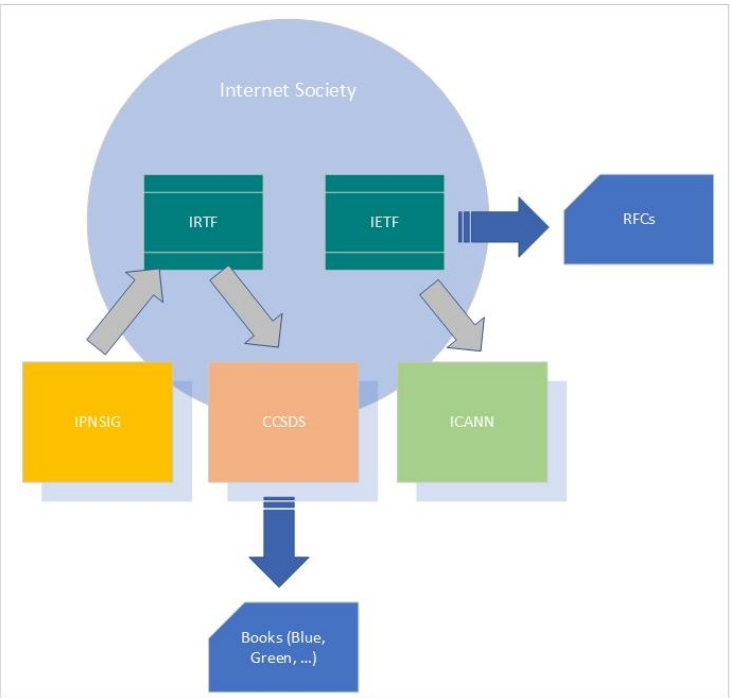


2.1. Estandarización y especificaciones

2.1.1. Actores en la IPN

DTN surge para dar respuesta al concepto IPN en 1998 [10]. La tecnología comienza a desarrollarse a mediados de 2000 y se establece en 2010 como estándar para la adopción en SSI. Diferentes instituciones han hecho posible la estandarización. En la figura 3 se puede ver la relación entre las diferentes organizaciones para el desarrollo de las tecnologías para IPN y SSI.

Figura 3 Actores en el desarrollo de la arquitectura de red interplanetaria.



Internet Society (ISOC): es una organización global sin ánimo fundada en 1992 por Vint Cerf y Bob Kahn cuya misión es promover y potenciar el desarrollo de Internet como un recurso para la sociedad. ISOC provee de infraestructura corporativa a IETF y IRTF. Desde ISOC se ha constituido IAB.

Internet Architecture Board (IAB): es una organización designada por ISOC, encargada del desarrollo de Internet desde una perspectiva técnica. IAB confirma los directores de IETF y de IRTF y el presidente de IRTF. Además, se encarga de las revisiones de los procesos de estandarización y de la publicación de los RFCs.

Internet Engineering Task Force (IETF): es una organización internacional y abierta encargada de la evolución de la arquitectura de Internet y que está organizada en grupos de trabajo. Se enfoca en el corto plazo y la generación de estándares. Los trabajos se presentan a la comunidad de IETF en forma de publicación llamada Request for Comments (RFC). Estas publicaciones son revisadas por la comunidad.

Internet Research Task Force (IRTF): es una organización internacional y abierta encargada de la investigación para el desarrollo de tecnologías de Internet. Está enfocada al largo plazo y sus publicaciones tienen el formato RFC. Uno de los grupos de investigación fue DTN Research Group (DTNRG) cuyo objetivo fue la arquitectura de red DTN y sus protocolos. DTNRG concluyó su trabajo en 2016.

En IETF y en IRTF existen diferentes tipos de publicaciones. Las propuestas técnicas se conocen como *Internet Drafts* (I-D) y no tienen un estatus formal. Cualquiera puede presentar un I-D, pero no tendrá ninguna relevancia en IETF o IRTF a menos que sea adoptado por un grupo de trabajo o se apruebe como RFC. Un RFC no tiene por qué ser un estándar. Existen varios tipos de RFC:

a. Standard Track
    a. *Best Current Practices* (BCP)
    b. *Proposed Standard* (PS)
    c. *Internet Standard* (STD)

b. Informational

c. Experimental

d. Historical

Interplanetary Networking Special Interest Group (IPNSIG): Es un grupo de trabajo de los llamados *Special Interest Groups* de ISOC. Su misión es realizar un sistema funcional y escalable de comunicaciones de datos interplanetario. Promocionan la investigación y desarrollo de tecnologías así como su uso comercial.

Consultative Committee for Space Data Systems (CCSDS): organización formada por agencias espaciales formada en 1982. CCSDS es un foro de discusión para el desarrollo y operación de sistemas de datos espaciales con el fin de publicar estándares de uso en las misiones espaciales. Una de sus áreas es Space Internetworking Services (SIS) donde CCSDS ofrece estándares para servicios y protocolos para la interacción de redes. Las publicaciones se clasifican por colores:

e. *Blue*: estándares recomendados.
f. *Magenta*: prácticas recomendadas.
g. *Green*: artículos informativos
h. *Orange*: experimental
i. *Yellow*: registros
j. *Silver*: histórico

Internet Corporation for Assigned Names and Numbers (ICANN): organización internacional sin animo de lucro que administra la funciones de *Internet Assigned Numbers Authority* (IANA), que se encarga de las asignaciones de direcciones del protocolo IP, los números de protocolo y la gestión de dominios para el Sistema de Nombres de Dominio o *Domain Name System* (DNS), y administra el sistema de servidores raíz [11].

2.1.2. Publicaciones de interés de IRTF (DTNRG)

Los documentos publicados por el grupo de trabajo DTNRG están recogidos en la tabla 1. En el ámbito de IRTF cabe destacar dos RFCs que marcan el comienzo del desarrollo de las tecnologías IPN. La primera publicación es la especificación para la arquitectura DTN con el primer I-D de marzo de 2003 y publicado como RFC4838 (*Informational*) en abril de 2007 [12].

La segunda publicación a destacar es para el protocolo BP con el nombre *Bundle Protocol Specification*. El I-D es de marzo de 2003 y se publica como RFC5050 (*Experimental*) en noviembre de 2007 [13].

Tabla 1: Publicaciones de IRTF relacionados con DTN.

Publicación	Título	Fecha	Tipo
RFC 4838	Delay-Tolerant Networking Architecture	2007-04	Information RFC
RFC 5050	Bundle Protocol Specification	2007-11	Experimental RFC
RFC 5325	Licklider Transmission Protocol – Motivation	2008-09	Informational RFC
RFC 5326	Licklider Transmission Protocol – Specification	2008-09	Experimiental RFC
RFC 5327	Licklider Transmission Protocol – Security Extensions	2008-09	Experimental RFC
RFC 6255	Delay-Tolerant Networking Bundle Protocol IANA Registries	2011-05	Informational RFC
RFC 6256	Using Self-Delimiting Numeric Values in Protocols	2011-05	Informational RFC
RFC 6257	Bundle Security Protocol Specification	2011-05	Experimental RFC
RFC 6258	Delay-Tolerant Networking Metadata Extension Block	2011-05	Experimental RFC
RFC 6259	Delay-Tolerant Networking Previus-Hop Insertion Block	2011-05	Experimiental RFC
RFC 6260	Compressed Bundle Header Enconding (CBHE)	2011-05	Experimental RFC
RFC 6693	Probabilistic Routing Protocol for intermittently Connected Networks	2012-08	Experimental RFC
RFC 7116	Licklider Transmission Protocol (LTP), Compressed Bundle Header Encoding (CBHE), and Bundle Protocol IANA Registries	2014-02	Informational RFC
RFC 7122	Datagram Convergence Layers for the Delay- and Disruption- Tolerant Networking (DTN)	2014-03	Experimental RFC

	Bundle Protocol and Licklider Transmission Protocol (LTP)			
RFC 7242	Delay-Tolerant Networking Convergence-Layer Protocol	TCP	2014-06	Experimental RFC

(Fuente: IETF Data-Tracker https://datatracker.ietf.org).

2.1.3. Publicaciones CCSDS (SIS-DTN)

El área *Space Internetworking Services* (SIS) se encarga de los servicios de comunicación y protocolos. En SIS se encuentra el grupo de trabajo *Delay Tolerant Networking Working Group* (SIS-DTN) dedicado a especificar los protocolos necesarios par implementar el concepto SSI.

Los diferentes proyectos en los que ha trabajado SIS-DTN han generado la serie de publicación 734.n. En la tabla 2 se muestran las publicaciones de esta serie, junto con otras adicionales que son de interés, entre las que se encuentran [14][15].

Tabla 2: Publicaciones CCSDS relacionadas con DTN.

Publicación	Título	Fecha	Tipo
CCSDS 702.1-B-1	IP over CCSDS Space Links	2012-09	Blue Book
CCSDS 702.1-B-1 Cor. 1	Technical Corrigendum 1 to CCSDS 702.1-B-1	2014-04	Blue Book
CCSDS 727.0-B-5	CCSDS File Delivery Protocol (CFDP)	2020-07	Blue Book
CCSDS 730.1-G-1	Solar System Internetworking (SSI)	2014-07	Green Book
CCSDS 730.2-G-1	Concepts and Rationale for Streaming Services over Bundle Protocol	2018-09	Green Book
CCSDS 734.0-G-1	Rationale, Scenarios, and Requirements for DTN in Space	2010-08	Green Book
CCSDS 734.1-B-1	Licklider Transmission Protocol (LTP) for CCSDS	2015-05	Blue Book
CCSDS 734.2-B-1	CCSDS Bundle Protocol Specification	2015-09	Blue Book
CCSDS 734.3-B-1	Schedule-Aware Bundle Routing	2019-07	Blue Book

(Fuente: SIS CCSDS https://public.ccsds.org/Publications/SIS.aspx).

2.1.4. Publicaciones IETF (DNTWG)

DTN Work Group (DTNWG) de IETF, surge en 2014 ante la necesidad de actualizar los planteamientos experimentales definidos en las publicaciones escritas por DTNRG de IRTF. Este grupo de trabajo, tiene el objetivo de publicar los estándares que ofrecerán un marco de trabajo para el ámbito comercial. Esta iniciativa está motivada por la necesidad de que empresas del mercado tengan unas características globales sobre las cuales implementar tecnologías en sus productos, que después se distribuirán para su uso. Estas características en las tecnologías cuando se escriben como estándar, facilitan la implementación por parte de diferentes fabricantes, los cuales tienen la seguridad de diseñar un producto compatible y competitivo. DTNWG en IETF persigue la estandarización las tecnologías DTN también con la misma idea de expansión y adopción en el mercado.

Las publicaciones de DTNWG todavía están en proceso por lo que durante la redacción de esta memoria son propuestas de estándar. El grupo de trabajo describe la versión 7 del protocolo *Bundle Protocol*, una revisión de las especificaciones de seguridad del RFC 6257, ofreciendo como resultado una descripción de BPSec, y repasan las capas de convergencia para UDP y TCP. En la tabla 3 se muestran las publicaciones hasta la fecha:

Tabla 3: Borradores activos de DTNWG.

Publicación	Título	Fecha	Tipo
Internet-Draft (I-D)	Bundle Protocol Version 7	25/01/2021 (Revisión 31)	Proposed Standard
Internet-Draft (I-D)	Bundle Protocol Security Specification	23/02/2021 (Revisión 27)	Proposed Standard
Internet-Draft (I-D)	Asyncrhonous Management Architechture	11/03/2021 (Revisión 1)	-
Internet-Draft (I-D)	DTN TCP Convergence Layer Protocol Version 4	15/02/2021 (Revisión 26)	Proposed Standard
Internet-Draft (I-D)	DTN UDP Convergence Layer Protocol	26/03/2021 (Revisión 1)	Proposed Standard

(Fuente: IETF Data-Tracker https://datatracker.ietf.org/wg/dtn/about/).

2.2. Delay-Disruption Tolerant Network (DTN)

2.2.1. Arquitectura de Internet

Internet está construida fundamentalmente sobre la familia de protocolos TCP/IP diseñados para interconexión redes de conmutación de paquetes o *packet-switched* networks [16]. En las redes TCP/IP el origen y destino de la transmisión están identificados y la información se divide en paquetes de datos. Estos paquetes son transmitidos a través de la red y pueden recorrer diferentes rutas, pasando por otros hosts -routers- que reenvían estos paquetes entre el origen y el destino. Internet está construida para que los datos se puedan transmitir desde un origen a un destino, siempre que ambos estén operativos y el enlace cumpla unos requisitos para el correcto funcionamiento de los protocolos y aplicaciones.

2.2.2. Características de la red DTN

Este apartado recoge las características de las redes DTN en comparación de las redes IP [17].

El correcto funcionamiento de las redes IP requiere de un sistema que cumpla los siguientes puntos:

a. La comunicación entre origen y destino debe ser continua. Las interrupciones prolongadas impiden el correcto funcionamiento de una red TCP/IP.
b. El tiempo de transmisión de ida y vuelta o *round-trip time* (RTT) entre emisor y receptor en Internet puede ser de milisegundos o segundos. Cuando ese tiempo es de minutos, la consistencia de la comunicación no es posible.
c. Las tasas de transferencia de datos o *data rates* deben ser similares en la transmisión y la recepción de datos. Cuando la asimetría entre ambas tasas es demasiado amplia, genera problemas en protocolos como TCP.
d. Las tasas de error o *error rates* deben ser bajas para el correcto funcionamiento de la comunicación.

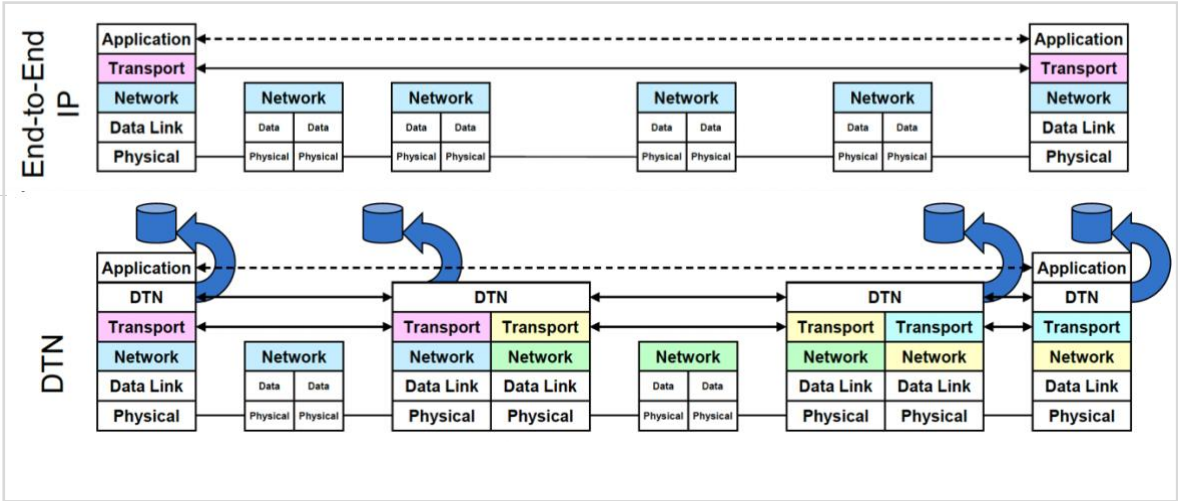
Existen diferentes entornos en los cuales se producen incidencias en la comunicación, que impiden asegurar el uso de las tecnologías utilizadas en Internet:

a. Transmisiones de datos en grandes distancias como en las comunicaciones para el espacio profundo con retardos por propagación o *propagation delays* de minutos.
b. Interrupciones predecibles en la comunicación con satélites al quedar fuera de la línea de visión debido a la dinámica planetaria y las órbitas del satélite.
c. Desconexiones en redes militares debido a condiciones de entorno y movilidad.

d. Recursos muy limitados en redes de sensores que provocan interrupciones en la comunicación.

El uso de las redes DTN son una respuesta para estas incidencias. La arquitectura de Internet tiene respuesta limitada en algunas situaciones como las comentadas. La transmisión mediante almacenamiento y reenvío o *store-and-forward* en las redes de conmutación de paquetes, permiten a los routers tomar decisiones de enrutamiento y priorizar los datos que se deben transmitir una vez recibido un paquete completo. Este almacenamiento está orientado a almacenar los bits que se reciben de un paquete hasta que se ha recibido el paquete completo, antes de reenviarlo. Es por ese motivo que, ante incidencias de conexión, este almacenamiento no soluciona el problema. Las redes IP en una transmisión para satélites LEO puede ser viable, aunque costosa en reenvíos por los tiempos de desconexión. En la comunicación en el espacio profundo es inviable; TCP tiene un saludo que necesita 1,5 RTT y 2 minutos de *timeout*.

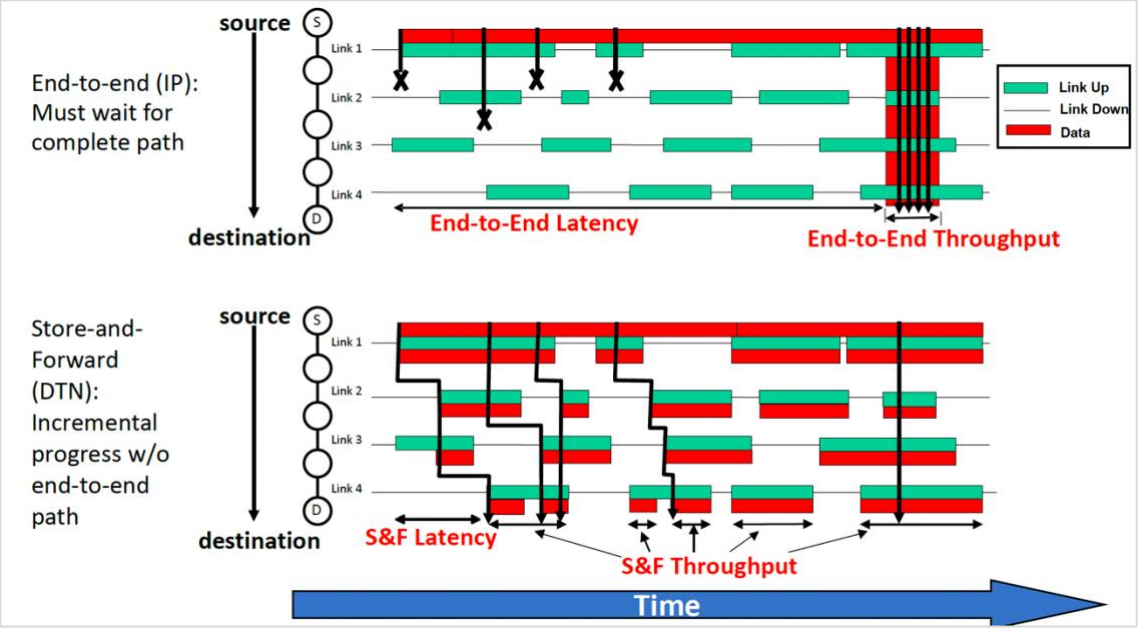
Figura 4: Comparación de capas de protocolo IP y DTN.



(Fuente: Curso ION 2019 de la NASA).

En respuesta a esta debilidad, la arquitectura DTN añade una capa de protocolo sobre los ya existentes e introduce el modelo de **almacenamiento-transporte-y-reenvío o *store-carry-and-forward***. La finalidad de esta solución es dotar a los nodos DTN de memoria persistente suficiente para almacenar los paquetes de datos recibidos y retenerlos, y así proporcionar a la red resistencia a desconexiones, retardos y fallos en la transmisión (figura 4). Es aquí donde reside unos de los factores clave de las redes DTN y por eso se ha convertido en la arquitectura propuesta para la red IPN. A partir de este almacenamiento persistente, los paquetes pueden quedar a la espera de disponer del enlace para el siguiente salto en su ruta (figura 5).

Figura 5: Comparación de envío de paquetes en una red IP y una red DTN.



En la red IP las interrupciones anulan los envíos ya que necesita una conexión origen y destino. La red DTN no está orientada a conexión origen-destino y se aplica almacenamiento persistente en los nodos de la ruta entre un origen, lo que permite aprovechar los enlaces disponibles entre cada salto de la ruta. (Fuente: Curso ION 2019 de la NASA).

Por otra parte, existen situaciones en las cuales los elementos de una red tienen conexiones intermitentes y de diferentes capacidades para la transmisión de datos. Esta problemática ya ocurre en otros casos para una red TCP/IP, y ha dado lugar a diferentes protocolos de enrutamiento que han sido punto de partida para encontrar un protocolo adecuado para las redes DTN [18]. Es por ese motivo, otro elemento clave en las redes DTN es el concepto de **contactos o *contacts***. Estos contactos pueden ser de distintos tipos como se puede ver en la tabla 4. En la práctica, los contactos DTN se tratan en tres grupos[3] para adecuar los protocols de enrutamiento, aunque cada situación real varía en sus factores específicos:

a. Un **contacto oportunista**: se produce cuando un host DTN no tiene información sobre cuándo tendrá contacto con otro host DTN, ni cuales serán las características precisas de la comunicación con dicho host.
b. Un **contacto probabilístico**: se produce cuando un host DTN tiene información sobre contactos anteriores con otros hosts DTN, y tiene la capacidad de calcular la probabilidad de contactos futuros.
c. Un **contacto determinista**: se produce cuando un host DTN tiene la información de cuales van a ser sus contactos con los hosts DTN vecinos y cuales van a ser las características del enlace con dichos hosts.

[3] La caracterización de los contactos en una red DTN determina el tipo de protocolo de enrutamiento utilizado.

Tabla 4: Tipos de contactos según RFC 4838.

Tipo de contacto	Subtipo	Descripción
Persistent	-	Conexión siempre disponible (ej. Conexión a Internet mediante DSL o Cable Modem)
On-Demand	-	Requieren una acción para instanciar una conexión (ej. conexión *dial-up*).
Intermittent	*Scheduled*	Acuerdo de contacto en hora y tiempo. (ej. lista de contactos planificados con un satélite)
Intermittent	*Opportunistic*	Contactos no planificados que se presentan de forma imprevista (ej. una conexión inalámbrica esporádica)
Intermittent	*Predicted*	Contactos no planificados que son estimados mediante la estadística de conexiones observadas y otra información.

El tercer elemento imprescindible para una red DTN consiste en la **adecuación de las aplicaciones**. Las aplicaciones no deben estar diseñadas para una comunicación orientada a conexión mediante un sistema de conversación -como ocurre en una red IP, por el contrario, deben diseñarse para trabajar con una comunicación asimétrica orientada a mensajes. Su desarrollo debe ser preparado para:

a. Tolerar reinicios de sistema.
b. Tolerar interrupciones de acceso a red.
c. Minimizar el número de intercambios de ida y vuelta.
d. Informar a la red del tiempo útil de la información y la importancia de los datos.

En las comunicaciones de las misiones espaciales los recursos son limitados y es necesario su máximo aprovechamiento. Un paquete transmitido que ha utilizado unos recursos y un tiempo de transmisión en un canal de comunicación es valioso. Es costoso realizar reenvíos de paquetes o reintentos de conexión por incidencias utilizando una arquitectura TCP/IP, incluso puede ser inviable la transmisión de información durante ciertos periodos de tiempo. Para la comunicación en las misiones espaciales se está incrementando es uso de DTN y se pretende normalizar su utilización. En este escenario la red DTN es determinista y los contactos son conocidos, así como las características de los enlaces y la capacidad de transmisión.

2.2.3. Arquitectura DTN

Se debe tener en cuenta que en los borradores para la estandarización de los protocolos en los que está trabajando DTNWG en IETF (llamado de forma simplificada BP versión 7) modifican conceptos y características definidos en los documentos generados por DTNRG en IRTF (llamado de forma simplificada BP versión 6). Estas modificaciones se van indicando en las próximas secciones.

2.2.3.1. Enfoque de la arquitectura DTN

El principio de la arquitectura DTN ofrece un cambio sobre la idea de su concepción con respecto a TCP/IP. Internet está construido con un enfoque telefónico donde un remitente[4] llama[5] al destinatario[6], y al responder, se inicial una comunicación. Una vez finalizada el intercambio de mensajes el remitente finaliza la llamada[7]. Cabe señalar que, aunque se trate de un enfoque telefónico, entendemos que tratamos redes conmutadas por paquetes; no se hace referencia a una red conmutada por circuitos.

En cambio, DTN está construido con un **enfoque postal o de correo**. En este caso, el remitente envía un paquete -sin establecer una conexión dependiente del enlace- con el destinatario. El sistema de correo se encarga de identificar la ubicación del destinatario y establecer la ruta del paquete. Durante la ruta, el paquete es custodiado cuando es necesario antes del siguiente movimiento en ruta. El sistema de correo también utiliza notificaciones para emisor, receptor, y puntos de intercambio en ruta. En los siguientes apartados se detallan las características de la arquitectura DTN que hace comprender mejor la analogía del correo postal.

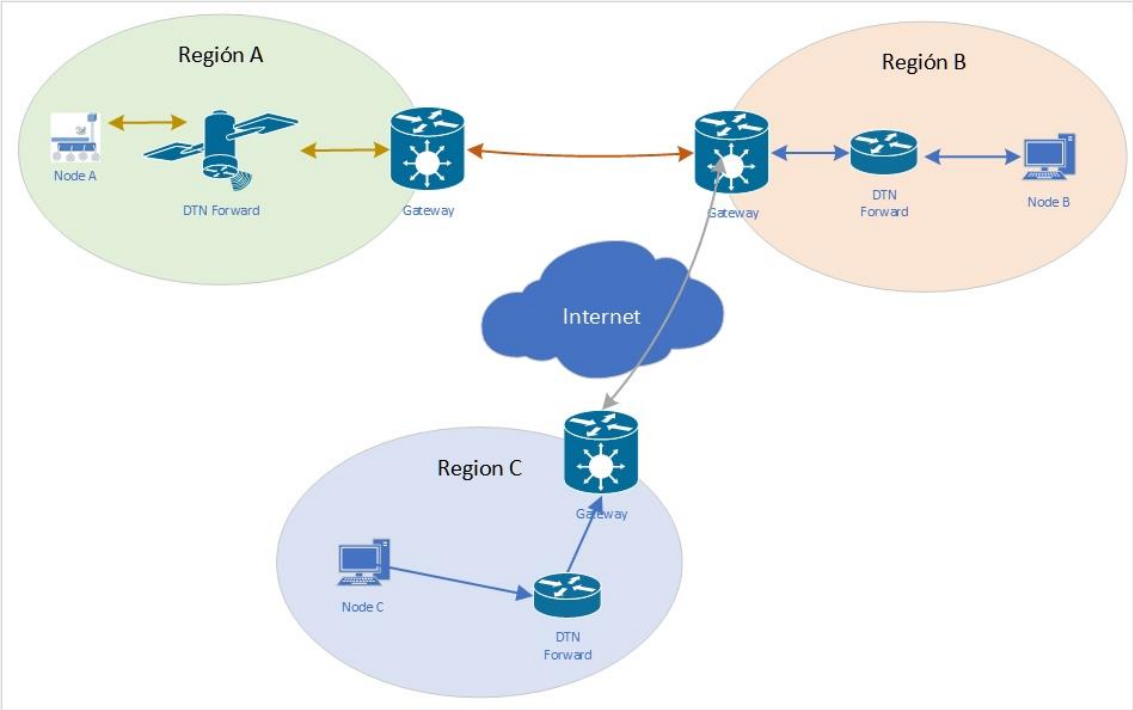
2.2.3.2. Regiones y puertas de enlace

DTN tiene la finalidad de operar como una capa superior a pilas de protocolos de redes existentes. Cualquier dispositivo puede pertenecer en una red DTN, comunicándose con otros dispositivos que usan el mismo protocolo, y además pueden actuar como puerta de enlace o ***Gateway*** entre diferentes redes con diferentes protocolos o familias de direccionamiento. Los nodos pertenecientes a una red y que se comunican entre ellos sin utilizar un *gateway*, pertenecen a lo que se denomina región o ***region*** (figura 6). Los *gateways DTN* son los

[4] Analogía donde el remitente es, por ejemplo, el host A el cual corre una aplicación que abre un socket con una IP y un puerto TCP con el que pretende inciar una sesión con un host B.
[5] Analogía a una sesión del host A desde su IP y puerto hacia la IP y puerto del host B.
[6] Analogía donde el destinatario es el host B, el cual corre una aplicación que tiene un socket con una IP y puerto TCP que está en escucha para recibir conexiones.
[7] Analogía a un cierre de sesión en la comunicación entre el host A y el host B una vez finalizado el intercambio de mensajes.

responsables de almacenar mensajes que se envían a otras regiones y mapear entre diferentes transportes, resolviendo nombres globales a los nombres locales de una región. Además, pueden ejercer una función de autenticación y control de acceso para validar el reenvío de tráfico.

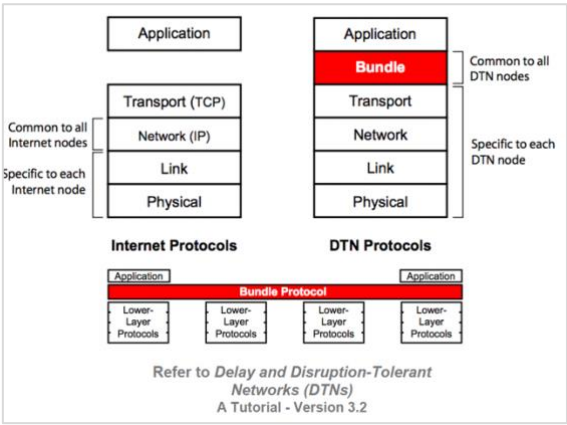
Figura 6: Regiones y gateways.



2.2.3.3. Bundles, Nodos y endpoints

La capa de protocolo de mensajes en un enlace DTN punto a punto se llama ***bundle layer*** (figura 7) [19], que existe sobre las capas de transporte de redes a la que pertenece un host y sus aplicaciones. Un dispositivo que implementa la capa bundle se llama nodo DTN o ***DTN node***. La capa bundle se diferencia del diseño de Internet, por ser independiente a los protocolos de trasporte y enfocarse en el reenvío virtual de mensajes o ***virtual message switching*** en lugar del intercambio de paquetes o *packet switching*.

Figura 7: Bundle Protocol.



(fuente: DTN A Tutorial, versión 3.2)

Se considera un intercambio virtual, ya que los mensajes pueden no reenviarse al instante por falta de enlaces disponibles, quedando a la espera en un almacenamiento persistente y siendo enviados cuando el enlace apropiado esté habilitado.

Una aplicación DTN envía mensajes de una longitud variable, llamados *Application Data Units* (ADUs). Las ADUs pasan a *bundle layer* en unidades de datos de protocolo o *protocol data units* (PDUs) llamados ***bundles***. Un *bundle* tiene las siguientes características:

a. Está formado por dos o más **bloques.**
b. Puede ser fragmentado en otros *bundles* más pequeños.
c. Persiste a reinicios del sistema.
d. Contiene una marca de tiempo o *timestamp* de origen.
e. Contiene un indicador de tiempo de vida útil.
f. Contiene una designación de clase de servicio o *class of service* (CoS). Esta designación desaparece en la versión 7 del protocolo.
g. Contiene una longitud.

En la sección 2.1.4 se amplia la información sobre el formato de los *bundles*.

Un **DTN *endpoint*** es un conjunto de nodos. Un *endpoint* puede estar formado por un solo nodo. Un *bundle* se considera que ha sido entregado correctamente a un endpoint cuando un mínimo subconjunto de nodos en el *endpoint* ha recibido el *bundle* sin error. Este subconjunto de nodos se llama grupo de recepción mínimo **o *minimum reception group*** (MRG) de un *endpoint.* Un nodo puede pertenecer al MRG de múltiples *endpoints*.

2.2.3.4. Endpoint Identifer y los esquemas "ipn" y "dtn".

Un endpoint DTN se identifica en la red mediante un nombre llamado ***Endpoint Identifier*** (EID). Este nombre tiene una sintaxis *Uniform Resource Identifier*[8] (URI). Un URI se compone por un nombre de esquema o *globally-managed scheme*, que está gestionado de forma global y mantenido por IANA. La segunda parte es el esquema específico o scheme-specific part (SSP).

Según la definición de la arquitectura DTN, un EID no se requiere para estar relacionado con el enrutamiento o la topología de la red, aunque deja abierto la implementación y el uso de EIDs para organizar la topología y la organización de las redes DTN y de las regiones. En la implementación de una red DTN se debe tener en cuenta cómo interpretar un SSP para determinar si se refiere a unicast, multicas o anycast.

Un **registro** se produce cuando una aplicación se establece en un EID para recibir ADUs.

[8] La sintaxis URI se utiliza para designar nombres y direcciones para multiples propósitos y está descrita en la RFC 3986.

Para identificar un EID se utilizan dos esquemas URI:

- Esquema “dtn”: identificación arbitraria expresada en cadenas de caracteres.

  Ejemplos:

  `dtn://redesysistemas.uoc.edu/correo` (EID específico de aplicación)

  `dtn://redesysistemas.uod.edu` (EID administrativo)

  `dtn:` (EID nulo[9])

- Esquema “ipn”: identificación mediante pares de números no negativos.

  Ejemplos:

  `ipn:1214:32` (EID específico de aplicación)

  `ipn:1214:0` (EID administrativo)

  `ipn:` (EID nulo)

2.2.3.5. Anycast, multicast y clases de prioridad y fragmentación

La transmisión DTN se considera:

k. ***Unicast***: cuando el MRG se refiere a un único nodo del *endpoint*.
l. ***Anycast****:* cuando el MRG se refiere a un subconjunto de nodos del *endpoint.*
m. ***Multicast***: cuando el MRG se refiere a todos los nodos del *endpoint.*

La clase prioridad o ***priority class*** de una aplicación DTN se transporta en un bloque del *bundle* correspondiente. Esta prioridad puede adaptarse a lo largo de los reenvíos según las políticas de prioridad de los nodos que participan en la ruta del paquete. Las clases de prioridad son relativas a un mismo EID de origen. Es decir, las prioridades de un origen no afectan a las prioridades de otro origen. En el caso que un origen envía un *bundle* con prioridad alta, este envío no tendrá prioridad a un *bundle* de prioridad baja de otro origen. Las prioridades corresponden a la versión 6 del protocolo.

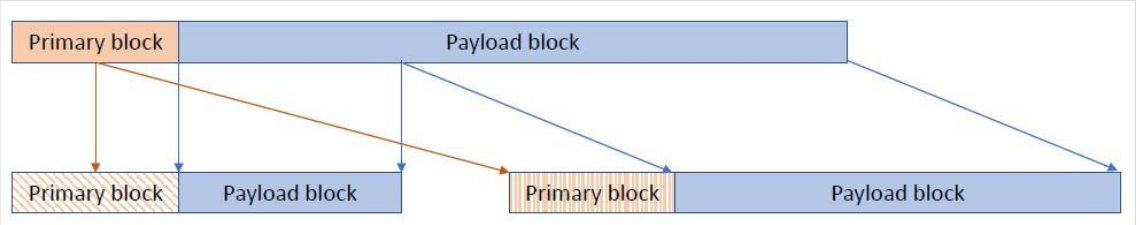
Existen cuatro **prioridades** representadas con dos bits:

a. Bulk (00): Son los *bundles* de menos prioridad y no se transmitirán hasta que los paquetes con prioridades más altas del mismo origen sean enviados.
b. Normal (01): prioridad intermedia.
c. Expedited (10): prioridad alta. Estos mensajes se envían antes que los marcados con las prioridades más bajas.
d. Reservado (11): reservado para futuros usos.

[9] Null *endpoint* es, por definición, un *endpoint* que no tiene ningún miembro.

En DTN la **fragmentación y reensamblado** se utiliza para mejorar la eficiencia. DTN está concebido para el máximo aprovechamiento de los enlaces. En este sentido, los nodos deben ajustar el tamaño de los *bundles* que se van a transmitir fragmentándolos cuando es necesario (figura 8). De esta forma se asegura que, para el tiempo de contacto restante, el paquete transmitido tiene el tamaño adecuado para ajustarse a la capacidad del enlace. Además, esta fragmentación permite la transmisión por múltiples enlaces.

Figura 8: Fragmentación de un bundle.



(Fuente: curso ION 2019 de la NASA).

2.2.3.6. Contactos para la selección de ruta

El enrutamiento de las redes DTN se basa en los tipos de contactos y la forma de tratarlos. Tal y como se ha comentado en la sección 2.1.2, los contactos oportunistas requieren un enrutamiento basado en predicciones y estadística para definir una probabilidad de éxito la trazabilidad de un paquete de origen a destino. En cambio, los contactos deterministas están más orientados al control de los enlaces y al máximo aprovechamiento de los recursos de los sistemas de comunicaciones disponibles. En éstos últimos, si los recursos fueran mucho mayores a los datos a transmitir, no sería necesario realizar esfuerzos para tener definidos con precisión los enlaces y sus características en listas de contactos.

La arquitectura DTN ofrece un marco de trabajo donde se define como un multigrafo[10]. Las aristas del grafo pueden ser distintos en tiempo, retardo, capacidad y direccionalidad[11]. El tiempo que tarda un paquete en ser transmitido por una arista del grafo, se calcula con la función:

$$T(t) = t_0 + D(t) + \left(\frac{1}{C(t)}\right) B \qquad (1)$$

Donde t_0 es el tiempo de inicio de la transmisión en segundos, $D(t)$ es el retardo en segundos, $C(t)$ es la capacidad en bits por segundo del enlace en la arista del grafo, y B son los bits para transmitir. Este periodo de tiempo $T(t)$ -en segundos - se llama **contacto**. El **volumen del contacto** será el producto entre la capacidad $C(t)$ y el contacto $T(t)$.

[10] Un multigrafo es un grafo donde dos vértices pueden conectarse con una o más aristas o *edges*.
[11] Direccionalidad indica que el sentido de la comunicación puede ser unidireccional o bidireccional. Los contactos son unidireccionales.

Se distingue entre enrutamiento o ***routing*** y reenvío o ***forwarding***. *Routing* se refiere a la ejecución de un algoritmo para determinar el camino de origen a destino, y utiliza el estado de la ruta o *routing state* de la base de datos de información de rutas o *routing information base* (RIB). *Forwarding* se refiere a la transmisión de los datos de un nodo a otro utilizando el estado de reenvío o *forwarding state* de la base de información de reenvío o *forwarding information base* (FIB). RIB depende del algoritmo de enrutamiento y FIB depende de la implementación.

Las comunicaciones en las misiones espaciales están planificadas mediante contactos. A partir de esta información se definen técnicas para la selección de las rutas óptimas. En el apartado 2.1.6 se amplia la información sobre los protocolos de enrutamiento.

2.2.3.7. Marcas de tiempo y sincronización

La sincronización de tiempo se utiliza en la arquitectura DTN con los siguientes propósitos:

a. Identificación de fragmentos y *bundles*
b. Enrutamiento con contactos planificados
c. Tiempo de validez de un *bundle*
d. Tiempo de validez del registro de una aplicación

En un *bundle* existe una marca de tiempo del origen. Acompaña a esta marca de tiempo un campo que indica -en segundos- el tiempo de expiración de la validez del paquete. Cada conjunto de *bundles* que corresponden a un ADU debe contener una marca de tiempo única para el EID del emisor. El EID, marca de tiempo y los datos de tamaño y *offset*[12] identifican de forma única un *bundle*. Esta identificación de un paquete se utiliza para la custodia y el reensamblaje de fragmentos.

2.2.3.8. Confiabilidad y custodia

Las aplicaciones deben implementar su propio mecanismo de **confiablidad** en el intercambio de datos punto a punto. Uno de los *flags* del campo de control que está definido para los paquetes del *bundle protocol* -llamado ***acknowledgement-***, se utiliza para ofrecer la opción a las aplicaciones de solicitar acuse de recibo.

[12] *Offset* es un término utilizado en la fragmentación de paquetes de datos para indicar el tamaño que ocupan los datos -*payload* o carga útil de datos sin cabeceras de protocolo- de todos fragmentos anteriores. Por ejemplo, si un paquete de 3072 bytes se fragmenta en tres paquetes de 1024 bytes, el primer fragmento tendrá un *offset* igual a 0 bytes, ya que no existe un fragmento anterior. El segundo fragmento tendrá un *offset* igual a 1024 bytes correspondiente al tamaño del *payload* del primer fragmento. El tercer fragmento tendrá un *offset* igual a 2048 bytes correspondiente a la suma de la carga útil del primero y segundo fragmento.

En la red DTN la custodia se refiere al almacenamiento del bundle en almacenamiento persistente durante su tiempo de vida útil hasta que es posible su reenvío. Un *bundle* transmitido con la opción de solitud de transferencia de custodia o ***custody transfer*** *requested option*, está solicitando la custodia del paquete al siguiente nodo -o siguientes nodos- en la ruta a destino. El objetivo es conseguir que el paquete quede almacenado progresivamente lo más cerca posible del destino a medida que se transfiere de un nodo a otro en su ruta.

La transferencia de custodia permite a los nodos delegar la responsabilidad del envío de los datos, recuperando recursos una vez enviado el *bundle*. No todos los nodos deben aceptar la custodia y pueden ofrecer únicamente su reenvío. Un bundle contiene el EID que actualmente tiene la custodia o *Current Custodian* EID. Un nodo que acepta la custodia envía una aceptación o *Custody Transfer Accepted Signal* al *Current Custodian* EID.

Las aplicaciones envían sus ADU a la capa BP solicitando las siguientes opciones de entrega relacionadas con la custodia:

a. *Custody Transfer Requested*: la aplicación solicita el uso de procedimientos de custodia.
b. *Source Node Custody Acceptance Required*: la aplicación requiere que la custodia se realice por parte el emisor. Si el emisor no tiene capacidad de custodia la transmisión fallará.
c. *Report When Bundle Delivered*: la aplicación solicita un informe de entrega del ADU.
d. *Report When Bundle Acknoledged by Application*: la aplicación solicita el informe de acuse de recibo a la aplicación destino en lugar de al host destino como en el punto c. Esta opción se puede utilizar en *gateways* que enlazan diferentes redes con distintos protocolos, para no perder los acuses de recibo en los intercambios de protocolos y delegarlos directamente a la capa de aplicación.

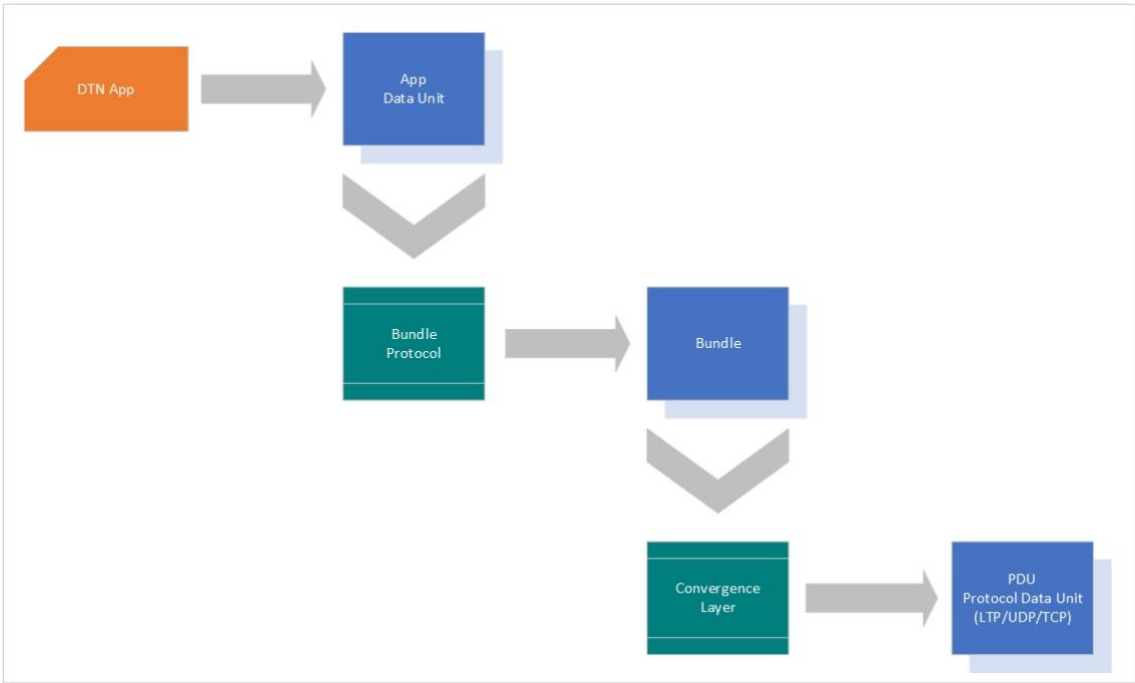
Puede ocurrir que el *Custody Transfer Accepted Signal* no pueda ser enviado al *Current Custodian* EID o que el paquete salte por diferentes nodos sin obtener el servicio de custodia. En estos casos, el *Current Custodian* EID se marcará como *NULL*. Una aplicación puede requerir la aceptación de custodia con Source *Node Custody Acceptance Required*. En este caso la aplicación se asegurará encadenar los nodos con aceptación de custodia y confiar en que los datos están almacenados durante la ruta.

2.2.3.9. Capas de convergencia

La arquitectura DTN como capa intermedia entre las aplicaciones y las capas de red involucradas en la transmisión para los enlaces de los nodos, requiere de una capa de una fase **adaptación** que se produce en la capa de convergencia o ***Convergence Layer*** (figura 9). La capa de

convergencia es diferente para cada familia de protocolos y es un interfaz que maneja los detalles específicos para cada protocolo, ofreciendo consistencia a la capa *bundle*. En la sección 2.2.5. se describirán tres adaptaciones de protocolos para la capa de convergencia según los nuevos borradores de DTNWG de IETF.

Figura 9: Proceso DTN desde la aplicación hasta la capa de convergencia.



2.2.3.10. Congestión y control de flujo

Control de flujo o *flow control* se refiere a la limitación del ratio de envió en un enlace de un nodo al siguiente nodo en la ruta. Esta limitación está orientada a que el nodo de origen no exceda la capacidad del nodo destino en la transmisión y que el receptor está preparado para recibir los datos. El control de flujo se resuelve mediante la implementación y los protocolos de enrutamiento, donde se calcula la ruta en función de los contactos planificados y definidos con una capacidad de transmisión específica. Además, se debe tener en cuenta los elementos de control de flujo de los protocolos de capas inferiores que formarán parte de la red, así como la capa de convergencia en el diseño de una red DTN.

Control de congestión o *congestion control* se refiere a la gestión del almacenamiento persistente para la custodia de *bundles*. Las características de la red DTN -en cuanto a interrupciones de conexión y almacenamiento persistente- hacen que sea una red vulnerable a congestión. Existen un gran número de propuestas para la gestión de la congestión con esquemas proactivos y reactivos, implicando las aplicaciones, el protocolo de enrutamiento y combinado con diferentes mecanismos como se puede ver en [15]. El control de congestión es un elemento fundamental

en DTN. Se espera recibir atención por la comunidad de investigadores de DTN y acertar los métodos adecuados y eficaces, una vez se comprueben en el despliegue de redes DTN más allá de las simulaciones.

2.2.3.11. Seguridad

En este apartado se trata la seguridad para la versión 6 descrita en RFC 6257 además del borrador de la arquitectura de seguridad BPSec que acompaña la propuesta de estándar BP7.

Gran parte de las técnicas para ofrecer seguridad están desarrollados para redes IP. Las funciones de seguridad de redes se pueden clasificar en cuatro categorías:

a. **Autenticación**: concepto que significa identificar una identidad, es decir, quién o qué sistema tiene permiso de acceso al mensaje. También puede determinar quién ha sido el último en reenviar el mensaje. Son servicios que se centran en quién maneja el mensaje y tiene acceso a su gestión, envío o recepción. Por ejemplo, una infraestructura de clave pública o *Public Key Infraestructe* (PKI) permite a los usuarios identificarse frente a otros usuarios.
b. **Integridad**: servicios que protegen el mensaje en tránsito para que no sean modificados. La integridad se consigue aplicando un algoritmo al mensaje para obtener un código resultante (*hash*) que representa el resumen del mensaje. Se le llama resumen ya que el *hash* se genera con un tamaño reducido determinado por la implementación. Obtener el mismo *hash* desde dos mensajes distintos se llama colisión y es poco probable y todavía menos probable obtener el mismo *hash* de un mensaje al modificarlo. Cuando mayor sea el tamaño del *hash*, menos probable es que se produzcan colisiones. Si se modifica el mensaje el *hash* cambiará y se podrá identificar que los datos han sido modificados. Por ejemplo, los dos algoritmos más utilizados son MD5 y SHA.
c. **Confidencialidad**: servicios para prevenir que los datos contenidos en el mensaje puedan ser leídos. La confidencialidad se consigue aplicando al mensaje con un algoritmo matemático (*cipher*) para producir un nuevo mensaje (*cipher text*) para que el remitente realice el proceso inverso y recuperar el mensaje original. Estos procesos de conversión del mensaje en datos inteligibles y recuperación del mensaje original se llaman encriptación y desencriptación. Por ejemplo, un algoritmo que realiza este procedimiento es AES.
d. **Disponibilidad**: Servicios para reducir la probabilidad de que la red no esté disponible desde un enfoque de seguridad de red o ciberseguridad. Por ejemplo, los ataques de denegación de servicio o *Denial of Service*[13] (DoS) o los ataques distribuidos de

[13] Es un ataque a un sistema con el objetivo anular sus funciones. Un ataque DoS utiliza técnicas como consumir, alterar o interrumpir los recursos de un sistema.

denegación de servicio *Distributed Denial of Service*[14] (DDoS). Los ataques DDoS son problemas de seguridad sofisticados y los más difíciles de frenar. Los cortafuegos de nueva generación o *Next Generation Firewalls*[15] (NGF) son sistemas proactivos, diseñados para identificar ataques con un procedimiento de identificación y acción, pero no son infalibles ante ataques DoS o DDoS. Estos sistemas se pueden combinar con otras acciones, por ejemplo, la distribución geográfica del servicio en diferentes centros de datos, distintos enlaces de red y otras contramedidas.

Los **componentes de seguridad** en la arquitectura DTN están orientados a los siguientes **objetivos**:

a. Evitar la transmisión de datos cuando la aplicación no está autorizada.
b. Evitar el control de la red DTN por parte de aplicaciones no autorizadas.
c. Evitar la transmisión de datos con un ratio de transferencia o con una clase de permiso no autorizado.
d. Descartar *bundles* modificados en tránsito.
e. Detectar elementos de la red comprometidos.

Las **aplicaciones** envían sus ADU a la capa BP solicitando las siguientes **opciones de entrega** relacionadas con la seguridad:

a. *Confidentiality Required*: la aplicación requiere que el ADU sea confidencial para el origen y el EID destino.
b. *Authentication Required*: la aplicación requiere la verificación de acceso al ADU además de los datos del *bundle* como EIDs de origen y destino.
c. *Error Detection Required*: la aplicación requiere la detección de modificación de datos para asegurar la integridad en la transmisión.

El modelo de seguridad DTN está enfocado a adaptar la situación de uso. DTN es una arquitectura que forma una capa superpuesta a las redes existentes, con las cuales interactúa con una interfaz de convergencia. Este hecho provoca que la seguridad se debe implementar con un enfoque por capas y por topología y uso. Es por ese motivo, que el **modelo de seguridad DTN** analiza los siguientes elementos:

[14] DDoS consiste en un ataque coordinado desde diferentes atacantes que realizan un DoS simultaneamente, lo que aumenta la probabilidad de anular la funciones del sistema atacado.
[15] Un *firewall* es un dispositivo de ciberseguridad situado entre dos o más segmentos de red, que analíza el tráfico de datos con el objetivo de proteger de ataques a uno o varios de los segmentos. NGF hace referencia a una generación de *firewalls* que añaden técnicas de análisis de tráfico de red, para detección de intrusiones y ataques con mecanismos de respuesta, ademas de otros servicios dinámicos y proactivos.

a. Lista de capas protocolos utilizados en la red DTN y las implementaciones de seguridad para cada una de las capas.
b. Adaptación de la seguridad en cuanto a políticas y configuraciones para cada situación.

En Bundle ***Security Protocol Specification*** (RFC 6257) [20] se utilizan los bloques de extensión de BP para que el propio paquete transporte la implementación de los servicios de seguridad. El protocolo BP soporta los servicios para la autenticidad y la confidencialidad. Se describen cuatro bloques que pueden incluirse en un *bundle*, como se puede ver en la tabla 5.

Tabla 5: Bloques y cifrado Bundle Protocol 6.

Bloque	**Descripción**	**Cifrado obligatorio[16]**
Bundle Authentication Block (BAB)	Utilizado para asegurar la autenticidad y la integridad del *bundle* en la transmisión de un solo salto, de un nodo a otro	HMAC-SHA1[17]
Payload Integrity Block (PIB)	Utilizado para asegurar la autenticidad y la integridad del *payload* entre el emisor y el receptor	RSA[18]-SHA256
Payload Confidentiality Block (PCB)	Utilizado para asegurar confidencialidad del *payload* mediante encriptación	RSA-AES128[19]. Encriptación del algoritmo AES con Galois/Counter Mode (GCM)[20]
Extension Security Block (ESB)	Utilizado para proveer seguridad a otros bloques. No se aplica a los bloques PIB, PCB, bloque principal y bloques con *payload*	RSA-AES128. Encriptación del algoritmo AES con Galois/Counter Mode (GCM)

(Fuente: RFC 6257).

[16] Los cifrados obligatorios se describen en la RFC 6257 y se refieren a la especificación de *Bundle Security Protocol*.
[17] HMAC-SHA1 es un algoritmo que calcula el código de autentificación de un mensaje o *Message Authentication Code* (MAC) y aplica la función *hash* SHA1 en combinación con una clave criptográfica secreta. Utilizado para ofrecer integridad de datos y autenticación del mensaje.
[18] Rivest, Shamir, Adleman (RSA) es un algoritmo de cifrado y firma digital.
[19] Advanced Encription Standard (AES) es un sistema de cifrado por bloques.
[20] GCM es un modo de operación para la encriptación que ofrece confidencialidad y autenticación.

La especificación *Bundle Protocol* Security (BPSec) [21] es un borrador de DTNWG y complementa al borrador de *Bundle Protocol* Versión 7. El borrador BPSec define dos bloques de extensión (tabla 6).

Tabla 6: Bloques y cifrado Bundle Protocol 7.

Bloque	Descripción	Cifrado Obligatorio
Block Integrity Block (BIB)	Utilizado para asegurar la autenticidad y la integridad del *payload* entre el emisor y el receptor	HMAC-SHA2[21] utilizado en tres variantes: HMAC 256/256, HMAC 384/384 y HMAC 512/512.
Block Confidentiality Block (BCB)	Utilizado para asegurar confidencialidad del *payload* mediante encriptación	AES-GCM utilizado en dos variantes: A128-GCM y A256-GCM.

2.2.3.12. Consideraciones de administración

Existen registros administrativos en la capa BP equiparables al protocolo ICMP[22] en las redes IP.

Por una parte, las **aplicaciones** envían sus ADU a la capa BP solicitando las siguientes **opciones de entrega con propósitos de diagnóstico**. Su uso puede ser limitado en entornos con congestión o bajo ataques:

a. *Report When Bundle Received*: la aplicación solicita un informe de recepción generado cuando cada *bundle* enviado llega a un nodo.
b. *Report When Bundle Custody Accepted*: la aplicación solicita un informe de aceptación de custodia cuando cada *bundle* enviado ha sido aceptado usando una transferencia de custodia.
c. *Report When Bundle Forwarded*: la aplicación solicita un informe de estado de envío para cada *bundle* llegado a un nodo después de ser enviado.
d. *Report When Bundle Deleted*: la aplicación solicita un informe de estado de eliminación generado para cada *bundle* que es borrado en un nodo.

Por otra parte, los diferentes **tipos de registros** definidos en la arquitectura DTN están listados en la tabla 7.

[21] Secure Hash 2 (SHA-2) es una versión de SHA que resuelve las vulnerabilidades de SHA1.
[22] Internet Control Message Protocol (ICMP) usado para envío de información y mensajes de error.

Tabla 7: Tipos de registros.

Registro	Tipo	Descripción
Bundle Reception	BSR[23]	Enviado cuando un *bundle* llega a un nodo DTN
Custody Acceptance	BSR	Enviado cuando un nodo ha aceptado la custodia de un *bundle* cuando se usa la opción *Custody Transfer Requested*
Bundle Forwarded	BSR	Enviado cuando un *bundle* contiene la opción *Report When Bundle Forwarded* enviado de un nodo DTN
Bundle Deletion	BSR	Enviado desde un nodo DTN cuando se descarta un *bundle* que contiene la opción *Report When Bundle Deleted*
Bundle Delivery	BSR	Enviado por el nodo destinatario final cuando se completa la recepción de un ADU cuyos *bundles* contienen la opción *Report When Bundle Delivered*
Acknowledged by application	BSR	Enviado por un nodo destinatario final cuando se completa la recepción de un ADU cuyos *bundles* contiene la opción *Application Acknowledgment*. Este registro generalmente involucra la acción de la aplicación
Custody Signal	Custody Signal	Indica que la custodia ha sido transferida. Es una señal expresada con un valor booleano indicando si la transferencia de custodia ha sido satisfactoria o ha fallado

(Fuente RFC 4838).

2.2.4. Bundle Protocol

BP está implementado en dos versiones: la versión 6 y versión 7. En enero de 2015 se publica el primer borrador del protocolo Bundle Protocol versión 7 (BP7) en IETF, con el estatus *Proposed Standard* [22]. El último borrador durante la redacción de esta memoria es de enero de 2021. El borrador de BP7 está basado en el RFC 5050 publicado por IPNRG de IRTF, que debemos recordar que es una publicación experimental y desde donde se construye la versión 6

[23] Bundle Status Reports (BSR).

de BP. BP7 es una propuesta de estándar que introduce algunos cambios con respecto a las características descritas en RFC 5050 (BP versión 6).

Se debe tener en cuenta que existen implementaciones disponibles que no han sido actualizadas o no son compatibles con BP7. Siendo DTN una tecnología en pleno desarrollo y proceso de estandarización, esta sección se ocupará de ofrecer las últimas propuestas BP7 y algunas comparaciones con la versión BP6 que todavía está en uso.

En el protocolo BP se utilizan los siguientes componentes:

a. ***Bundle Protocol Agent*** (BPA): es el componente que ofrece servicios BP al nodo.
b. ***Application Agent*** (AA): es el componente del nodo que accede a los servicios del BPA y se compone de dos elementos:
    1. Elemento específico de aplicación: construye, solicita transmisiones y acepta retransmisiones de ADUs.
    2. Elemento administrativo: construye y acepta retransmisiones de registros administrativos, los cuales son ADUs de propósito administrativo.
c. ***Convergence Layer Adapter*** (CLA): es el componente del nodo que envía y recibe *bundles* con origen -o destino- del BPA hacia las pilas de protocolos de transporte disponibles en el nodo.

Entre los diferentes cambios en BP7 con respecto BP6, cabe destacar los siguientes:

a. Concepto de *node ID* con la misma sintaxis que *endpoint ID* pero distinta funcionalidad. Cada nodo queda registrado con un EID como *singleton*[24] en el *endpoint* administrativo del nodo. Este EID administrativo es el *node* ID.
b. Cambio de la representación SDNV (apartado 2.2.4.2.) por CBOR (apartado 2.2.4.3.).
c. Restructuración de *Primary Block* que pasa a ser inmutable[25].
d. Valores de tiempo en milisegundos en lugar de en segundos.
e. Añadida la opción de utilizar CRCs[26] (CRC-16 y CRC-32) en los bloques del *bundle*.
f. Diferenciar entre:
    - Transmisión o *transmission*: BPA en un nodo copia el bundle para entregarlo al *endpoint* correspondiente en respuesta a una petición de AA.
    - Reenvío o *forwarding*: el nodo invoca los servicios de los CLAs para conseguir que un *bundle* sea enviado a un nodo destino.

[24] Un *singleton* es un *endpoint* con un único nodo.
[25] *Primary Block* no se modificará desde el momento de su creación hasta su entrega, es decir, será inmutable.
[26] *Cyclic Redundancy Check* (CRC) es un código para verificar errores en los datos. No es un método de seguridad para evitar la manipulación de datos. Es un métrodo para detectar errores en la transmisión.

BP tiene las siguientes funcionalidades:

a. Retransmisión basado en custodia.
b. Afrontar conectividad intermitente.
c. Aprovechar las características de una conectividad planificada, oportunista o estadística.

Además BP ofrece los siguientes servicios:

a. Registro:
    1. Registrar un nodo en un *endpoint*.
    2. Finalizar el registro de un nodo en un *endpoint*.
    3. Cambiar un registro de un nodo entre los estados Activo y Pasivo.
    4. Agrupar los registros que están en un estado Pasivo.
b. Transmisión de *bundles*:
    1. Transmitir un *bundle* a un *endpoint*.
    2. Cancelar una transmisión.
    3. Entregar un *bundle* recibido.

En el anexo 1 se muestran tablas comparativas de Bundle Protocol para la versión 6 y versión 7.

2.2.4.1. Self Delimiting Numeric Values (SDNVs) para BP6

Este apartado describe la representación SDNV utilizada para la versión BP6. DTNRG utilizó *Self Delimiting Numeric Values* (SDNVs) [23] con el propósito de limitar al máximo el consumo de recursos y ancho de banda, además de dejar espacio para añadir más datos y adaptarse a distintas capas de protocolos de transporte. SDNV es una forma de representar enteros no negativos con pocos recursos. Es interesante observar el formato **SDNV** ya que **se usa en los campos que forman parte de los bloques de los *bundles* y es una codificación que nos podemos encontrar** en algunas implementaciones de la versión 6.

SDNV consiste en representar un número entero representado en N octetos. El último octeto tiene su bit más significativo o Most Significant Bit (MSB) a cero. El resto de los octetos necesarios tienen el MSB a uno. El primer octeto se rellenará de ceros hasta el uno del MSB. Algunos ejemplos de SDNV:

```
0 ceros de relleno.
0 ceros eliminados.
1 MSB de un octeto que no es el último octeto que codifica el número entero
  representado.
0 MSB de un octeto que es el último que codifica el número entero
  representado.
```

```
2748                        (decimal)
0xABC                       (hexadecimal)
101010111100                (binario)
Codificado como:
            octeto 1        octeto 2
          {1 00 10101}     {0 0111100}
             = 10010101 00111100
```

```
16948                       (decimal)
0x4234                      (hexadecimal)
0100 0010 0011 0100         (binario)
       =  100 0010 0011 0100
Codificado como:
           octeto 1    octeto 2    octeto 3
         {1 000000 1} {1 0000100} {0 0110100}
         = 10000001 10000100 00110100
```

```
127                         (decimal)
0x7F                        (hexadecimal)
0111 1111                   (binario)
       =  111 1111
Codificado como:
          octeto 1
         {0 1111111}
         = 01111111
```

2.2.4.2. Concise Binary Object Representation (CBOR) para BP7

Este apartado describe la representación CBOR [24] utilizada en la propuesta de estándar de BP7. El propósito de esta representación, además de una codificación compacta, tiene que ver con la capacidad de representar **diferentes tipos de datos** y un obtener un resultado **estructurado**. La especificación BP7 describe la representación CBOR que se debe utilizar en cada campo que forma el protocolo. Al igual que ocurre con SDNV, es interesante conocer la codificación CBOR ya que aparecerá en los campos del protocolo BP7 así como en las implementaciones.

CBOR está basado en JSON y está formado por **elementos de datos**. Un elemento de datos tiene una **cabecera** y una **carga de datos o *payload*** opcional. La cabecera de 1 byte contiene información sobre las características de los datos del elemento. La cabecera se codifica en dos campos, el primero formado por 3 bits y el segundo por los restantes 5 bits. Los primeros 3 bits se utilizan para identificar uno de los 8 tipos de datos -llamados ***major types***- definidos en CBOR. Los siguientes 5 bits se utilizan para introducir el dato directamente -embebido si es un valor menor que 24- o especificar el tamaño en bytes del dato, y es en este segundo caso, cuando

aparece la carga de datos opcional del elemento. La tabla 8 identifica los diferentes tipos de datos y sus características.

Un elemento CBOR puede estar compuesto por conjuntos de datos de diferentes tipos. Estos conjuntos de datos pueden estar anidados. Además, está permitida la representación de las cadenas de datos tanto con tamaño específico o ***definite-length***, como con tamaño no definido o ***undefinite-lenght***. Cuando el tamaño de datos no está definido, los datos se forman con una cadena de bytes "*chunck*", y por último un byte con un código de stop[27] o "*break*" que representa el final de la cadena.

Tabla 8: Tipos de datos CBOR y características.

Major type number	**3 bits cabecera**	**Major type name**	**Descripción**
0	000	*Unsigned integer*	Entero sin signo en el rango de 0 a 2^{64}-1. El dato puede ser embebido en el byte de cabecera si es menor a 24.
1	001	*Negative integer*	Entero negativo en el rango de -2^{64} a a -1. El dato codificado es -1-N. El dato puede ser embebido en el byte de cabecera si está entre -1 y -24.
2	010	*Byte string*	Cadena de bytes.
3	011	*Text string*	Cadena de texto codificada con UTF-8.
4	100	*Array*	Secuencia de datos. Puede combinar a su vez diferentes tipos de datos y estar anidado.
5	101	*Map*	Pares de datos formados por clave/valor.
6	110	*Tag*	Etiquetas. Suele utilizarse para añadir información en conjuntos de datos.
7	111	*Simple/float*	Números representados en coma flotante. Los valores simples son *true*, *false*, *null* y *undefined*.

(fuente: RFC8949).

Los detalles de los procesos de representación de conjuntos de datos complejos con CBOR, quedan fuera del alcance de este texto. A continuación se describen los procesos para conjuntos de datos simples, como números enteros y una cadena de texto:

[27] El byte que representa el código de *stop* en un elemento CBOR -o dato empaquetado en un conjunto de datos en el elemento CBOR- que no tiene el tamaño definido es 0xFF.

Ejemplo 1

Tipo de dato a representar: entero sin signo

Dato: 10

Dato en bits: 1010

Major type: 0

Cabecera:

3 bits type: 000 (Major type 0)

5 bits (dato embebido): 01010

Codificación CBOR (bin): 00001010

Codificación CBOR (hex): 0xA

Descripción de la codificación: 10 es un número entero sin signo y menor de 24 por lo que puede ser embebido en los 5 bits de la cabecera.

Ejemplo 2

Tipo de dato a representar: entero negativo

Dato: -100 >> -1-(-100) = 99

Dato en bits (99): 01100011

Major type: 1

Cabecera:

3 bits type: 001 (Major type 1)

5 bits (tamaño en bytes): 11000

Codificación CBOR (bin): 00111000 01100011

Codificación CBOR (hex): 0x3863

Descripción de la codificación:

El dato para codificar -100 es un número negativo que se representará con la normalización -1-N = -1-(-100) = 99. Al no ser un número menor que 24, 99 no irá embebido en el campo de la cabecera de 5 bits, y este cambo deberá indicar el tamaño en bytes del dato. El decimal 99 en binario es 01100011, por lo que se puede representar con un byte, y éste será el tamaño de los datos del elemento CBOR con el que estamos trabajando.

Es de tipo 1. La cabecera se codifica con los correspondientes 3 bits de tipo 001. La cabecera se completa indicando el tamaño de datos de 1 byte en el campo de 5 bits. Para indicar este tamaño en el campo de 5 bits, hay que tener en cuenta que, cuando los 5 bits indican una valor menor que 24, se refiere a un valor embebido, por lo que si los 5 bits indican un tamaño de datos -en este caso se debe indicar 1 byte- se usa el número 24 para indicar dicho byte.

El número 24 en binario es 11000, por lo que la cabecera se completa con 00111000. Ya tenemos la cabecera y el byte que representa el número 99. El byte de cabecera y el byte de datos del elemento CBOR en hexadecimal es 0x3863.

Ejemplo 3

```
      Tipo de dato a representar:     Text string
      Dato:                           "IETF" (UTF-8)
      Dato en bits:                   01001001 01000101 01010100 01000110
      Major type:                     3
      Cabecera:
            3 bits type:              011 (Major type 3)
            5 bits (tamaño en bytes): 00100 (4 bytes)
            Codificación CBOR (bin):  011000100 01001001 01000101 01010100
                                      01000110
            Codificación CBOR (hex):  0x6449455446
      Descripción de la codificación:
```

El dato para codificar es una cadena de texto que se puede representar con 4 bytes. La cabecera codifica el tipo 3 en el campo de 3 bits mediante 011. En este tipo de datos el campo de 5 bits representará el tamaño de 4 bytes mediante 00100. El byte de cabecera junto a los 4 bytes del dato es el elemento CBOR, que codificado en hexadecimal es 0x6449455446.

2.2.4.3. Formato de bloques para la versión 6

Como se ha descrito anteriormente, un *bundle* está formado por una secuencia de dos o más bloques:

a. *Primary block*. Es el bloque principal y obligatorio en cada *bundle*.
b. *Extensión bloc*k. Es un bloque opcional. Puede haber múltiples bloques de extensión. Por ejemplo, BAB, PIB y otros.
c. *Payload block*. Contiene los datos que provienen de los ADUs.

Los campos que conforman el bloque primario y el bloque de carga útil se pueden ver en la figura 10.

Figura 10: Primary block y payload block para la versión 6.

Primary Block	
Version	*Processing Control Flags (*)*
Block Length (a)	
Destination Scheme Offset *(*)*	Destination SSP offset *(*)*
Source Scheme Offset *(*)*	Source SSP offset *(*)*
Report to Scheme Offset *(*)*	Report to SSP offset *(*)*
Custodian Scheme Offset *(*)*	Custodian SSP offset *(*)*
Creation Timestamp Time *(*)*	
Creation Timestamp Sequence Number *(*)*	
Lifetime *(*)*	
Dictionary Length *(*)*	
Dictionary Byte Array	
Fragment Offset *(*)*	
Total Application Data Unit Length *(*)*	

Payload Block	
Block Type	Processing Control Flags (*)
Block Length (*)	
Bundle Payload	

(*) SDNV de longitud variable

(Fuente: [15]).

Los bloque distintos al *primary block* tendrán un formato canónico (*canonical block*), compuesto por los siguientes campos:

- Código de tipo de bloque.
- *Block Processing Control Flags*: indicadores de las propiedades del *bundle*.
- *Block EID reference*.
- Tamaño del bloque.
- Datos para bloque de tipo específico.

2.2.4.4. Formato de bloques para la versión 7

Para la versión 7, se modifican los campos del *primary block*:

- *Version*: versión del protocolo.
- *Bundle Processing Control Flags*: indicadores de las propiedades del *bundle*.
- *CRC Type*: tipo de código CRC.
- *EID* destino: identifica el enpoint destino.
- *Node ID* origen: identifica el nodo que transmitió el *bundle*.
- *Report-to EID*: identifica el *endpoint* al que se debe enviar notificaciones.
- *Creation Timestamp*: marca de tiempo.
- *Lifetime*: tiempo en milisegundos a partir del *creation timestamp* que indica el periodo de validez del *bundle*.
- *Fragment offset*: *offset* en caso de que sea un fragmento.
- *Total Application Data Unit Length*: indica el tamaño total del ADU original generado por la aplicación.
- *CRC*: Campo para CRC del *primary block*. Este campo no se utiliza si el *bundle* incluye BPSec.

Los bloque distintos al *primary block* tendrán un formato canónico (*canonical block*), compuesto por los siguientes campos:

- Código de tipo de bloque.
- Número de bloque.
- *Block Processing Control Flags*: indicadores de las propiedades del *bundle*.
- Tipo de CRC.
- Datos para bloque de tipo específico.
- Si, y solo si, se aplica CRC, se incluye el campo CRC.

2.2.5. Capas de Convergencia

La capa de convergencia o *convergence layer* (CL) es la capa de protocolo entre BP y los protocolos de transporte. Un *convergence layer* adapter (CLA) es una adaptación de CL para un protocolo de transporte específico con unas características determinadas. Se han definido CLAs para TCP, UDP, Licklider Transmission Protocol (LTP), Bluetooth y Ethernet. En este apartado trataremos las CLAs de LTPCL, TCPCL y UDPCL, donde LTPCL es la clara apuesta para los enlaces en el espacio y TCPCL es el principal adaptador de convergencia para las redes IP.

2.2.5.1. LTPCL

LTP es un protocolo diseñado para enlaces en el espacio [25]. LTP tiene las siguientes características:

- Divide la información -de un *bundle*- en diferentes segmentos.
- Forma bloques de segmentos.
- LTP puede transmitir segmentos de diferentes *bundles* de forma concurrente en un mismo bloque.
- Detecta fallos de transmisión y reenvía segmentos perdidos. La unidad de retransmisión es el segmento.
- Los segmentos no tienen por qué llegar al destino en orden y pueden llegar duplicados.
- LTP reordena y reconstruye la información en destino.
- LTP puede contener datos marcados como “red”, que requieren detección de datos perdidos y retransmisión, así como datos marcados como “green”, para los que no se detectan datos perdidos ni hay retransmisión.
- Un mismo bloque puede contener datos “red” y “green”.

2.2.5.2. TCPCL

En esta versión para la capa de convergencia, el adaptador prepara el envío para ser transportado mediante TCP. TCPCL tiene, a su vez, dos versiones:

- TCPCL: definida por IRTFRG en el RFC7242 [26] con la siguiente descripción de protocolo:
    - Un nodo establece una conexión con otro nodo iniciando una conexión TCP.
    - Los nodos intercambian una cabecera de contacto con los parámetros de conexión y el intercambio de los EIDs.
    - TCPCL es un protocolo simétrico entre los puntos. Ambos nodos pueden transmitir información y está permitida la comunicación bi-direccional.
    - Los nodos intercambian *bundles*. Estos *bundles* se envían divididos en segmentos TCP.
    - Los *bundles* se transmiten secuencialmente. Los segmentos de diferentes *bundles* no se mezclan en la transmisión. Únicamente, es posible entrelazar diferentes *bundles* si éstos han sido fragmentados previamente en BP. Aunque, a efectos de TCPCL, cada fragmento de la capa BP es un *bundle* independiente que entrará en la cola de transmisión.

- TCPCL tiene la opción de enviar confirmación[28] de entrega de segmentos, con el objetivo de determinar qué cantidad de información del *bundle* transmitido ha sido entregado. Con esta información, ante interrupciones de transmisión, es posible iniciar un proceso reactivo de fragmentación y transmisión de los datos pendientes de transmitir.
- Otra función opcional es la de rechazar un *bundle* durante la transmisión, con el objetivo de optimizar los enlaces en la detección de datos duplicados.
- La última función opcional es la transmisión de mensajes de mantenimiento de la conexión TCP o *keepalive*.
- Una vez finalizada la transmisión se envía un mensaje de parada. Este mensaje también se utiliza para terminar una sesión con errores.

- TCPCLv4: definida por IETFWG en el I-D TCPCLV4 [27]. En esta versión de CLA se introduce la capacidad de negociar el uso de seguridad TLS durante el establecimiento de la sesión TCP.
- MTCP: *Minimal TCP Convergence-Layer Protocol* [28], es un protocolo unidireccional con protocolo de transporte TCP.

2.2.5.3. UDPCL

El adaptador de capa de convergencia UDPCL [29] es un servicio para transmisión de *bundles* vía UDP especificado en el I-D de IETFWG para DTN. UDPCL es un protocolo unidireccional definido como “transmitir y olvidar” o “*transmit and forget*” que no ofrece respuesta al transmisor. UDPCL soporta añadir seguridad TLS.

2.2.6. Protocolos de enrutamiento

En enrutamiento para las redes DTN es un ámbito todavía abierto y en investigación [30][31][32][33][34][35]. Existen diferentes propuestas de las cuales se pueden destacar dos. Por una parte, para redes DTN donde los contactos son probabilísticos, un protocolo de enrutamiento destacable es *Probabilistic Routing Protocol for Intermittently Connected Networks* (PRoPHET) [36]. Por otra parte, para redes DTN donde los contactos son deterministas, entre las diferentes propuestas [37], el protocolo que actualmente es destacable es *Contact Graph Routing* (CGR).

[28] *Transport Layer Security* (TLS) es un protocolo descrito en RFC4346, utilizado para ofrecer seguridad a las conexiones TCP. Garantiza por una parte la atenticidad del servidor mediante certificados digitales. Por otra parte, asegura la integridad del mensaje con un código MAC. Por último, encripta el mensaje y el código MAC con una clave generada a partir de las firmas digitales.

PRoPHET es un protocolo de enrutamiento epidémico[29] con restricciónes basadas en la probabilidad, es decir, es un protocolo donde los nodos transmiten copias de los paquetes de datos a sus vecinos cuando entran en contacto. El nodo que intenta transmitir el paquete tiene información histórica de los contactos, y puede calcular la probabilidad de que el nodo vecino con el que entra en contacto, una vez identificado, tenga más o menos probabilidad de tener una ruta más corta al destino que él mismo. En el caso el nodo vecino tenga más probabilidad de enrutar el paquete a su destino, se transmite la información, y en caso contrario, no se transmite la información al nodo vecino.

La primera publicación sobre CGR es de 2010 por parte de DTNRG de IRTF. Desde entonces, se han propuesto múltiples variantes pensando en solucionar problemas de congestión y escalabilidad [38]. CGR es propuesto como estándar para las redes DTN por CCSDS en 2019 en [32]. A continuación se describen distintos aspectos del protocolo de enrutamiento.

2.2.6.1. Las claves de Contact Graph Routing (CGR)

En primer lugar hay dos términos interesantes para el enrutamiento. El primero es el significado del tiempo de vida o *Time to Live* (TTL) en una red DTN. TTL en una red IP es el número de saltos que un paquete puede recorrer en la ruta hasta ser descartado. En cambio, en DTN el TTL es realmente el tiempo durante el cual el paquete es válido.

El siguiente término es One-Way Light Time (OWLT)[30], que consiste en el tiempo en milisegundos transcurrido en la transmisión de una señal de radio desde un origen hasta su destino. OWLT es el rango o *range* en un contacto. En este contexto, el enrutamiento tiene en cuenta la variación en la distancia de tiempo de transmisión (delta de OWLT), que corresponde a la diferencia en el valor OWLT durante el tránsito de un *bundel* debido a la variación de la distancia entre emisor y receptor. Esta delta se conoce como OWLT *margin*, y se debe tener en cuenta durante el diseño del plan de contactos, ya que afecta al tiempo de recepción de los *bundels* en el receptor.

El proceso de enrutamiento se divide en tres fases (figura 11):

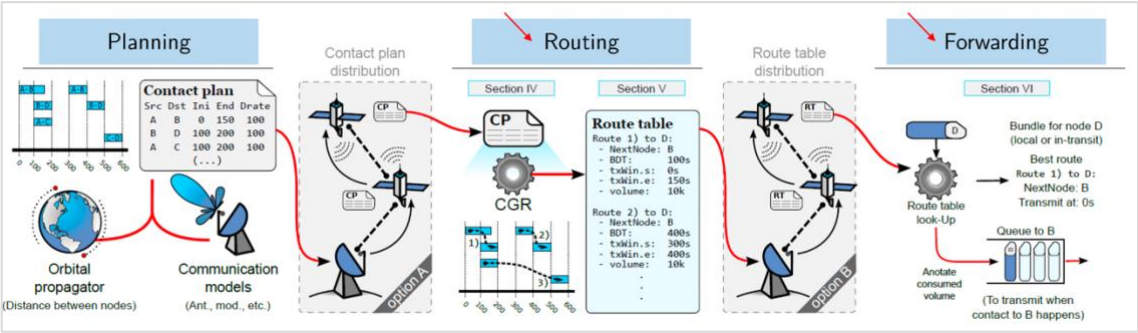
1. Planificación o *Planning*: en una red con contactos deterministas, esta fase consiste en elaborar un Plan de contactos o *Contact Plan*.
2. Enrutameinto o Routing: en el caso que el *Contact Plan* se distribuya entre los nodos, los algoritmos CGR se procesarán en dichos nodos para generar la Tabla de Rutas o

[29] Un protocolo epidémico es aquel donde los nodos de la red transmiten la copia de un paquete de información a otro nodo cuando sucede un contacto con el mismo, inundando así la red con copias del paquete hasta que la informaicón llega a su destino. Normalmente, se utiliza este tipo de protocolos de enrutamiento cuando los contactos son oportunistas. Se introducen variantes de protocolos epidémicos en redes con contactos probabilísticos.
[30] https://mars.nasa.gov/mgs/faqs/faq_mplan.html

Route Table. En el caso que el *Contact Plan* permanezca en el control de misión, se procesará con los algoritmos CGR de forma centralizada para generar la *Route Table,* que será distribuida entre los nodos.

3. Reenvío o *Forwarding*: este proceso selecciona la mejor ruta de la Route Table teniendo en cuenta el tamaño y prioridad de los datos y las condiciones de los contactos en el momento de la transmisión.

Figura 11: Proceso de enrutamiento.



(fuente: A Contact Graph Routing tutorial. Juan A. Fraire).

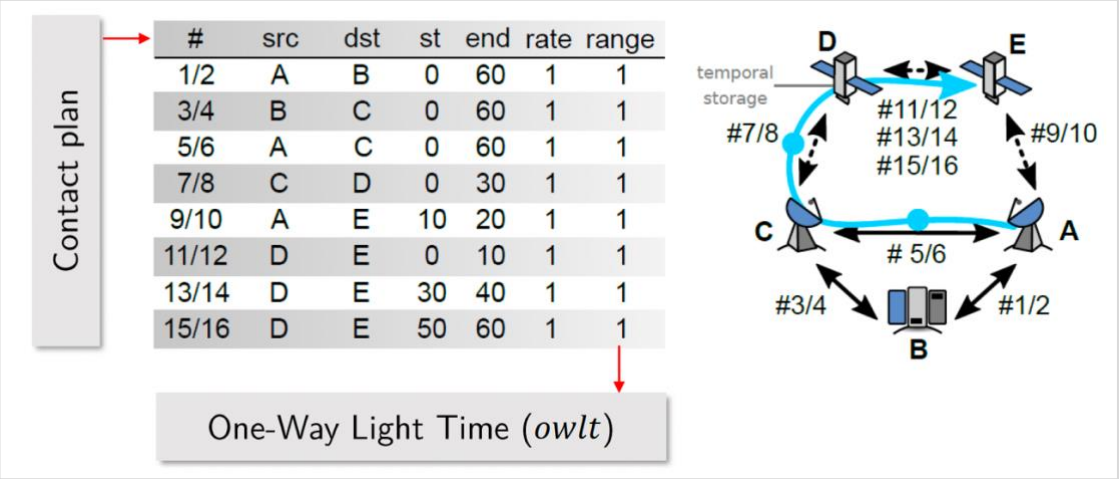
En el apartado 2.2.3.6. se define un contacto y la ecuación que indica el tiempo de transmisión en un contacto. En esta sección se define un contacto y una ruta como entidades de un grafo.

Un contacto $C_{A,B}^{t1,t2}$ es un intervalo de tiempo que transcurre de t_1 a t_2, durante el cual es posible la transmisión de datos entre los nodos A y B con un ratio de transmisión R. Como se puede ver en la figura 12, el plan de contactos se construye con el número de contacto (#)[31], nodo de origen (src), nodo destino (dst), tiempo de inicio del contacto o *start* (st), tiempo en de fin de contacto o *end* (end) el ratio de transmisión (rate) y el rango (range) correspondiente a OWLT. Cabe recordar el volumen del contacto (ecuación 2).

$$Volume = (end - st) * rate \qquad (2)$$

[31] El número de contacto tiene el formato n/n+1 cuando el enlace al que hace referencia entre dos nodos es bidireccional. En el caso de un enlace unidireccional únicamente se representará con un sólo número de contacto n.

Figura 12: Plan de contactos.



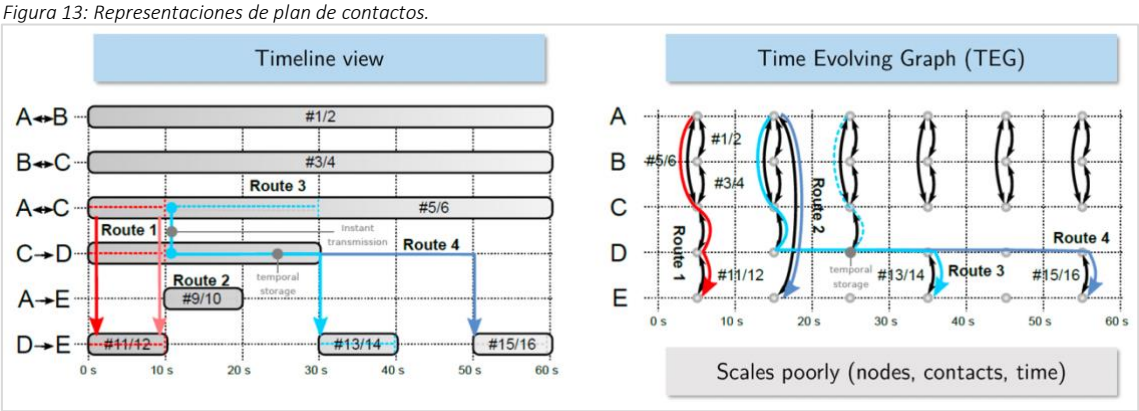
Contact plan

#	src	dst	st	end	rate	range
1/2	A	B	0	60	1	1
3/4	B	C	0	60	1	1
5/6	A	C	0	60	1	1
7/8	C	D	0	30	1	1
9/10	A	E	10	20	1	1
11/12	D	E	0	10	1	1
13/14	D	E	30	40	1	1
15/16	D	E	50	60	1	1

(Fuente: IEEE ComSoc Argentina: Enrutando la Internet espacial con CGR. Fraire).

Los planes de contacto pueden representarse de diferentes formas. En la figura 13 hay un ejemplo de una vista cronológica o *timeline view*, en cuyo caso se puede ver con claridad las ventanas de transmisión que se producen por los contactos en el tiempo. La segunda representación de la figura es un gráfico de tiempo de evolución o *Time Evolving Graph* (TEG), donde mejora la representación de las rutas. Estas dos representaciones no son escalables a medida que aumentan los nodos y contactos. Por ese motivo, el modelo utilizado es el Grafo de Contactos o *Contact Graph*.

Figura 13: Representaciones de plan de contactos.

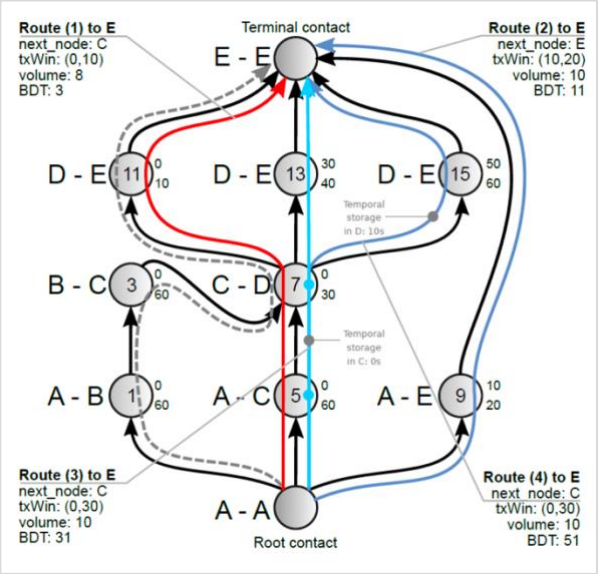


(Fuente: IEEE ComSoc Argentina: Enrutando la Internet espacial con CGR. Juan A. Fraire).

Un Grafo de Contacto (CG) de un origen S a un destino D, es un grafo acíclico directivo o *directed acyclic graph* (DAG)[32] (ecuación 5) donde los vértices V corresponden a contactos y las aristas o *edges* E pueden ser episodios de contacto y retención de información en un nodo (figura 14).

$$CG_S^D = (V, E) \tag{3}$$

Figura 14: Grafo de Contacto.



(Fuente: IEEE ComSoc Argentina: Enrutando la Internet espacial con CGR. Juan A. Fraire).

Un Grafo de Contacto no representa la topología de la red. Por ese motivo lo convierte en una herramienta poco intuitiva, pero útil para aplicar algoritmos de enrutamiento. Los vértices no son satélites o estaciones de tierra y las aristas no son rutas; representan intervalos de almacenamiento de información en un nodo hasta que el siguiente contacto (vértice V) está disponible. Cada par origen-destino produce una estructura distinta de Grafo de Contacto.

Un ruta R_A^B es la secuencia de contactos entre un origen S y un destino D (ecuación 5). Por ejemplo, en la figura 15 se obtienen las rutas óptimas de entre algunas de las rutas posibles.

$$R_S^D = \left\{ C_{n_{src,dest}}^{t_{st},t_{end}} \right\} \tag{4}$$

[32] Un DAG es un digrafo o grafo dirigido donde cada arista tiene un orden asignado en los extremos y el orden se indica con una flecha. Un grafo a → b, donde a y b son vértices, la arista → es {a, b} en el orden específico. Además, un DAG es acíclico lo que insica que no tiene un recorido circular, y tiene una topología ordenada por las direcciónes de las aristas.

Figura 15: Cálculo de rutas.

- Route (1):
  - $R_A^E = \{C_{A,C}^{0,60}, C_{C,D}^{0,30}, C_{D,E}^{0,10}\}$
  - Best delivery time $BDT = 3$
- Route (3):
  - $R_A^E = \{C_{A,C}^{0,60}, C_{C,D}^{0,30}, C_{D,E}^{30,40}\}$
  - Storage at node D until $t = 30$
  - Best delivery time $BDT = 31$

(Fuente: IEEE ComSoc Argentina: Enrutando la Internet espacial con CGR. Juan A. Fraire).

Determinar una ruta consiste en calcular el mejor tiempo de entrega o *Best Delivery* Time (BDT), teniendo en cuenta los parámetros cada contacto. A partir de estos grafos es posible utilizar algoritmos de búsqueda como **Dijkstra**[33] que permite calcular rutas de un Grafo de Contacto. A partir de este resultado el **algoritmo de Yen**[34] permite analizar las características de los contactos para encontrar las mejores rutas.

2.3. Implementaciones

2.3.1. Adaptación del estándar a la implementación

Las implementaciones deben ajustarse a las definiciones de arquitectura y las especificaciones de formatos en los protocolos. Aunque una implementación debe cumplir unas pautas predefinidas con motivos de compatibilidad, siempre puede haber variaciones entre implementaciones. Estas variaciones están motivadas por los márgenes que quedan, cuando algún detalle no queda definido de forma explícita, o cuando una arquitectura deja abiertas opciones para que en la práctica se formule un diseño concreto. Es decir, puede darse el caso que una arquitectura o protocolo deje un camino para desarrollar uno de los puntos clave, pero sin indicaciones explícitas. Estos casos son más evidentes en el caso que la tecnología esté en proceso de investigación o discusión de la formalización de un estándar. Serán los actores que exploran estas definiciones y las llevan a la práctica, los que concretarán qué forma específica tendrán las implementaciones.

Por ejemplo, la arquitectura DTN define los EID y un esquema URI; en ION se utilizan dos tipos de identificadores que cumplen con lo descrito en la arquitectura, y posteriormente en la propuesta de estándar los esquemas “ipn” y “dtn” quedan definidos de forma más concreta. En

[33] Dijkstra es un algorimo que permite obtener la distancia entre dos vértices en un grafo y se utiliza para obtener rutas.
[34] Yen’s algorithm es un algoritmo de comparación de rutas utilizado para obtener las rutas más óptimas.

el enrutamiento ocurre algo similar, y se convierte en un campo de investigación donde existen muchas posibilidades. También en la organización de las jerarquías para la topología de red así como en otros detalles.

En resumen, se debe tener en cuenta cuando se observa un estándar o definición de una tecnología, que algunos puntos pueden permanecer abiertos para acabar precisados en la aplicación práctica o en revisiones posteriores. Serán los usuarios y los participantes en completar y utilizar las tecnologías en cuestión, los que darán la razón a qué implementación es más adecuada para satisfacer las necesidades de los entornos reales.

En los siguientes apartados se describen algunas de las implementaciones existentes. Todas las implementaciones descritas son de código abierto.

2.3.2. Implementaciones para la versión 6 de *Bundle Protocol*

DTN2 es la implementación desarrollada en C y C++ de referencia basada en la RFC5050 [39] en la versión 6 del protocolo, y puede operar en Linux, Mac, Windows y FreeBSD. Puede funcionar con diferentes CLs como TCP, Ethernet, Bluetooth y serie. También trabaja con diferentes protocolos de enrutamiento como PRoPHET, *epidemic* y Bonjour y enrutamiento con rutas estáticas. DTN2 no ha sido actualizada a la versión 7.

Para esta misma versión del protocolo BP, la implementación IBR-DTN escrita en C++ está diseñada para sistemas embebidos. Es una implementación portable a múltiples arquitecturas. Puede utilizar TCP, UDP, HTTP y IEEE 801.15.4[35] como CLs, así como enrutar con rutas estáticas y protocolos de enrutamiento PRoPHET y *epidemic* entre otros. IBR-DTN no ha sido actualizada a la versión 7.

[35] IEEE 801.15.4 es un estándar que define el nivel físico y control de acceso al medio de redes inalámbricas de área personal o con bajas tasas de transmisión de datos o *low-rate wireless personal area network* (LR-WPAN). (Fuente: https://es.wikipedia.org/wiki/IEEE_802.15.4)

2.3.3. Implementaciones para la versión 7 de *Bundle Protocol*

Las implementaciones valoradas para el banco de pruebas y simulación han sido seleccionadas con la premisa de estar adaptadas a la versión 7 de *Bundle Protocol*. También se ha valorado su compatibilidad. Las dos implementaciones más destacadas son *Interplanetary Overlay Network* (ION) y Micro Planetary Communication Network (μPCN)[36] [40], en adelante μD3TN. En las siguientes secciones se comentan tres implementaciones para BP7.

2.3.3.1. ION

Interplanetary Overlay Network (ION) es una implementación propuesta por investigadores de JPL en 2007, desarrollada en C y que sigue en constante crecimiento. Está orientada a su uso en entornos espaciales con restricciones de recursos -procesador, almacenamiento, etc- y con las características de enlaces espaciales, como tasas de transmisión asimétricas, interrupciones de transmisión y amplios retardos. Incluye TCPCL, UDPCL y LTPCL como principales CLs. Cabe destacar el uso de CGR como protocolo de enrutamiento, y el uso del algoritmo *Yen's Algorithm* a partir de la versión 3.7 del software. Ha sido probado en sistemas operativos en tiempo real[37] como VxWorks[38] y RTEMS[39], así como en Linux y Windows.

ION presenta las siguientes características destacadas:

- Memoria dinámica privada: al arranque, el sistema asigna una memoria física fija a ION. El software gestiona esta memoria de forma dinámica internamente. Esta técnica de memoria compartida o *shared* memory, se utiliza para que el software no solicite memoria al sistema más allá de la disponible inicialmente y así evitar errores. Está diseñado para cumplir con las reglas del software de control de vuelo.
- *Zero-copy objects*. la información se almacena en objetos referenciados a los diferentes módulos del software. De esta forma se consiguen dos objetivos: mejorar eficiencia en el procesamiento de la información, y evitar la duplicidad de la información.
- Distribución de procesos: la arquitectura de ION permite el proceso en diferentes tareas las cuales están distribuidas de manera eficiente con comunicaciones *inter-task* basadas en la memoria compartida.

[36] Hasta la versión 0.8.0. de octubre de 2020 bajo el nombre *Himalayas,* publicada con el nombre μPCN en https://upcn.eu. En adelante, según el repositorio del desarrollador (D3TN https://d3tn.com) en https://gitlab.com se publica con el nombre μD3TN.
[37] Sistema operativo en tiempo real o Real-time Operating System (RTOS) es un sistema operativo diseñado para hacer tareas concretas con una restricciones de tiempo y planificación ordenada y priorizada, de forma que las operaciones son predecibles y determinadas en el tiempo.
[38] https://www.windriver.com/products/vxworks
[39] https://www.rtems.org

2.3.3.2. μD3TN

μD3TN es una implementación desarrollada en C y diseñada para su uso en microcontroladores y sistemas POSIX40. Los CLs usados son MTCP y TCPCL. Puede compilarse en sistemas Linux y FreeRTOS. Utiliza un enrutamiento basado en la confiabiliad de los nodos vecinos.

El desarrollador enfoca el software a diferentes áreas de aplicación, destacando los CubeSat y las redes de sensores.

2.3.3.3. DTN7

DTN7 [41] es una implementación escrita en Go, con un diseño modular y una API simple e independiente al lenguaje de programación para desarrollar aplicaciones. La CL utilizada es MTCP. El enrutamiento se basa en rutas estáticas con descubrimiento de nodos vecinos. DTN7 tiene una versión programada en Rust.

2.4. Ejemplos y beneficios de uso de DTN

Durante los últimos años se han realizado diferentes aplicaciones prácticas en escenarios reales para comprobar y analizar la funcionalidad de la red DTN. Seguidamente se listan algunos de los despliegues y pruebas realizados:

2008: Constelación de satélites *UK's Disaster Monitoring Constellation*[41]. Primera red de satélites con aplicación de tecnología DTN.
2008: JPL realiza el experimento llamado *Deep Impact Network Experiment* (DINET) utilizando la nave del proyecto EPOXI[42] para realizar una simulación de una red Tierra-Marte.
2009: Implementación de DTN en la ISS [42][43].
2011: Artic Village Case [44]. Proyecto de red DTN.
2013: Demostración de transmisión DTN desde la misión LADEE[43] utilizando la transmisión laser LLCD[44] [45].

---

[40] *Portable Operating System Interface* UniX (POSIX) es un estándar de interfaz para sistema operativo, cuyo objetivo es facilitar la portabilidad del software entre diferentes sistemas operativos. https://ieeexplore.ieee.org/document/8277153
[41] https://directory.eoportal.org/web/eoportal/satellite-missions/d/dmc
[42] EPOXI fué una misión que transcurrió entre 2005 y 2013 cuyo objetivo era expolorar un cometa para examinar su estructura y composición.
[43] *Lunar Atmosphere and Dust Environment Explorer* (LADEE) fue una misión para el estudio de la atmósfera lunar y el polvo en suspensión de la Luna.
[44] *Lunar Laser Communication Demonstration* (LLCD) fué una prueba experimental, que consistió en incorporar un terminal de comunicación laser en la nave de la misión LADEE para la transmisión de información a dos estaciones ópticas de tierra -Table mountain en California y Observatorio del Teide en España-, con el resultado de una tasas de transferencia de 622Mbps.

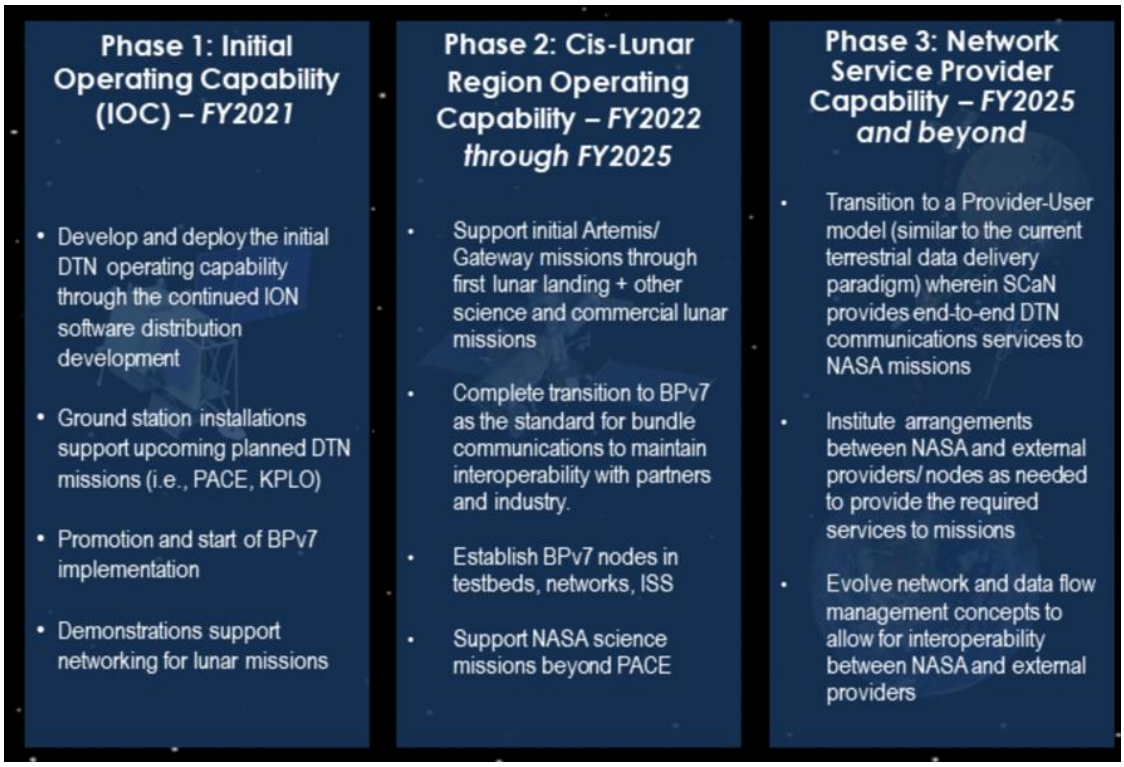
2017: Publicación de la arquitectura de red para la ISS [43].

2017: Desde la estación McMurdo en la Antártida se envía una fotografía a la ISS utilizando la red DTN para demostrar su funcionalidad.[45]

2019: La Agencia Espacial Europea lanzó a finales de 2019 OPS-SAT, un CubeSat con el que se utilizará DTN.

Además, está prevista la utilización de redes DTN para futuras misiones, destacando la arquitectura LunaNet[46] del programa Artemis de la NASA pretende ser la red de interconexión de la Luna. El programa SCaN[47] de la NASA tiene previsto incorporar DTN en su infraestructura de red (figura 16).

Figura 16: NASA SCaN network DTN infusion plan.



(Fuente: NASA).

Por otra parte, *Science Mission Directorate*[48] (SMD) de la NASA [46], después de analizar la aplicación de DTN en múltiples escenarios, identificó diferentes beneficios de su uso en misiones futuras (tabla 9), así como los temas pendientes para la implementación de DTN (tabla 10).

[45] https://www.nasa.gov/feature/antarctic-selfie-s-journey-to-space-via-disruption-tolerant-networking
[46] https://esc.gsfc.nasa.gov/projects/TEMPO?tab=lunanet
[47] Space Communications and Navigation (SCaN) es un programa de la NASA que opera sus redes de comunicaciones y cuyo objetivo es obtener un sistema de comunicaciones unificadas para sus misiones y redes en el espacio.
[48] *Science Mission Directorate* (SDM) es una organización cuya misión es crear un flujo de conocimiento interdisciplinar, es decir, hacer que los descubrimientos en una disciplina científica llegen a otras areas.

Tabla 9: Beneficios del uso de DTN en misiones espaciales.

Característica	Beneficio	Descripción
Estandarización	Reutilización	Los diseños de hardware y software, así como procedimientos de implementación pueden ser comunes para diferentes misiones.
	Interoperabilidad	Interfaz de comunicación común para diferentes misiones, incluyendo agencias e industria espacial.
Comunicaciones en red	Escalabilidad	DTN facilita añadir un nuevo, nodo, enlace o ruta a la red de comunicaciones.
	Autonomía	DTN permite la transmisión de datos mediante diferentes saltos en los nodos de la red.
	Aplicaciones desacopladas	Las aplicaciones son independientes a los enlaces y sistemas de comunicaciones, facilitando el desarrollo paralelo en ambas partes.
Enfoque *Store-and-Forward*	Retardos en la red	DTN puede mitigar los retardos de la red.
	Desconexiones de red	DTN puede mitigar las desconexiones de red.
	Desajustes en ratios de transmisión	DTN puede mitigar las asimetrías en los ratios de transmisión entre emisor y receptor.
	Eficiencia de uso de ancho de banda y tiempo de contacto	DTN es eficiente en el uso del ancho de banda y los tiempos disponibles para la transmisión de información.
	Simplificación de las implementaciones y operaciones de vuelo y de los sistema de tierra	DTN puede simplificar las soluciones necesarias para cumplir con los requisitos de las misiones.

(fuente: SDM).

Tabla 10: Cuestiones que deben abordarse para el uso de DTN en misiones espaciales.

Incidencias pendientes de estudio	Descripción
Seguridad	Afianzar la seguridad en DTN
Navegación	Impacto en el rendimiento y cómo mitigarlo
Sobrecarga de red (*overhead*)	Sobrecarga de la red por el uso de DTN
Incentivos	Incentivos para la adopción de DTN
Disponibilidad de proveedores	Proveedores de red DTN
Adhesión a los estándares	Reforzar la estandarización
Avances en TRL[49]	Qué se necesita para avanzar en la valoración TRL
Administración de red	Cómo se administrará la red DTN

(fuente: SDM).

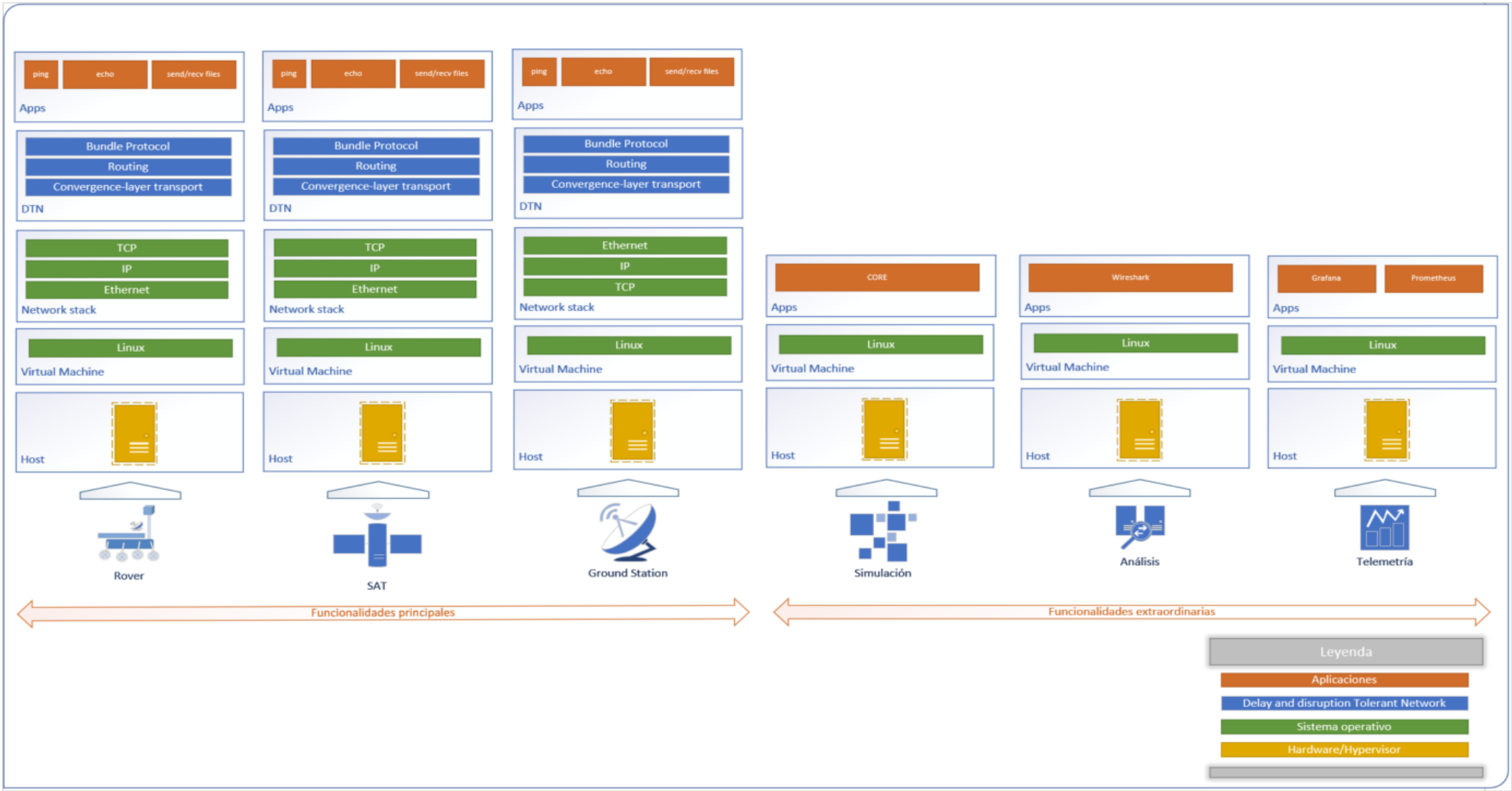
[49] Nivel de madurez tecnológica o *Technology Readiness Level* (TRL) es una escala de medida de 1 a 9 utilizada para valorar la madurez de una tecnología.
(https://www.esa.int/Enabling_Support/Space_Engineering_Technology/Shaping_the_Future/Technology_Readiness_Levels_TRL)
(https://www.nasa.gov/directorates/heo/scan/engineering/technology/technology_readiness_level)

3. Diseño del banco de pruebas y simulación

3.1. Requerimientos

Este apartado se describe las características del sistema utilizado para construir un entorno de pruebas. En el entorno de pruebas se modela un escenario propuesto y se estudian las configuraciones y funcionamiento de aplicaciones DTN. Por otra parte, se explica el emulador utilizado para continuar el escenario de pruebas con otro enfoque. Por último, se detallan las herramientas utilizadas para análisis.

3.2. Diagrama de bloques del sistema



3.3. Entorno de laboratorio

El escenario de pruebas y el sistema de simulación será modelado sobre un entorno de virtualización donde correrán las diferentes máquinas virtuales utilizadas en el proyecto.

La infraestructura está compuesta por los siguientes equipos:

Servidor:	Cisco UCSC-C220-M5SX
Hostname:	HostB129
Procesador:	2 procesadores Intel(R) Xeon(R) Bronze 3106 CPU 1.70GHz (16 cores)
Memoria:	256GB
Red:	1 puerto de 1Gbps – 4 puertos de 10Gbps
IP de gestión HW:	10.101.100.129/24 (1Gbps)
IP de gestión ESXi:	10.101.111.129/24 (10Gbps)
IP 1 NFS[50]v4:	10.101.109.129/24 (10Gbps)
IP 2 NFS v4:	10.101.119.129/24 (10Gbps)
Red VMs:	Trunk (10Gbps)
Switch 1:	2 switches Cisco 4500X (Cluster de 4 puertos 10Gbps)
Hostname:	SWB45KX1
Red:	36 puertos 1/10Gbps SFP/SFP+
IP de gestión:	10.101.102.115/24
Switch 2:	1 switch Cisco 3750G
Hostname:	SWB3750G2
Red:	24 puertos 10Gbps Ethernet – 4 puertos 1Gbps SFP
IP de gestión:	10.101.102.114/24
Almacenamiento 1:	Qnap TS-832XU-RP
Hostname:	NAS10Q20
Volumen asignado:	3TB
HD instalados:	8 discos Samsung SSD 860 EVO 1TB
HD caché:	2 discos M.2 Samsung SSD 970 EVO Plus 500GB
Red:	2 puertos 1Gbps – 2 puertos 10Gbps
IP de gestión:	10.101.103.150/24 (1 Gbps)
IP 1 NFSv4:	10.101.109.150/24 (10Gbps)
IP 2 NFSv4:	10.101.119.150/24 (10Gbps)
Almacenamiento 2:	Qnap TS-832XU-RP
Hostname:	NAS10Q21
Volumen asignado:	14TB
HD instalados:	8 discos Western Digital Red NAS SATA 4TB
HD caché:	2 discos M.2 Samsung SSD 970 EVO Plus 500GB
Red:	2 puertos 1Gbps – 2 puertos 10Gbps
IP de gestión:	10.101.103.151/24 (1Gbps)
IP 1 NFSv4:	10.101.109.151/24 (10Gbps)
IP 2 NFSv4:	10.101.119.151/24 (10Gbps)

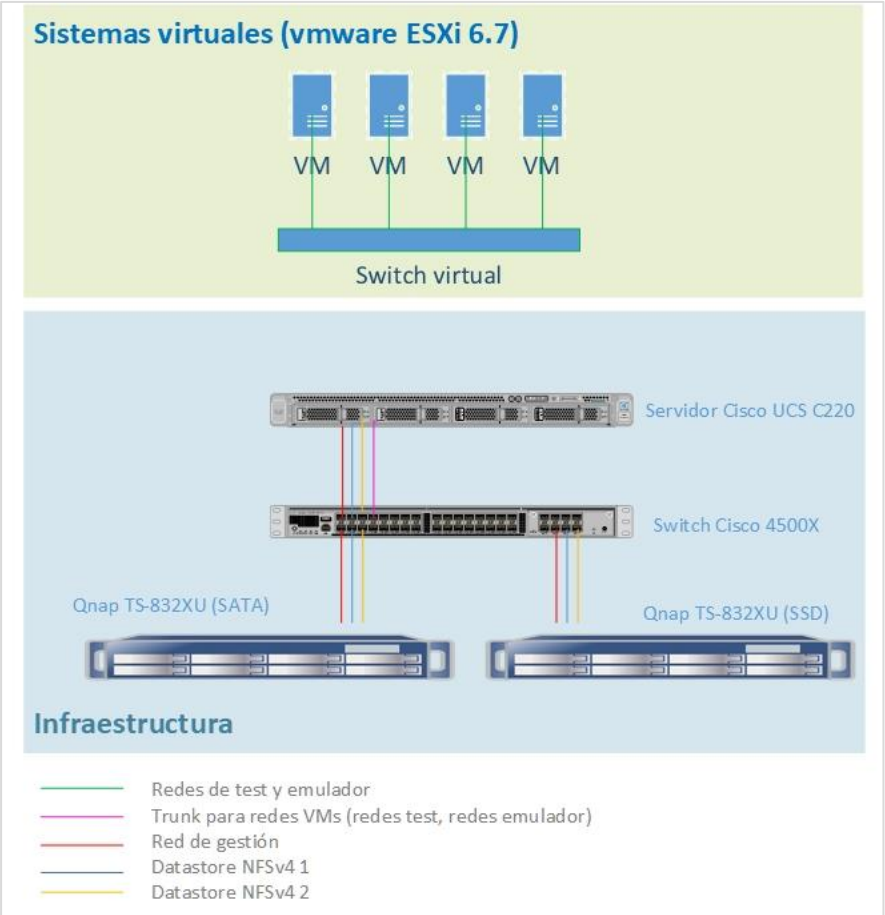
Las conexiones de red a 10Gbps se han realizado con adaptadores SFP+ de fibra óptica monomodo.

[50] Sistema de archivos de red o *Network File System* (NFS) es una aplicación cliente-servidor que permite compartir archivos en red. https://es.wikipedia.org/wiki/Network_File_System

Las unidades de almacenamiento (virtuales) de las VMs están ubicados en los sistemas de almacenamiento. Las unidades ubicadas en los almacenamientos son accesibles utilizando el protocolo NFSv4 mediante dos redes IP y dos enlaces físicos de 10Gbps, para ofrecer redundancia y aumentar la tasa transferencia con balanceo de tráfico de red. 'Almacenamiento1' tiene discos Serial ATA y se ha utilizado para VMs de gestión, plantillas y clones de máquinas virtuales. 'Almacenamiento2' tiene discos SSD y se ha utilizado para VMs del banco de pruebas y simulación.

El sistema hipervisor[51] utilizado es ESXi de Vmware, Inc., en la versión 6.7 U3.
El sistema utilizado se puede ver en la figura 17. La tabla 11 muestra la relación de VLANs utilizadas .

Figura 17: Sistema de virtualización para banco de pruebas y emulador.



[51] Un hipervisor o *hypervisor* es una capa de software que permite la abstracción de los recursos de hardware para correr diferentes máquinas virtuales sobre la misma máquina física. https://es.wikipedia.org/wiki/Hipervisor

Tabla 11: VLANs para la red del entorno de pruebas.

VLAN ID	Nombre VLAN	Descripción	Red IP
120	MGMTHW	Red de gestión de hardware del servidor.	10.101.100.0/24
121	MGMTHY	Red de gestión del hipervisor.	10.101.111.0/24
122	MGMTSW	Red de gestión de electrónica de red.	10.101.102.0/24
123	MGMTST	Red de gestión de almacenamiento.	10.101.103.0/24
129	NFSV41	Red 1 de almacenamiento NFS v4.	10.101.109.0/24
139	NFSV42	Red 2 de almacenamiento NFS v4.	10.101.119.0/24
240	MGMTVM	Red de gestión de VMs entorno y emulador	10.20.140.0/24
241	Net-Odyssey-EarthDSN	Red del enlace orbitador Odyssey y DSN	10.20.141.0/24
242	Net-MRO-EarthDSN	Red del enlace orbitador MRO y DSN	10.20.142.0/24
243	Net-Odyssey-MSL	Red del enlace orbitador Odyssey y Curiosity	10.20.143.0/24
244	Net-MRO-MSL	Red del enlace orbitador MRO y Curiosity	10.20.144.0/24
245	Net-DSNMars-MCC	Red del enlace DSN Mars y Mission Control Center	10.20.145.0/24
246	Net-DSNMars-DSNMoon	Red de enlace entorno de pruebas y emulador	10.20.146.0/24

El ‘switch 1’ opera en capa 3 y tiene una IP en todas las redes de gestión de la topología. En este caso, hará funciones de enrutador IP para todas las redes de gestión, de la VLAN 120 a la VLAN 240. El número de host en la IP en cada una de las redes del switch 1 es 200. Por ejemplo, 10.101.109.200 para la VLAN NFSV41, 10.101.119.200 para la VLAN NFSV42, etc.

3.4. Herramientas para el banco de pruebas

Una vez instaladas las máquinas virtuales que forman parte del banco de pruebas y que se detalla en el capítulo 4, las herramientas utilizadas para modelar el entorno son:

Traffic Control (tc) y **netem**[52]: utilizado para modelar los canales de comunicación de cada uno de los enlaces de red. (Anexo 3).

iperf[53]: herramienta de medida de la tasa de transferencia utilizado para comprobar las tasas de transferencia de los enlaces modelados con tc y netem en la red IP. (Anexo 3).

[52] https://man7.org/linux/man-pages/man8/tc-netem.8.html
[53] https://iperf.fr

DTNperf[54]

Herramienta utilizada para analizar el rendimiento en la red DTN. [47][48] (Anexo 7).

3.5. Herramientas para la simulación

El emulador utilizado es **Common Open Research Emulator** (CORE[55]). CORE es un software de emulación de redes para una o múltiples maquinas, desarrollado por Boeing Research and Technology Division.

CORE está desarrollado en Python. Los hosts que forman parte de una emulación son contenedores[56] de Linux LXC. Los nodos son accesibles mediante una sesión SSH desde el host. Emula características de red utilizando tc y netem, aunque en su lugar es posible integrarlo con un software para este propósito llamado Extendable Mobile Ad-hoc Network Emulator (EMANE), utilizado para detallar con más precisión características de enlaces de datos. Además, se integra con otras herramientas, como el enrutador Quagga[57] y el analizador de paquetes de red Wireshark[58].

Las emulaciones en CORE consisten en caracterizar los dispositivos y los enlaces de una red, incluyendo el desplazamiento de dichos dispositivos. Los desplazamientos de estos dispositivos se describen en un archivo que contiene el plan movilidad, y donde se detallan las posiciones de origen y destino, los intervalos donde se producen los movimientos y la velocidad de movimiento.

Se ha utilizado un kit de desarrollo de la NASA distribuido por MITRE Corporation[59], que consiste en una VM Ubuntu con el emulador CORE y herramientas. El software ION incluido ha sido actualizado para trabajar con la versión BP7. El escenario se ha basado en emulaciones incluidas en el kit y descritas en el curso ION de la NASA de 2020[60].

---

[54] https://gitlab.com/dtnsuite/dtnperf/tree/master/
[55] https://github.com/coreemu/core
[56] Un contenedor en el contexto de virtualización, es una instancia de un conjunto de procesos y redes en un mismo espacio de usuario o *namespace*. Esta tecnología permite instanciar diferentes aplicaciones encapsuladas sobre un único sistema operativo. Es diferente a la virtualización mediante un hipervisor, el cual instancia sistemas operativos completos.
[57] https://quagga.net
[58] https://www.wireshark.org
[59] https://www.mitre.org/download-nasas-dtn-development-kit
[60] https://www.nasa.gov/directorates/heo/scan/engineering/technology/disruption_tolerant_networking_software_options_ion

3.6. Herramientas de análisis y telemetría

Las herramientas descritas en esta sección se han utilizado como herramientas de apoyo a la instalación y observar el sistema sobre el que se prueba la red DTN.

Graylog[61]: servidor utilizado para recopilar logs de las máquinas virutales del entorno de pruebas. No se han enviado logs de los nodos de la emulación. (Anexo 5).

Telegraf[62]: software utilizado para recopilar información de un sistema y enviarlo a una base de datos como InfluxDB. (Anexo 6).

InfluxDB[63]: base de datos de serie temporal o *Time Serie Database* (TSD), es una base de datos optimizada para almacenar información relacionada con la variación de valores de una entidad en función del tiempo. Es útil para tener un registro del cambio de un valor de medida de un recurso en un sistema de información. (Anexo 6).

3.7. Topología de red Interplanetaria propuesto

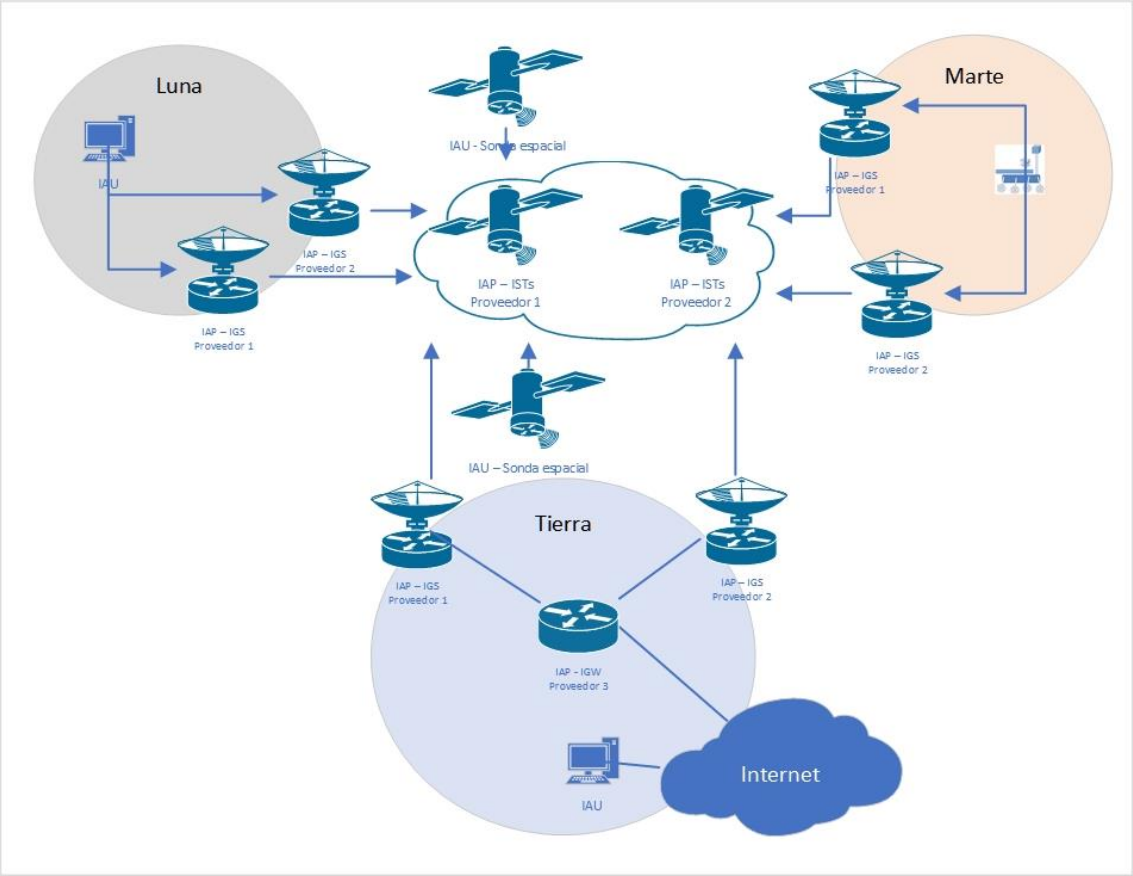
De forma conceptual, podemos reducir Internet a un entorno donde existen diferentes roles de participación. El rol fundamental corresponde -en una connotación clásica, pero fácil de comprender- a proveedores de servicio de Internet o *Internet Service Providers* (ISP), cuyo cometido es dotar de infraestructura de acceso a la red a los usuarios e interconectarlos, ya sea mediante su propia red o reenviando los datos a través de la red de otro(s) ISP(s). Como segundo rol están los usuarios, que podríamos a su vez dividirlos en dos grupos: el primero formado por usuarios -como profesionales, empresas e instituciones- que ofrecen diferentes tipos de servicios a otros usuarios utilizando Internet como un mercado o medio. Además de un segundo grupo que corresponde a usuarios finales, los cuales previo acceder a la red mediante un ISP, utilizan los servicios de los usuarios profesionales del primer grupo del rol de usuarios. En la práctica, los roles de los participantes en Internet son más numerosos y muchas veces, se solapan entre ellos. Aunque este concepto es una simplificación, todavía es válido y sirvió en su momento de punto de partida. En esta propuesta se utiliza como modelo.

El escenario propuesto, está orientado a trasladar dichos roles a la red interplanetaria. Endentemos dicha red como una extensión de Internet en el Sistema Solar. En este sentido definimos igualmente dos roles (figura 18):

[61] https://www.graylog.org
[62] https://www.influxdata.com/time-series-platform/telegraf/
[63] https://www.influxdata.com/products/influxdb/

1. Proveedores de Acceso IPN o *IPN Access Providers* (IAP). Son entidades privadas o gubernamentales cuya misión principal es ofrecer servicios de acceso a la red interplanetaria mediante un mecanismo de conexión. Este mecanismo de conexión lo entendemos como el sistema de comunicación necesario, para que el nodo o dispositivo del cliente que contrata el servicio de acceso pueda acceder a la red interplanetaria.
2. Usuarios de acceso IPN o IPN Access Users (IAU). Son usuarios divididos en dos grupos:
    a. Usuarios que ofrecen servicios a otros usuarios.
    b. Usuarios que utilizan la red IPN para acceder a los servicios de otros usuarios o para transmitir información privada.

Figura 18: Modelo IPN-IAU.



De esta forma, es viable que en los proyectos de misiones espaciales, las comunicaciones puedan tratarse como una serie de requerimientos. Una vez precisadas las necesidades de comunicaciones de la misión, estas podrían proveerse por los servicios de un IAP adecuado a los requisitos.

En los últimos años, las misiones espaciales están trabajando para llevar al marco de un mercado privado no gubernamental, la tarea de lanzar cohetes al espacio, con la intención de concentrarse en la carga útil consecuencia de una misión. El objetivo, además de crear un mercado privado que potencie la industria espacial, tiene que ver con simplificar los proyectos. Es decir, es mucho más sencillo diseñar, planificar, ejecutar y operar un proyecto, si contratamos a terceras partes los elementos que son necesarios pero que no forman parte del propósito de la misión. Como analogía, en el escenario de Internet, hoy es impensable para la mayoría de los particulares o instituciones públicas y privadas, plantear un proyecto donde es necesario trabajar con dos sedes separadas en ciudades distantes y ocuparse del enlace de comunicaciones. Habitualmente, resulta mucho más viable y productivo, contratar los servicios a un proveedor de servicios.

El rol IAP contendrá tres tipos, y el rol IAU supondrá un tipo único tal y como se detalla seguidamente y se resume en la tabla 12:

1. Los IAPs pueden ser de tres tipos, siempre teniendo en cuenta un rol de proveedor de servicios de acceso IPN:

a. Satélites IPN o *IPN Satellite* (IST): satélites de acceso IPN.
b. Estación de comunicaciones IPN de Tierra, o IPN Ground Station (IGS): estaciones de acceso IPN en la superficie de un planeta o satélite natural cuya función es el acceso directo desde cualquier ubicación con alcance[64].
c. Gateway IPN o IPN Gateway (IGW): IST o IGS con funcionalidad para hacer de puente entre las redes DTN y otras redes como Internet. En este proyecto se considerará que todos los ISTs e IGSs tienen o pueden tener funciones IGW, por lo que únicamente se utilizarán los tipos IST e IGS.

2. Los IAUs se consideran de un único tipo. Cualquier elemento de misión puede ser considerado un IAU. Por ejemplo, un orbitador o un rover de una misión en Marte, puede ser un elemento que utilice un IAP de un proveedor mediante un satélite en dicho planeta, para transmitir sus datos a través de la red del proveedor de servicios, hasta su Control de Misión, el cual a su vez, está conectado a un IAP en la Tierra que puede ser una Estación de Tierra de este u otro proveedor.

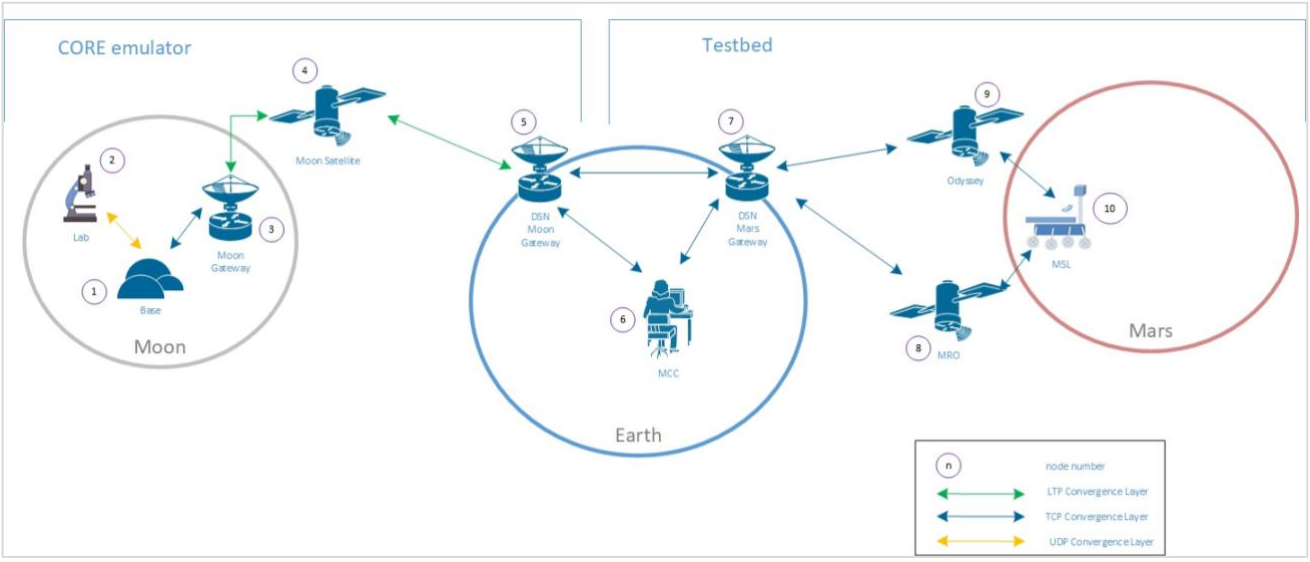
[64] Debido a la complejidad de las instalaciones de los sistemas de comunicaciones para el espacio profundo en la Tierra -denominados IGS en este proyecto-, un futuro escenario donde exista un uso generalizado de constelaciones de ISTs podría contrastar con el uso de comunicaciones directas mediante IGSs. Quedando los IGSs como puntos de conexión para los ISTs con una funcion principal IGW, es decir, como puentes entre Internet y/o las redes en superficie y la red IPN.

Tabla 12: Roles y tipos de elementos de la red IPN.

Rol	Tipos	Descripción
IAP	IST	IPN Satelite, orbitador
(IPN *Access Provider*)	IGS	IPN *Ground Station*
	IGW (IST-IGW, IGS-IGW)	IPN Gateway para IST o IGS
IAU	-	Satélite, orbitador, rover, etc.
(IPN *Access User*)		

El escenario propuesto se basará en tres cuerpos celestes de nuestro Sistema Solar: Tierra, Luna y Marte (figura 19).

Figura 19: Sistema de pruebas y emulador para red Luna-Tierra-Marte.



Este escenario tiene sentido en un contexto en el cual la tecnología de acceso entre estaciones y/o dispositivos de superficie siempre es inalámbrico. Contando con una constelación de ISTs en cada cuerpo celeste, suficiente en número y capacidad para permitir la comunicación interior como IAP -actuando los satélites como enrutadores entre dispositivos que se transmiten información en la misma superficie utilizando DTN- y comunicación exterior -actuando los satélites como IGW y reenviadores de datos a los planetas o satélites naturales vecinos con destino a equipos de la superficie de dicho cuerpo vecino. El escenario contempla un único proveedor IAP para cada uno de los IAUs.

En este escenario no se contemplan las comunicaciones de misiones en ruta interplanetaria. De la misma forma, no se consideran las comunicaciones entre satélites o *Inter-Satellite Link* (ISL[65]).

[65] Los enlaces entre satélites o *Inter-satellite link* (ISL) es la capacidad de comunicación y transmisión información entre satélites. Aunque es un concepto necesario para IPN no se contempla en los escenarios de este trabajo.

4. Escenario de red IPN 1: banco de pruebas

La implementación utilizada debe operar con la versión BP7 y debe estar lo más actualizada posible. Con esta premisa la opción más valorada ha sido ION en su versión 4.0.2.

4.1. Escenario para el sistema Tierra-Marte

4.1.1. Nodos de la red en el sistema Tierra-Marte

En este apartado se utilizan datos de trabajos y publicaciones relacionados con estudios de redes DTN en las comunicaciones con Marte [49][50][51][52].

De las redes operadas por la ESA y la NASA, para el escenario propuesto se considera la red para el espacio profundo o *Deep Space Network* (DSN) y sus antenas DSS [66] de la NASA, como estación de tierra o Ground Station (GS). DSN cuenta con tres complejos de antenas que servirán de *gateway* entre Internet y la red IPN:

1. Canberra (Australia): antena DSS35.
2. Madrid (España): antena DSS55.
3. Goldstone (US): antena DSS25.

El Centro de Control de Misión o *Mission Control Center* (MCC) son las instalaciones desde donde se realizan todos los controles y seguimientos de los sistemas de la misión. En el escenario, MCC se considera conectado a GS mediante red IP.

En Marte, se considera el uso de dos orbitadores:

1. Mars Odyssey: Misión iniciada en 2001 y de órbita circular con una altitud de 400km.
2. Mars Reconnaissance Orbiter (MRO): Misión iniciada en 2005 y de órbita circular con una altitud de 300km.

En escenario de pruebas se incluye en la superficie de Marte el rover Curiosity de la misión Mars Science Laboratory (MLS). El rover Curiosity puede transmitir datos directamente a la Tierra, pero en este escenario se considera únicamente la transmisión mediante los orbitadores, siguiendo la topología propuesta.

[66] Segimiento de operaciones DSN: https://eyes.nasa.gov/dsn/dsn.html

El objetivo de este escenario es que el rover Curiosity MLS se comunique con el Centro de Control de Misión.

4.1.2. Enlaces entre los nodos de la red

La distancia mínima entre la Tierra y Marte considerada es de 0,72 AU[67] (107 millones de km) que supone un OWLT de 360 segundos. Este tiempo de transmisión se considera en los enlaces entre la antenas de DSN y los orbitadores de Marte. Se estima que las tasas de transmisión del rover a los dos orbitadores son de diferentes capacidad y asimétricas. Los distintos valores de los enlaces se pueden apreciar en la tabla 13. Estos valores están basados en las publicaciones indicadas al inicio de la sección 4.2.

Tabla 13: valores de los enlaces entre los nodos del escenario propuesto para el sistema Tierra-Marte.

Nodo 1	**Nodo 2**	**Distancia media [AU]**	**OWLT [s]**	**Uplink data rate N1 → N2 [kbps]**	**Downlink data rate N2 → N1 [kbps]**	**BER (%)**
MSL	MRO	-	1	128	32	0,5
MSL	Odyssey	-	1	64	16	0,5
MRO	DSN	0,72	360	500	128	3
Odyssey	DSN	0,72	360	128	16	3
DSN	MCC	-	0,1	10000	10000	0

4.2. Plan de Contactos

Las comunicaciones están planificadas en función de las dinámicas de las órbitas y los planes de operaciones [53][54]. Las transmisiones de datos desde orbitadores y satélites se realizan de forma planificada. Conocida su órbita, se establecen intervalos de tiempo en los cuales se pueden enviar datos[68].

En la red de antenas de Madrid y Canberra se puede consultar el tiempo planificado para el uso de antenas para cada misión. Por ejemplo, como se puede ver en la tabla 14, existe una franja

[67] Unidad Astronómica o Astronomical Unit (AU) es una unidad de medida utilizada en el Sistema Solar. 1 AU = 149.597.870 km. Corresponde a la distancia entre la Tierra y el Sol.

[68] El tiempo en el cual un orbitador no puede comunicarse con una estación de tierra se llama pérdida de señal o *Loss Of Signal* (LOS). El tiempo en el cual un orbitador puede comunicarse con una estación de tierra se llama adquisición de señal o *Acquisition Of Signal* (AOS). Durante el intervalo LOS, los datos se almacenan en un registro por corte de comunicación o *Communication Outage Recorder* (COR) para su posterior envío cuando sea posible. https://darts.isas.jaxa.jp/iss/kibo/files/reference/aos_los.pdf

de tiempo muy amplia para atender a las transmisiones de MRO y Odyssey cada 24 horas. Lo mismo ocurre con las antenas en las tres estaciones de tierra de la red DSN.

Tabla 14: Contactos DSN Madrid días 27 a 30 de abril.

Fecha	Hora inicio	Hora fin	Antena	Misión
27/Apr	08:20	15:35	DSS-56	Mars Reconnaisance Orbiter
27/Apr	09:20	16:35	DSS-55	Mars Odyseey Orbiter
27/Apr	14:30	22:50	DSS-54	Mars Reconnaisance Orbiter
27/Apr	15:30	17:55	DSS-56	Mars Odyseey Orbiter
28/Apr	08:20	13:45	DSS-63	Mars Odyseey Orbiter
28/Apr	08:20	13:45	DSS-63	Mars Reconnaisance Orbiter
28/Apr	15:50	22:45	DSS-65	Mars Reconnaisance Orbiter
28/Apr	15:55	21:55	DSS-55	Mars Odyseey Orbiter
28/Apr	16:00	21:00	DSS-63	Mars Odyseey Orbiter
29/Apr	09:00	13:20	DSS-55	Mars Odyseey Orbiter
29/Apr	09:00	22:45	DSS-55	Mars Reconnaisance Orbiter
29/Apr	13:30	16:00	DSS-65	Mars Odyseey Orbiter
29/Apr	14:30	17:45	DSS-63	Mars Odyseey Orbiter
29/Apr	14:40	22:45	DSS-56	Mars Odyseey Orbiter
29/Apr	17:45	22:00	DSS-63	Mars Odyseey Orbiter
29/Apr	20:00	21:15	DSS-54	Mars Odyseey Orbiter
29/Apr	22:00	02:15	DSS-63	Mars Odyseey Orbiter
30/Apr	09:00	21:25	DSS-63	Mars Odyseey Orbiter
30/Apr	09:35	22:45	DSS-55	Mars Reconnaisance Orbiter
30/Apr	14:35	18:40	DSS-54	Mars Odyseey Orbiter
30/Apr	17:20	23:05	DSS-56	Mars Odyseey Orbiter

(fuente: https://www.mdscc.nasa.gov/index.php/sched-2/).

Por otra parte, se utiliza la herramienta WebGeocalc[69] (WGC). WGC es una herramienta de la *Navigation and Ancillary Information Facility* (NAIF) de la NASA. NAIF creó el sistema de información SPICE, que ofrece a científicos e ingenieros la posibilidad interpretar la información de instrumentos de observación de las naves robóticas. La base de datos SPICE contiene conjuntos de datos llamados *Kernels* los cuales hacen referencia a planetas, satélites, etc. WGC es una herramienta web que utiliza el sistema de información SPICE.

En el anexo 2 se sigue un proceso para la obtención de una lista de contactos.

4.3. Adaptación del escenario al banco de pruebas

La adaptación del escenario al banco de pruebas consiste en trasladar las características de la red propuesta al entorno disponible.

Se configura una VM para cada uno de los elementos del supuesto. La VM tendrá 1 core asignado, 1024MB de memoria RAM y 30GB de disco. El sistema instalado es Ubuntu Server 20.04 con el software ION en la versión 4.0.4 tal y como se ha comentado anteriormente.

[69] https://naif.jpl.nasa.gov/naif/webgeocalc.html

Los planes de contacto se reducen de 24 horas a 24 minutos de forma proporcional. La duración máxima de cada prueba será de 24 minutos. Se han utilizado tres planes de contactos diferentes. En primer lugar un plan de contactos respetando los tiempos originales adaptados de 24 horas a 24 minutos. Un segundo plan de contactos para pruebas de conectividad entre nodos y funcionamiento básico de los protocolos, que corresponde a 24 minutos, con los nodos del banco de pruebas para la red Tierra-Marte con contacto continuo. Y un tercer y último plan de contactos basado en el primer plan de contacto adaptado de 24 horas a 24 minutos, replicando los contactos MSL-MRO y MSL-Odyssey cada minuto y ampliando los contactos a un mínimo de 20 segundos, es decir, contactos en intervalos de un minuto. Los planes de contacto del primer plan de contactos en el cual se basan los demás quedan según la tabla 15.

Los retardos en la propagación de la señal no se han reproducido correctamente con tc y netem. Los enlaces se configuran para un retardo de 1 segundo para comunicaciones con satélites, adaptando con mayor precisión las tasas de transferencia.

Tabla 15: Plan de contactos redimensionado para un periodo de prueba máximo de 24 minutos.

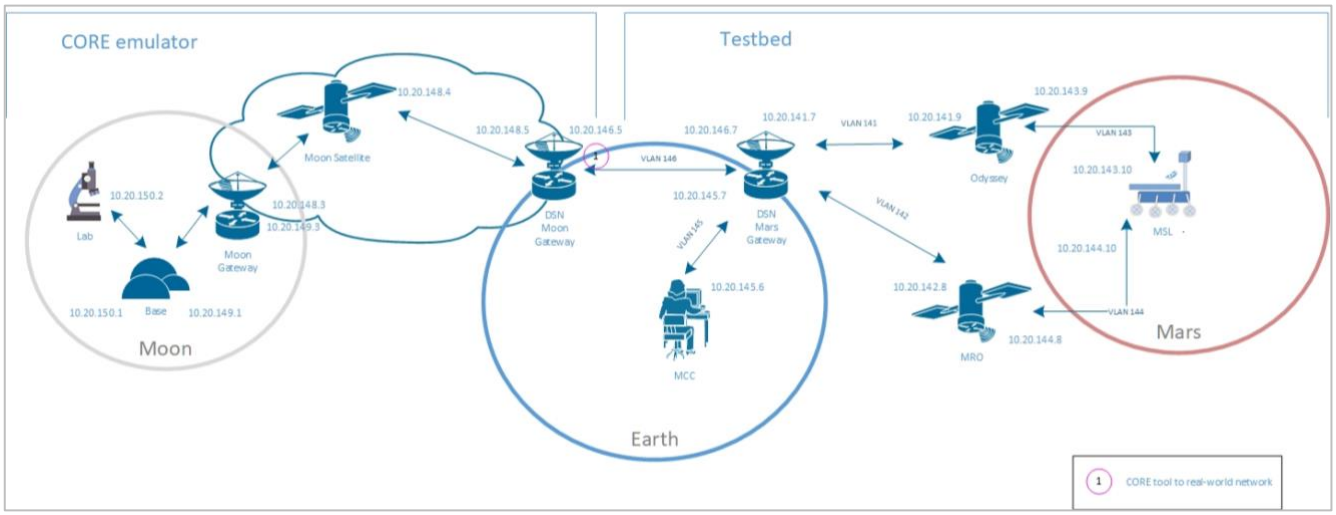
Nodo	Inicio	Fin	Duración [s]
MSL-MRO	7m 47s	7m 55s	8
	19m 53s	20m 2s	9
MSL-Odyssey	9m 54s	9m 59s	5
	11m 49s	11m 59s	10
	22m 45s	22m 56s	11
MRO-DSN	49s	1m 30s	41
	2m 41s	3m 22s	41
	4m 33s	5m 15s	42
	6m 26s	7m 7s	41
	8m 18s	8m 59s	41
	10m 10s	10m 51s	41
	12m 2s	12m 43s	41
	13m 54s	14m 36s	42
	15m 46s	16m 28s	42
	17m 39s	18m 20s	41
	19m 31s	20m 12s	41
	21m 23s	22m 4s	41
	23m 15s	23m 57s	42
Odyssey-DSN	1m 10s	1m 32s	22
	3m 8s	3m30s	22
	5m 7s	5m 29s	22
	7m 5	7m 27s	22
	9m 4s	9m 26s	22

	11m 2s	11m 24s	22
	13m 1s	13m 23s	22
	14m 59s	15m 22s	23
	16m 58s	17m 20s	22
	18m 57s	19m 19s	22
	20m 55s	21m 17s	22
	22m 54s	23m 16s	22

La tase de error está reducida del 3% al 1% en las configuración de los enlaces con netem. Se eliminan en algunas pruebas las tasa de error.

El mapa de red del sistema Luna-Tierra-Marte puede observarse en el esquema de la figura 20. La figura incluye la red utilizada en el emulador en el capítulo 5.

Figura 20: Mapa de red de escenario Luna-Tierra-Marte para banco de pruebas y emulador.



4.4. Nodos y red DTN

Los nodos están configurados con la implementación ION en la versión 4.0.2 con la versión *Bundle Protocol* 7. El software es totalmente adaptable y configurable para cada situación o escenario. Cuenta con diferentes opciones que se pueden establecer en tiempo de compilación y otros parámetros que se deben incluir en los archivos de configuración para iniciar el nodo.

En los archivos de configuración se especifica el plan de contactos, el cual es el mismo para todos los nodos. Además, se detallan los protocolos de convergencia así como los nodos vecinos. En el anexo 4 se detallan configuraciones de los nodos y el proceso de inicio de los servicios para la puesta en marcha de un nodo.

4.5. Pruebas de aplicaciones DTN

Existen diferentes aplicaciones para su uso en redes DTN. En esta sección se describen tres aplicaciones básicas que sirven para ver el comportamiento básico de la red y comprobar su funcionalidad.

4.5.1. bpecho y bping

Es una aplicación similar a ping en redes IP. La aplicación envía un *bundle* desde un nodo origen a un nodo destino y espera respuesta, la cual es indicada en consola. A diferencia de la aplicación ping en una red IP, la aplicación bping envía un *bundle* el cual viaja por los diferentes nodos a su destino a medida que los enlaces están disponibles. Pueden pasar minutos antes de que el nodo origen reciba la contestación.

Se observa, como los paquetes se van acumulando en cada salto en su camino a destino y retorno al origen. En muchas ocasiones, dependiendo de los contactos, el nodo origen puede recibir múltiples respuestas del destino, que estaban acumuladas y pendientes de transmitir.

Para iniciar la aplicación de escucha en el nodo 2 y servicio 1:

```
$ bpecho ipn:2.1
```

Para iniciar la aplicación de petición de respuesta desde el nodo 1 servicio 2 al nodo 1:

```
$ bping ipn:1.2 ipn:2.1
```

4.5.2. Messages (chat)

Messages es una aplicación sencilla de intercambio de mensajes de texto.

Para iniciar la escucha en el nodo 2 servicio 3:

```
$ bpsink ipn:2.3
```

Para enviar un mensaje desde el nodo 4 nodo 2:

```
$ bpsource ipn:2.3 “prueba”
```

4.5.3. Transferencia de archivos: bprecvfile bpsendfile

La aplicación bprecvfile queda en espera de recepción de un archivo. Para iniciar bprecvfile en el nodo 3 servicio 10:

```
$ bprecvfile ipn:3.10
```

La aplicación bpsendfile envía un archivo al nodo destino. Para iniciar el envío del archivo “prueba.txt” desde el nodo 5 servicio 12:

```
$bpsendfile ipn:5.12 ipn.3.12 prueba.txt
```

4.5.4. DTNperf3

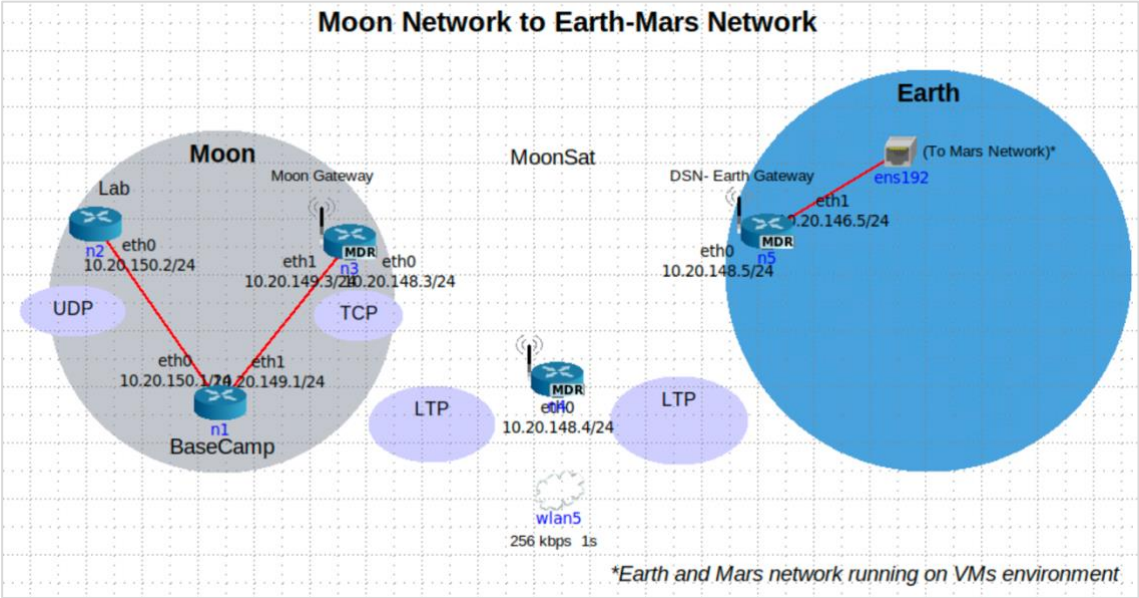
DTNperf 3 es una herramienta utilizada para medir el rendimiento de la red DTN. En el anexo 7 se detallan su uso.

5. Escenario de red IPN 2: emulador.

5.1. Topología y escenario de la red. Enlace con escenario de red IPN 1

En este escenario los elementos de red son supuestos, aunque están basados en [55]. En este caso, se ha supuesto que el orbitador se comunica con las tres estaciones de Tierra DSN. En la figura 21 se muestra la topología de la red tal y como se reproduce en el emulador CORE. El escenario se ha realizado partiendo de los modelos de ejemplo y las herramientas incluidas en el kit de desarrollo para ION.

Figura 21: Entorno de red obtenido del emulador CORE.



El tiempo de propagación de ida y vuelta de la señal desde la Tierra a la Luna está en el rango de 2,4 a 2,7 segundos, con una media de 2,56 segundos, siendo la distancia entre la Tierra y la Luna de 384.000 km. Por simplicidad, se utilizará un segundo OWLT en la propagación de la señal. Los ratios de transferencia se consideran simétricos. Los valores de los enlaces se muestran en la tabla 16.

Tabla 16: valores de los enlaces entre los nodos del escenario propuesto para el sistema Tierra-Luna.

Nodo 1	Nodo 2	Distancia media [AU]	OWLT [s]	Data rate Up/Down [kbps]	BER (%)
DSNGW	MCC	-	0,1	10000	0
DSNGW	MoonSat	0,0026	1	128	0,1
MoonSat	MoonGW	-	1	128	0
MoonGW	MoonBase	-	0,1	10000	0
MoonBase	MoonLab	-	0,1	10000	0

5.2. Plan de contactos

Los contactos entre MoonBase y MoonLab, así como entre Moon-Base y MoonGW son permanentes. MoonGW-MoonSAT tiene un intervalo de 10 segundos cada minuto. DSNGW-MoonSAT tiene un intervalo de 20 segundos cada minuto.

El plan de contactos completo para la red Luna-Tierra-Marte se puede consultar en el anexo 8.

5.3. Nodos, red DTN y pruebas de transmisión

La configuración de los nodos, así como las pruebas de transmisión son similares a las realizadas en el capítulo 4.

Se ha utilizado la función RJ45 del software CORE para unificar el escenario del banco de pruebas con la emulación. De esta forma se ha conseguido iniciar una simulación conjunta de los elementos de la red Tierra-Marte formados por máquinas virtuales, al mismo tiempo que el proyecto de la red Tierra-Luna de CORE. En este contexto, los planes de contactos son los mismos en los 10 nodos que completan el sistema Luna-Tierra-Marte, donde la Tierra tiene el rol de punto de interconexión entre las redes de Marte y la Luna. Se ha comprobado la funcionalidad enviando tráfico desde la VM que simula el rover Curiosity (nodo 10) en Marte a un nodo de la Luna que representa un laboratorio (nodo 2).

Se ha observado, que en un escenario incluyendo transmisiones desde Marte, se producen retardos e interrupciones importantes que provocan congestiones si la transmisión de datos es constante. En el caso de la Luna estos retardos de pocos segundos no representan grandes problemas para la transmisión con la Tierra. En cambio, el tiempo necesario para la transmisión en Marte y las limitaciones en la tasa de transferencia, evidencian la necesidad de planificar las

transmisiones para evitar congestiones. Estas congestiones pueden convertir la red DTN impracticable pese a sus características de retención de datos en cada salto entre el origen y destino. Por lo tanto, no solo es necesario tener un detallado plan de contactos junto con las capacidades de transmisión. Además, se debe tener en cuenta las aplicaciones que transmitirán información y cual será el volumen, para dimensionar el tamaño de los datos y los tiempos de inicio de transmisión, teniendo en cuenta los recursos ofrecidos por los enlaces.

Planificar las transmisiones de datos es un trabajo que ya se realiza, aunque con la red DTN se vuelve más simple, ya que la transmisión es más confiable y eficiente. Las ventajas de las redes DTN, no implican que la capacidad de transmisión se vuelve infinita. Se debe seguir controlando el volumen de información a transmitir en el tiempo.

6. Viabilidad técnica

La viabilidad técnica ya está comprobada. Además, se han verificado y documentado propuestas de implantación y hojas de ruta, tal y como se ha comentado en el apartado 2.4. El sistema está en uso de forma experimental y las propuesta de estándar en IETF ya están en marcha. CCSDS está impulsando la tecnología como solución para SSI, y JPL/NASA también lo está haciendo con la evolución de ION y las planificación de las misiones contando con esta tecnología. La estandarización, las aplicaciones experimentales, así como las numerosas publicaciones están consolidando la propuesta de DTN como la arquitectura para IPN y SSI, así como diferentes misiones futuras en la Luna y Marte.

En los siguientes apartados se consideran los puntos fuertes y débiles desde la opinión del autor.

6.1. Puntos fuertes

a. La era de la información y el desarrollo de Internet nos ha llevado a un mundo donde los datos no dejan de crecer, y cada vez es más necesario permanecer on-line de forma permanente. Esta necesidad se traslada igualmente a redes IoT, redes para misiones espaciales y otras situaciones con difícil conectividad. La tecnología DTN y su capacidad de retención de la información es decisiva para soportar esta necesidad.
b. Otros paradigmas se suman a la capacidad de conservación de la información de una red DTN. Los esquemas de identificación propuestos son muy interesantes. Además, la arquitectura y las propuestas de estándar BP y el concepto de las capas de convergencia, generan una abstracción entre las aplicaciones y la red de datos que puede cambiar la forma de diseñar aplicaciones.
c. DTN es una tecnología que ayuda a la investigación de nuevos protocolos de enrutamiento. Estos trabajos proponen enfoques diferentes para el enrutamiento, que pueden habilitar nuevas aplicaciones.
d. Las telecomunicaciones son esenciales para el desarrollo del *new space*. Al igual que otras disciplinas, es imprescindible que existan soluciones inminentes para un posible aumento drástico del volumen de datos, como consecuencia del crecimiento progresivo de misiones y programas espaciales en todo el mundo por parte de diferentes agencias y empresas privadas. DTN es una propuesta viable para este desarrollo.

6.2. Puntos débiles

e. Los contactos deterministas pueden ser viables si una entidad tiene el control y la gestión de los recursos. Si existen hipotéticos operadores de servicios de red de Internet Interplanetario -quienes ofrecen servicios a sus clientes-, es difícil saber qué se va a transmitir y en qué momento, para medir los recursos que se van a utilizar. Si en un futuro se llega a un punto de uso generalizado, no tendrá sentido utilizar contactos deterministas. El avance en protocolos basados en modelos predictivos, la proliferación de nodos y el aumento de la capacidad de transmisión de los enlaces, pueden hacer crecer la capacidad de la red IPN y alejarse progresivamente de un enrutamiento determinista.

f. Gran parte de las aplicaciones de red están programadas para trabajar con una red orientada a conexión. Aunque en la topología de una red, se utilicen *gateways* para interactuar con protocolos DTN, las aplicaciones pueden sufrir fallos debidos a tiempos de espera, reinicios de sistema y desconexiones. Adaptar el software existente para ser válido en IPN no es una tarea trivial. Es una transformación que requiere un esfuerzo en la codificación y adaptación de aplicaciones o programación de software nuevo. Este punto representa una dificultad para la adopción de la tecnología. Puede facilitar esta tarea, la inclusión de adaptadores DTN y/o capas de abstracción en los sistemas operativos más comunes, además de paquetes de librerías DTN en diferentes lenguajes de programación, con el fin de simplificar el desarrollo de aplicaciones.

g. Es necesaria la inclusión de entidades privadas en la industria espacial para operar sus propias redes y servicios de todo tipo para las misiones. Es difícil el desarrollo competitivo y la proliferación de nuevos incentivos en un mercado dominado y controlado por instituciones públicas y/o gubernamentales. La era del *new space* requiere que las empresas privadas participen ofreciendo servicios, los cuales puedan ser contratados a su vez por gobiernos, empresas e incluso usuarios. De esta forma, se simplifica la puesta en marcha de misiones e ideas de negocio si se consigue un tejido empresarial. Este aspecto facilitaría el desarrollo de la industria y aceleraría la consolidación de la red IPN.

7. Conclusiones

7.1. Lecciones aprendidas

Las lecciones aprendidas tienen relación con los objetivos propuestos para este proyecto.

El punto más interesante ha sido estudiar todo el proceso de ingeniería que se origina desde el momento que surge una necesidad hasta que se da forma a una solución final. Entender que en cada industria existen unos actores que participan -y deben participar- en la definición formal de una tecnología, cambio, o mejora, así como el tiempo necesario, el gran esfuerzo de muchas personas y las dificultades que surgen en cada paso.

En segundo lugar, se ha conseguido obtener una visión más profunda de los diferentes aspectos para tener en cuenta en el desarrollo de software de transmisión de datos. Además, la investigación de opciones y propuestas expuestas en diferentes publicaciones -revistas, universidades e instituciones- desvelan los pilares de las tecnologías sobre los que después se fundamentan los productos comerciales y soluciones del mercado. Es decir, es en estas fuentes de información donde se encuentran las bases de las tecnologías utilizadas después por los fabricantes. Aunque, hay que tener en cuenta que es un trabajo denso y pesado, comparado con el estudio de un manual, guía o curso técnico de un fabricante.

En tercer lugar, pese a los esfuerzos para generar nuevas tecnologías y soluciones, queda el paso final y tal vez el más impredecible en cuanto a resultado: la adopción de la tecnología en la industria o mercado. Convertir una tecnología o solución en producto, puede no ser sencillo, y puede no ser un paso natural. Sobre todo, en un mercado dominado por otras soluciones clásicas y profundamente ancladas en la industria o en la sociedad. Es interesante tener en cuenta este punto en el momento que pretende desarrollar un producto o solución para el mercado: ¿es una solución complementaria o disruptiva? ¿qué cambios se deben producir para la adopción de dicha tecnología o producto? Para las redes DTN, es un desafío todavía pendiente.

7.2. Tiempo dedicada a cada PAC y análisis de planificación

Por una parte, la planificación -intervalos de fechas- propuestos inicialmente se han ajustado muy bien a la realidad. Estos planes han necesitado pequeños ajustes sin un impacto importante en las fechas de entrega. En cualquier caso, el tiempo disponible es demasiado limitado para el proyecto. Estas limitaciones de fecha acarrean una toma de decisiones constante en cuanto a qué tareas se dan por buenas y cuales se desechan; que caminos se abandonan para seguir otros más eficientes, pensando en la proporción horas/resultados.

Un ejemplo de este punto se ha producido en el objetivo extraordinario de conseguir una telemetría de la red DTN similar al que se puede obtener en una red IP -en este proyecto, realizado con telegraf e influxDB. Esta sección se ha realizado finalmente midiendo el tráfico de red IP que servía de transporte para la red DTN, pero no se han medido los datos de las red DTN en particular. Esta cuestión puede ser profundizada en un proyecto independiente, valorando diferentes opciones, como utilizar las funciones de las implementaciones, usando una capa de abstracción común para distintas implementaciones, así como evaluar diferentes bases de datos de tiempo y herramientas de representación y gestión de los datos. Por ese motivo, las medidas para observar el tráfico generado se han limitado a la red IP y se ha asumido que la opción de adaptar estas herramientas para la red DTN era inviable en este proyecto

Por otra parte, las horas dedicadas han sido superiores a las planificadas. Dentro de cada apartado de la planificación, las horas necesarias han superado las proyectadas. Esta cuestión tiene que ver con la percepción personal para valorar el tiempo necesario, y es una ajuste pendiente de realizar en futuros proyectos profesionales y académicos.

7.3. Objetivos alcanzados

Siguiendo los objetivos propuestos en el punto 1.3, se pueden valorar cada uno de ellos en diferentes grados de alcance, pero todos han sido trabajados. Las habilidades y el tiempo dedicado para los objetivos propuestos han dado forma a los resultados.

Los objetivos mejor desarrollados han sido el estudio de las características de una arquitectura de red, la integración de un entorno de pruebas y simulación y el análisis de una solución de red.

El objetivo menos desarrollado ha sido el análisis del código de las implementaciones. En este punto, se han valorado diferentes implementaciones en sus características relacionadas con la tecnología en estudio, algunas de estas estimaciones relacionadas en las partes de diseño y programación de las aplicaciones, que ha quedado en una evaluación fundamental.

7.4. Ampliación del trabajo

El trabajo realizado ha ayudado a identificar distintos ámbitos a desarrollar:

- Trabajar las implementaciones en mayor profundidad y estudiar su adaptación y optimización para diferentes sistemas y escenarios.
- Desarrollo de herramientas de gestión y administración de implementaciones.
- Adaptación de software DTN para sistemas embebidos y su uso en diferentes ámbitos (IoT, Cubesat, otros).
- Estudio de protocolos de enrutamiento y topologías de red DTN.
- Desarrollo de software para redes DTN.
- Sistemas de red y software DTN en la nube.

Anexos

Anexo 1: Tablas comparativas BP6 Y BP7

Los bloques de BP7 en comparación con la versión BP6 están definidos por IANA según la tabla 17. En la tabla 18 se comparan los *flags* de control de procesos de *bundles*. En la tabla 19 se comparan *flags* de control de proceso de bloques. En la tabla 20 se muestra una comparación de los códigos utilizados en las notificaciones de estado de los *bundles*.

Tabla 17: Tipos de bloques para versiones 6 y 7.

Versión BP	Valor	Descripción
-	0	Reservado
6, 7	1	Bundle Payload Block
6	2	Bundle Authentication Block
6	3	Payload Integrity Block
6	4	Payload Confidentiality Block
6	5	Previous-Hop Insertion Block
7	6	Previous node
7	7	Bundle age (en milisegundos)
6	8	Metadata Extension Block
6	9	Extension Security Block
7	10	Hop Count
7	11-191	Sin asignar
6, 7	192-255	Reservado para uso privado o experimental

(fuente: borrador Bundle Protocol Version 7).

Tabla 18: Flags de control de procesos de bundles para versiones 6 y 7.

Versión BP	Posición de bit (de derecha a izquierda)	Descripción
6, 7	0	El *bundle* es un fragmento
6, 7	1	ADU es un registro administrativo
6, 7	2	El *bundle* no debe ser fragmentado
6	3	Solicitada transferencia de custodia
6	4	El *endpoint* destino es un *singleton*
6, 7	5	La aplicación solicita respuesta

7	6	Tiempo de estado solicitada en las notificaciones
6	7	*Class of Service*[70], prioridad
6	8	*Class of Service*, prioridad
6	9	*Class of Service*, prioridad
6	10	*Class of Service*, prioridad
6	11	*Class of Service*, prioridad
6	12	*Class of Service*, prioridad
6	13	*Class of Service*, prioridad
6, 7	14	Solicitad de notificación de recepción de *bundle*
6	15	Solicitad de notificación de aceptación de custodia
6, 7	16	Solicitad de notificación de reenvío de *bundle*
6, 7	17	Solicitud de notificación de entrega de *bundle*
6, 7	18	Solicitud de notificación de eliminación de *bundle*
6, 7	19	Reservado
6, 7	20	Reservado
	21-63	No asignado

(fuente: borrador Bundle Protocol Version 7).

Tabla 19: Flags de control de procesos de bloques para versiones 6 y 7.

Versión BP	**Posición de bit (de derecha a izquierda)**	**Descripción**
6, 7	0	El bloque se debe replicar en cada fragmento
6, 7	1	Notificar el estado de transmisión si el bloque no puede ser procesado
6, 7	2	Eliminar *bundle* si el bloque no puede ser procesado
6	3	Último bloque
6, 7	4	Descartar bloque si no puede ser procesado
6	5	El bloque fue reenviado sin ser procesado
6	6	El bloque contiene un campo de referencia al EID
	7-63	No asignado

(fuente: borrador Bundle Protocol Version 7).

[70] Class of Service (CoS) es una etiqueta para clasificar un *bundle* con fines de clasificación y priorización.

Tabla 20: Códigos de notificaciones de estado del bundle para versiones 6 y 7.

Versión BP	Valor	Descripción
6, 7	0	Sin información adicional
6, 7	1	Tiempo de vida expirado
6, 7	2	Reenviado mediante enlace unidireccional
6, 7	3	Transmisión cancelada
6, 7	4	Almacenamiento agotado
6, 7	5	EID de destino no disponible
6, 7	6	Ruta a destino desconocida
6, 7	7	No existe oportunidad de contacto con el siguiente nodo de la ruta
6, 7	8	Bloque ilegible
7	9	Límite de saltos sobrepasado
7	10	Transmisión interrumpida
7	11	Bloque no soportado
	12-254	No asignado
6, 7	255	Reservado

(fuente: borrador Bundle Protocol Version 7).

Anexo 2: Cálculo de contactos

El intervalo de tiempo seleccionado para consultar el plan de contactos corresponde a 24 horas del día 1 de mayo de 2020. En esta franja temporal se recogen los contactos entre MSL y orbitadores, así como los orbitadores y DSN. En el caso DSN, para una mayor precisión se requiere la planificación en la fecha seleccionada para las tres Estaciones de Tierra. De esta forma, se debe obtener la intersección entre el dato del tiempo de contacto por la dinámica y movimiento de planetas y orbitadores en WGC, con el tiempo disponible de las antenas DSN en la planificación. No está disponible la información de planificación para la fecha indicada, por ese motivo los contactos se obtienen únicamente con los datos de WGC.

Los contactos MSL-MRO se calculan con los parámetros de la figura 23 y se puede ver el resultado en la tabla 21 y en la figura 24. MRO tiene periodo orbital de unos 112 minutos, por lo que realiza aproximadamente 13 órbitas/sol[71].

Figura 22: parámetros WGC para los contactos MSL-MRO.

Input Values	
Calculation type	Occultation Event Finder
Occultation type	Any
Front body	MSL
Front body shape	Point
Back body	MARS
Back body shape	Ellipsoid
Back body frame	IAU_MARS
Observer	MRO
Light propagation	No correction
Time system	UTC
Time format	Calendar date and time
Time range	2020 may 1 00:01 to 2020 may 1 23:59
Step	1 minutes
Output time unit	minutes
Complement result window	no
Result interval adjustment	No adjustment
Result interval filtering	No filtering

Figura 23: contactos MSL-MRO.

[71] Un sol es a Marte lo que un día es a la Tierra. Un sol dura 24 horas 39 minutos y 35,244 segundos. (https://www.giss.nasa.gov/tools/mars24/help/notes.html)

Tabla 21: Contactos MSL-MRO.

Start Time	Stop Time	Duration (mins)
2020-05-01 07:47:10.506242 UTC	2020-05-01 07:55:27.169136 UTC	8,2777149
2020-05-01 19:53:00.157287 UTC	2020-05-01 20:01:47.459978 UTC	8,78837818

Los contactos MSL-Odyssey se calculan con los parámetros de la figura 25 y se puede ver el resultado en la tabla 22 y en la figura 26. MRO tiene una órbita de unos 120 minutos, por lo que realiza aproximadamente 12 órbitas/sol.

Figura 24: parámetros WGC para los contactos MSL-Odyssey.

Input Values

Calculation type	Occultation Event Finder
Occultation type	Any
Front body	MSL
Front body shape	Point
Back body	MARS
Back body shape	Ellipsoid
Back body frame	IAU_MARS
Observer	MARS ODYSSEY
Light propagation	No correction
Time system	UTC
Time format	Calendar date and time
Time range	2020 may 1 00:01 to 2020 may 1 23:59
Step	1 minutes
Output time unit	minutes
Complement result window	no
Result interval adjustment	No adjustment
Result interval filtering	No filtering

Figura 25: contactos MSL-Odyssey.

Tabla 22: Contactos MSL-Odyssey

Start Time	Stop Time	Duration (mins)
2020-05-01 09:54:43.041707 UTC	2020-05-01 09:59:19.494695 UTC	4,6075498
2020-05-01 11:49:36.885219 UTC	2020-05-01 11:59:14.666852 UTC	9,6296939
2020-05-01 22:45:31.310765 UTC	2020-05-01 22:56:25.570759 UTC	10,90433322

Los contactos orbitador-DSN se dividen en tres listas de contactos:

1. Orbitador-DSS55: Estación de Tierra DSN de Madrid. Antena DSS-55.
2. Orbitador-DSS25: Estación de Tierra DSN de Goldstone. Antena DSS-25.
3. Orbitador-DSS35: Estación de Tierra DSN de Canberra. Antena DSS-35.

Para realizar el cálculo DSN, en primer lugar se calcula el contacto del orbitador con la antena DSS de referencia para la fecha propuesta desde la perspectiva del orbitador, interpretando a la antena como el cuerpo que bloquea parcialmente la visibilidad con la Tierra. En el caso de MRO, con este cálculo obtenemos el intervalo de la figura 27. Este intervalo no tiene en cuenta los periodos orbitales del MRO y su ocultación tras Marte, impidiendo la visión directa con la antena DSS. Es por este motivo que hay que guardar el resultado para pasar al segundo paso.

Figura 26: MRO-DSS25(Goldstone).

Opción de visibilidad de la antena DSS25 desde la perspectiva del MRO teniendo en cuenta la rotación de la Tierra.

En segundo lugar, se calcula el contacto de con el orbitador para el intervalo de tiempo almacenado en el primer cálculo, desde la perspectiva de la antena DSS, interpretando el propio orbitador como el cuerpo que bloquea parcialmente la visibilidad de Marte.

Siguiendo estos pasos obtenemos los contactos de los orbitadores y las tres Estaciones de Tierra como se puede apreciar en las figuras 28 a 39, y la tablas 23 a 29.

Figura 27: parámetros WGC para los contactos MRO-DSS-55 (Madrid).

Input Values

Calculation type	Occultation Event Finder
Occultation type	Any
Front body	MRO
Front body shape	Point
Back body	MARS
Back body shape	Ellipsoid
Back body frame	IAU_MARS
Observer	DSS-55
Light propagation	No correction
Time system	UTC
Time format	Calendar date and time
Time windows	["2020-05-01 02:18:35.965346 UTC to 2020-05-01 12:16:24.983017 UTC"]
Step	1 minutes
Output time unit	minutes
Complement result window	no
Result interval adjustment	No adjustment
Result interval filtering	No filtering

Figura 28: contactos MRO-DSS-55 (Madrid).

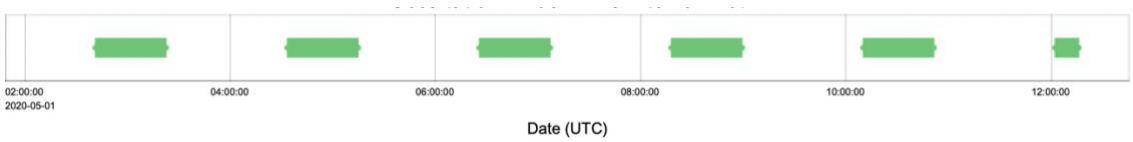


Tabla 23: Contactos MRO-DSS-55 (Madrid).

Start Time	Stop Time	Duration (mins)
2020-05-01 02:41:30.244471 UTC	2020-05-01 03:22:50.933126 UTC	41,34481091
2020-05-01 04:33:45.683072 UTC	2020-05-01 05:15:06.235549 UTC	41,34254128
2020-05-01 06:26:00.738692 UTC	2020-05-01 07:07:21.054009 UTC	41,33858862
2020-05-01 08:18:13.856765 UTC	2020-05-01 08:59:34.164519 UTC	41,33846257
2020-05-01 10:10:24.184044 UTC	2020-05-01 10:51:44.749711 UTC	41,34276112
2020-05-01 12:02:33.645102 UTC	2020-05-01 12:16:24.983017 UTC	13,85563191

Figura 29: parámetros WGC para los contactos MRO-DSS-25 (Goldstone).

Input Values

Calculation type	Occultation Event Finder
Occultation type	Any
Front body	MRO
Front body shape	Point
Back body	MARS
Back body shape	Ellipsoid
Back body frame	IAU_MARS
Observer	DSS-25
Light propagation	No correction
Time system	UTC
Time format	Calendar date and time
Time windows	["2020-05-01 09:39:01.942013 UTC to 2020-05-01 19:56:10.085621 UTC"]
Step	1 minutes
Output time unit	minutes
Complement result window	no
Result interval adjustment	No adjustment
Result interval filtering	No filtering

Figura 30: contactos MRO-DSS-25 (antena de Goldstone).

Tabla 24: Contactos MSL-DSS-25 (Goldstone).

Start Time	Stop Time	Duration (mins)
2020-05-01 10:10:24.180144 UTC	2020-05-01 10:51:44.759674 UTC	41,34299216
2020-05-01 12:02:33.647940 UTC	2020-05-01 12:43:54.558233 UTC	41,34850488
2020-05-01 13:54:43.607847 UTC	2020-05-01 14:36:04.351372 UTC	41,34572542
2020-05-01 15:46:55.916071 UTC	2020-05-01 16:28:16.599024 UTC	41,34471589
2020-05-01 17:39:10.159722 UTC	2020-05-01 18:20:30.870663 UTC	41,34518234
2020-05-01 19:31:25.127480 UTC	2020-05-01 19:56:10.085621 UTC	24,74930236

Figura 31: parámetros WGC para los contactos MRO-DSS-35 (Canberra).

Input Values

Calculation type	Occultation Event Finder
Occultation type	Any
Front body	MRO
Front body shape	Point
Back body	MARS
Back body shape	Ellipsoid
Back body frame	IAU_MARS
Observer	DSS-35
Light propagation	No correction
Time system	UTC
Time format	Calendar date and time
Time windows	["2020-05-01 00:01:00.000000 UTC to 2020-05-01 03:46:33.823691 UTC", "2020-05-01 14:22:21.487661 UTC to 2020-05-01 23:59:00.000000 UTC"]
Step	1 minutes
Output time unit	minutes
Complement result window	no
Result interval adjustment	No adjustment
Result interval filtering	No filtering

Figura 32: contactos MRO-DSS-35 (Canberra).

Tabla 25: Contactos MRO-DSS-35 (Canberra).

Start Time	Stop Time	Duration (mins)
2020-05-01 00:49:17.997001 UTC	2020-05-01 01:30:38.456870 UTC	41,34099781
2020-05-01 02:41:30.195883 UTC	2020-05-01 03:22:50.870769 UTC	41,34458145
2020-05-01 14:22:21.487661 UTC	2020-05-01 14:36:04.307643 UTC	13,71366636
2020-05-01 15:46:55.867293 UTC	2020-05-01 16:28:16.563350 UTC	41,34493429
2020-05-01 17:39:10.116917 UTC	2020-05-01 18:20:30.841507 UTC	41,34540984
2020-05-01 19:31:25.090415 UTC	2020-05-01 20:12:45.965027 UTC	41,34791019
2020-05-01 21:23:38.574819 UTC	2020-05-01 22:04:59.538994 UTC	41,34940291
2020-05-01 23:15:48.548705 UTC	2020-05-01 23:57:09.534392 UTC	41,34976145

Se obtiene una lista de contactos (tabla 23) final del Odyssey con DSN uniendo intervalos de las tres antenas. Se ha unificado en una única lista todos los contactos para las tres estaciones de Tierra con el objetivo de simplificar, siendo DSN un único nodo en la red DTN de pruebas. Se tiene en cuenta que el contacto, retardo y tasa de transferencia entre DSN con MCC es permanente y típico de una red IP.

Tabla 26: Contactos MRO-DSN para el día 1 de mayo de 2020.

Start Time	Stop Time	Duration (mins)
2020-05-01 00:49:17.997001 UTC	2020-05-01 01:30:38.456870 UTC	41,34099781
2020-05-01 02:41:30.195883 UTC	2020-05-01 03:22:50.870769 UTC	41,34458145
2020-05-01 04:33:45.683072 UTC	2020-05-01 05:15:06.235549 UTC	41,34254128
2020-05-01 06:26:00.738692 UTC	2020-05-01 07:07:21.054009 UTC	41,33858862
2020-05-01 08:18:13.856765 UTC	2020-05-01 08:59:34.164519 UTC	41,33846257
2020-05-01 10:10:24.180144 UTC	2020-05-01 10:51:44.759674 UTC	41,34299216
2020-05-01 12:02:33.647940 UTC	2020-05-01 12:43:54.558233 UTC	41,34850488
2020-05-01 13:54:43.607847 UTC	2020-05-01 14:36:04.351372 UTC	41,34572542
2020-05-01 15:46:55.867293 UTC	2020-05-01 16:28:16.563350 UTC	41,34493429
2020-05-01 17:39:10.116917 UTC	2020-05-01 18:20:30.841507 UTC	41,34540984
2020-05-01 19:31:25.090415 UTC	2020-05-01 20:12:45.965027 UTC	41,34791019
2020-05-01 21:23:38.574819 UTC	2020-05-01 22:04:59.538994 UTC	41,34940291
2020-05-01 23:15:48.548705 UTC	2020-05-01 23:57:09.534392 UTC	41,34976145

Figura 33: parámetros WGC para los contactos Odyssey-DSS-55 (Madrid).

Input Values

Calculation type	Occultation Event Finder
Occultation type	Any
Front body	MARS ODYSSEY
Front body shape	Point
Back body	MARS
Back body shape	Ellipsoid
Back body frame	IAU_MARS
Observer	DSS-55
Light propagation	No correction
Time system	UTC
Time format	Calendar date and time
Time windows	["2020-05-01 02:18:35.740828 UTC to 2020-05-01 12:16:25.469376 UTC"]
Step	1 minutes
Output time unit	minutes
Complement result window	no
Result interval adjustment	No adjustment
Result interval filtering	No filtering

Figura 34: contactos Odyssey-DSS-55 (Madrid).

Tabla 27: Contactos Odyssey-DSS-55 (Madrid).

Start Time	Stop Time	Duration (mins)
2020-05-01 03:08:38.840161 UTC	2020-05-01 03:30:36.049365 UTC	21,95348674
2020-05-01 05:07:10.148250 UTC	2020-05-01 05:29:04.061724 UTC	21,8985579
2020-05-01 07:05:46.836907 UTC	2020-05-01 07:27:37.604293 UTC	21,8461231
2020-05-01 09:04:23.738312 UTC	2020-05-01 09:26:15.578073 UTC	21,86399603
2020-05-01 11:02:58.355498 UTC	2020-05-01 11:24:54.028660 UTC	21,92788604

Figura 35: parámetros WGC para los contactos Odyssey-DSS-25 (Goldstone).

Input Values

Calculation type	Occultation Event Finder
Occultation type	Any
Front body	MARS ODYSSEY
Front body shape	Point
Back body	MARS
Back body shape	Ellipsoid
Back body frame	IAU_MARS
Observer	DSS-25
Light propagation	No correction
Time system	UTC
Time format	Calendar date and time
Time windows	["2020-05-01 09:39:01.918999 UTC to 2020-05-01 19:56:10.692029 UTC"]
Step	1 minutes
Output time unit	minutes
Complement result window	no
Result interval adjustment	No adjustment
Result interval filtering	No filtering

Figura 36: contactos Odyssey-DSS-25 (Goldstone).

Tabla 28: Contactos Odyssey-DSS-25 (Goldstone).

Start Time	Stop Time	Duration (mins)
2020-05-01 11:02:58.206517 UTC	2020-05-01 11:24:54.034586 UTC	21,93046782
2020-05-01 13:01:29.609741 UTC	2020-05-01 13:23:31.448364 UTC	22,03064372
2020-05-01 14:59:59.009840 UTC	2020-05-01 15:22:06.843598 UTC	22,13056263
2020-05-01 16:58:29.847551 UTC	2020-05-01 17:20:36.716446 UTC	22,11448157
2020-05-01 18:57:04.909924 UTC	2020-05-01 19:19:06.484329 UTC	22,02624009

Figura 37: parámetros WGC para los contactos Odyssey-DSS-35 (Canberra).

Input Values

Calculation type	Occultation Event Finder
Occultation type	Any
Front body	MARS ODYSSEY
Front body shape	Point
Back body	MARS
Back body shape	Ellipsoid
Back body frame	IAU_MARS
Observer	DSS-35
Light propagation	No correction
Time system	UTC
Time format	Calendar date and time
Time windows	["2020-05-01 00:01:00.000000 UTC to 2020-05-01 03:46:34.290969 UTC", "2020-05-01 14:22:21.707580 UTC to 2020-05-01 23:59:00.000000 UTC"]
Step	1 minutes
Output time unit	minutes
Complement result window	no
Result interval adjustment	No adjustment
Result interval filtering	No filtering

Figura 38: contactos Odyssey-DSS-35 (Canberra).

Tabla 29: Contactos Odyssey-DSS-35 (Canberra).

Start Time	Stop Time	Duration (mins)
2020-05-01 01:10:10.249825 UTC	2020-05-01 01:32:02.892500 UTC	21,87737792
2020-05-01 03:08:38.975822 UTC	2020-05-01 03:30:36.200194 UTC	21,95373953
2020-05-01 14:59:58.882106 UTC	2020-05-01 15:22:06.997342 UTC	22,13525392
2020-05-01 16:58:29.690195 UTC	2020-05-01 17:20:36.853005 UTC	22,11938016
2020-05-01 18:57:04.756032 UTC	2020-05-01 19:19:06.602976 UTC	22,03078241
2020-05-01 20:55:41.659180 UTC	2020-05-01 21:17:41.425944 UTC	21,99611274
2020-05-01 22:54:17.556723 UTC	2020-05-01 23:16:20.799647 UTC	22,05404873

Se obtiene una lista de contactos (tabla 30) final del Odyssey con DSN uniendo intervalos de las tres antenas, tal y como se ha comentado en la lista de contactos del MRO.

Tabla 30: Contactos Odyssey-DSN para el día 1 de mayo de 2020.

Start Time	Stop Time	Duration (mins)
2020-05-01 01:10:10.249825 UTC	2020-05-01 01:32:02.892500 UTC	21,87737792
2020-05-01 03:08:38.975822 UTC	2020-05-01 03:30:36.200194 UTC	21,95373953
2020-05-01 05:07:10.148250 UTC	2020-05-01 05:29:04.061724 UTC	21,8985579
2020-05-01 07:05:46.836907 UTC	2020-05-01 07:27:37.604293 UTC	21,8461231
2020-05-01 09:04:23.738312 UTC	2020-05-01 09:26:15.578073 UTC	21,86399603
2020-05-01 11:02:58.355498 UTC	2020-05-01 11:24:54.028660 UTC	21,92788604
2020-05-01 13:01:29.609741 UTC	2020-05-01 13:23:31.448364 UTC	22,03064372
2020-05-01 14:59:58.882106 UTC	2020-05-01 15:22:06.997342 UTC	22,13525392
2020-05-01 16:58:29.690195 UTC	2020-05-01 17:20:36.853005 UTC	22,11938016
2020-05-01 18:57:04.756032 UTC	2020-05-01 19:19:06.602976 UTC	22,03078241
2020-05-01 20:55:41.659180 UTC	2020-05-01 21:17:41.425944 UTC	21,99611274
2020-05-01 22:54:17.556723 UTC	2020-05-01 23:16:20.799647 UTC	22,05404873

Anexo 3: netem y uso de iperf3

La herramienta netem se utiliza para adecuar los adaptadores a la situación del escenario a simular. A continuación se muestra un ejemplo de una adaptación.

Cada VM tiene un adaptador para comunicarse con sus nodos vecinos. En el ejemplo, se adapta la transmisión desde VM que representa a MSL Curiosity con la VM MRO en dicho sentido. El tráfico desde MRO hacia MSL se debe configurar en la VM MRO en el interfaz de red empleado para conectar con MSL. Es decir, las adaptaciones hacen referencia al tráfico de salida.

En el código del ejemplo se observan los parámetros configurados:

1. tbf: (*Token Bucket Buffer Filter*), son colas sin clase donde un contenedor o *token* es la unidad de información donde se van almacenando los paquetes a transmitir.
2. *Delay*: es el retardo en la transmisión de los paquetes.
3. *Jitter*: margen de variación en el retardo. El retardo será *latency* +- *jitter*.
4. *Loss*: pérdida de paquetes (PER).
5. *Rate*: es la tasa de transferencia aproximada que el sistema tendrá como referencia para limitar el tráfico de salida.
6. *Burst*: es la capacidad de buffer
7. *IF*: es el interfaz de red de la VM que se está configurando.
8. *Latency*: es el tiempo que los paquetes pueden estar esperando un *token* para ser enviado.
9. *Jitter*: margen de variación en el retardo. El retardo será *latency* +- *jitter*.
10. *Loss*: pérdida de paquetes (PER).

Muestra parcial del script para adaptación MSL → MRO.

```
TC=/sbin/tc
# Traffic rate
RATE=128kbit
# Burst
BURST=32kb
Interface
 IF=ens192
 # Latency tbf
 LATENCY=5s
 # Delay
```

```
DELAY=1s
# Jitter
JITTER=100ms
# Loss (PER)
LOSS=1%
start() {

    $TC qdisc add dev $IF root handle 1:0 netem delay $DELAY $JITTER loss $LOSS
    $TC qdisc add dev $IF parent 1:1 handle 10: tbf rate $RATE burst $BURST latency $LATENCY
}
```

TC modifica el planificador (qdisc) para el interface en el tráfico de salida (root). Con netem queda definido el retardo en la transmisión así como el *jitter* que corresponde a la variación en la tasa de transferencia.

El siguiente paso consiste en probar la transmisión para el adaptador que se ha configurado con iperf3.

Servidor:

```
$ iperf3 -s
```

Cliente:

```
$ iperf3 -c 10.20.144.8
```

Respuesta:

```
-----------------------------------------------------------
Server listening on 5201
-----------------------------------------------------------

Accepted connection from 10.20.144.10, port 45018
[  5] local 10.20.144.8 port 5201 connected to 10.20.144.10 port 45020
[ ID] Interval           Transfer     Bitrate
[  5]   0.00-1.00   sec  0.00 Bytes  0.00 bits/sec
[  5]   1.00-2.00   sec  14.1 KBytes   116 Kbits/sec
[  5]   2.00-3.00   sec  8.48 KBytes  69.5 Kbits/sec
[  5]   3.00-4.00   sec  11.3 KBytes  92.7 Kbits/sec
[  5]   4.00-5.00   sec  14.1 KBytes   116 Kbits/sec
[  5]   5.00-6.00   sec  19.8 KBytes   162 Kbits/sec
[  5]   6.00-7.00   sec  8.48 KBytes  69.5 Kbits/sec
```

```
[  5]   7.00-8.00   sec  22.6 KBytes   185 Kbits/sec
[  5]   8.00-9.00   sec  14.1 KBytes   116 Kbits/sec
[  5]   9.00-10.00  sec  17.0 KBytes   139 Kbits/sec
[  5]  10.00-11.00  sec  11.3 KBytes  92.6 Kbits/sec
[  5]  11.00-12.00  sec  17.0 KBytes   139 Kbits/sec
[  5]  12.00-13.00  sec  17.0 KBytes   139 Kbits/sec
[  5]  13.00-14.00  sec  14.1 KBytes   116 Kbits/sec
[  5]  14.00-15.00  sec  14.1 KBytes   116 Kbits/sec
[  5]  15.00-16.00  sec  14.1 KBytes   116 Kbits/sec
[  5]  16.00-16.17  sec  5.66 KBytes   266 Kbits/sec
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate
[  5]   0.00-16.17  sec   223 KBytes   113 Kbits/sec                  receiver
```

Anexo 4: Configuración de un nodo DTN

La configuración de un nodo en ION está formada por diferentes archivos los cuales se ejecutan desde la herramienta de administración oportuno. En este anexo se muestra la configuración de la VM que representa el nodo MSL. Está configurado con la capa de convergencia TCP.

Archivo 1: curiosity.ionrc

```
# FILE: curiosity.ionrc
# CONTENT: ION node management commands file
# FOR: ionadmin
# NETWORK: Mars-Earth-Moon
# NODE: 10
# DATE: 2021-05-01
#
# INITIALIZE
# [1] ion node number : 10
# [2] ionconfig file name : curiosity.ionconfig

1 10 curiosity.ionconfig

#
# START

s
```

El archivo contiene dos líneas de comando. Consiste en identificar el número de nodo y el archivo de configuración de ION (curiosity.ionconfig) e iniciar el nodo.

Archivo 2: curiosity.ionconfig

```
#  FILE:     curiosity.ionconfig
#  CONTENT:  ION node configuration parameters file
#  FOR:      ionadmin
#  NETWORK:  Mars-Earth-Moon
#  NODE:     10
```

```
# DATE:     2021-05-1
#
# CONFIGFLAGS
# [1]ion configuration flags : 15  [SDR_IN_DRAM + SDR_FILE + SDR_REVERSIBLE + SDR_BOUNDED]
wmKey          0
sdrName        ion
wmSize         10000000
configFlags    1
heapWords      10000000
pathName       /home/user/ionnode/ion
```

wmKey: identificador de memoria compartida del nodo.
sdrName: identificador de la base de datos ION.
wmSize: tamaño del bloque de la memoria dinámica.
configFlags: 4 bits que caracterizan la ubicación de la base de datos. 1 indica ‘SDR_IN_RAM’.
heapWords: número de palabras (64 bits por palabra en una máquina de 64bits).
pathName: ubicación de la base de datos si su ubicación es en un archivo.

Archivo 3: curiosity.ionsecrc

```
# Initialization command (command 1).
1
# Select level of "echo control" activities
# 0 = None; 1 = print to both log an stdout

e 1
```

Archivo de configuración de seguridad. La configuración está por defecto según la plantilla y no se han realizado configuraciones específicas.

Archivo 4: curiosity.bprc

```
# FILE: curiosity.bprc
# CONTENT: Bundle Protocol management commands file
# FOR: bpadmin
# NETWORK: Moon-Earth-Mars
# NODE: 10
```

```
# DATE: 2021-05-01
#
# INITIALIZE
1
#
# SCHEME
# [1] scheme name : ipn [Interplanetary Overlay Network Scheme]
# [2] forwarder command : ipnfw [ION Forwarder]
# [3] admin app command : ipnadminep [ION Admin Endpoints]
a scheme ipn 'ipnfw' 'ipnadminep'
#
# ENDPOINT
# [1] node num : 10
# [2] service num : 1 [BP Echo]
# [3] bundle disposition : x [discard undeliverable bundles]
# [4] queue receiver task :
a endpoint ipn:10.1 x
#
# ENDPOINT
# [1] node num : 10
# [2] service num : 2
# [3] bundle disposition : x [discard undeliverable bundles]
# [4] queue receiver task :
a endpoint ipn:10.2 x
#
# ENDPOINT
# [1] node num : 10
# [2] service num : 3
# [3] bundle disposition : x [discard undeliverable bundles]
# [4] queue receiver task :
a endpoint ipn:10.3 x
#
# ENDPOINT
# [1] node num : 10
# [2] service num : 4
# [3] bundle disposition : x [discard undeliverable bundles]
# [4] queue receiver task :
a endpoint ipn:10.4 x
#
# ENDPOINT
```

```
# [1] node num : 10
# [2] service num : 5
# [3] bundle disposition : x [discard undeliverable bundles]
# [4] queue receiver task :
a endpoint ipn:10.5 x
#
# ENDPOINT
# [1] node num : 10
# [2] service num : 63
# [3] bundle disposition : x [discard undeliverable bundles]
# [4] queue receiver task :
a endpoint ipn:10.63 x
#
# ENDPOINT
# [1] node num : 10
# [2] service num : 64 [CFDP Service 1]
# [3] bundle disposition : x [discard undeliverable bundles]
# [4] queue receiver task :
a endpoint ipn:10.64 x
#
# ENDPOINT
# [1] node num : 10
# [2] service num : 6
# [3] bundle disposition : x [discard undeliverable bundles]
# [4] queue receiver task :
a endpoint ipn:10.6 x

# PROTOCOL
# [1] protocol name : udp/tcp
# [2] payload bytes per frame : 1400 (bytes)
# [3] overhead bytes per frame : 100 (bytes)
# [4] nominal data rate : 32000 (bytes per sec)
a protocol udp 1400 100 32000
a protocol tcp 1400 100 32000

# INDUCT_UDP
# [1] udp duct IP: all
# [2] udp duct port: 4556
# [3] udp cli task name: udpcli
a induct udp 0.0.0.0:4556 udpcli
```

```
#
# INDUCT_TCP
# [1] tcp duct IP: all
# [2] tcp duct port: 4556
# [3] tcp cli task name: tcpcli
a induct tcp 0.0.0.0:4556 tcpcli

# OUTDUCT_UDP
# [1] udp duct IP: localhost
# [2] udp clo task name : udpclo
a outduct udp 127.0.0.1:4556 udpclo
#
# OUTDUCT_TCP
# [1] tcp duct IP: 10.20.143.9 ODYSSEY
# [2] tcp clo task name: tcpclo
a outduct tcp 10.20.143.9:4556 tcpclo
#
# OUTDUCT_TCP
# [1] tcp duct IP: 10.20.144.8 MRO
# [2] tcp clo task name: tcpclo
a outduct tcp 10.20.144.8:4556 tcpclo

#
# Select leve of BP watch activities: 0 = None ; 1 = All
w 0

# RUN
# Program:                    ipnadmin
# Conifgurationfile name:     curiosity.ipnrc
r 'ipnadmin curiosity.ipnrc'

# Start all declared schemes and protocols on the local node
s
```

En primer lugar se configura el esquema utilizado en la red DTN. En este caso ipn (*a scheme ipn*). Seguidamente se configuran los *endpoints* que estarán disponibles para las aplicaciones. El siguiente paso es configurar los protocolos que se utilizarán en el nodo, en este caso, UDP y TCP. UDP se utilizará para la comunicación en el mismo nodo y TCP para la comunicación con nodos vecinos. Sigue la configuración con los puertos de escucha (*inducts*) aquí 4556 tanto para

UDP y TCP. La dirección 0.0.0.0 indica todos los adaptadores de red. La configuración de los canales de comunicación para nodos vecinos (*outducts*) indica como vecinos *localhosts*, MRO y Odyssey. Finalmente con el comando *w*, se indica si se reportan notificaciones. El comando *r* inicializa la aplicación *ipnadmin* para el archivo curiosity.ipnrc y el comando *s* inicializa la configuración del archivo en el nodo.

Archivo 5: curiosity.ipnrc

```
# FILE: curiosity.ipnrc
# CONTENT: IPN scheme configuration commands file
# FOR: ipnadmin
# NETWORK: Moon-Earth-Mars
# NODE: 10
# DATE: 2021-05-01
#

# PLAN_UDP
# [1] to node number : 10 MSL
# [2] ip : 127.0.0.1
a plan 10 udp/127.0.0.1:4556

# PLAN_TCP
# [1] to node number : 9 ODYSSEY
# [2] ip: 10.20.143.9
a plan 9 tcp/10.20.143.9:4556

# PLAN:TCP
# [1] to node number : 8 MRO
# [2] ip: 190.20.144.8
a plan 8 tcp/10.20.144.8:4556
```

El archivo .ipnrc identifica los nodos vecinos en el esquema ipn.

El nodo arranca introduciendo los siguientes comandos:

```
ionadmin curiosity.ionrc
```

ionadmin contacts.ionrc

ionsecadmin curiosity.ionsecrc

bpadmin curiosity.bprc

Anexo 5: Graylog

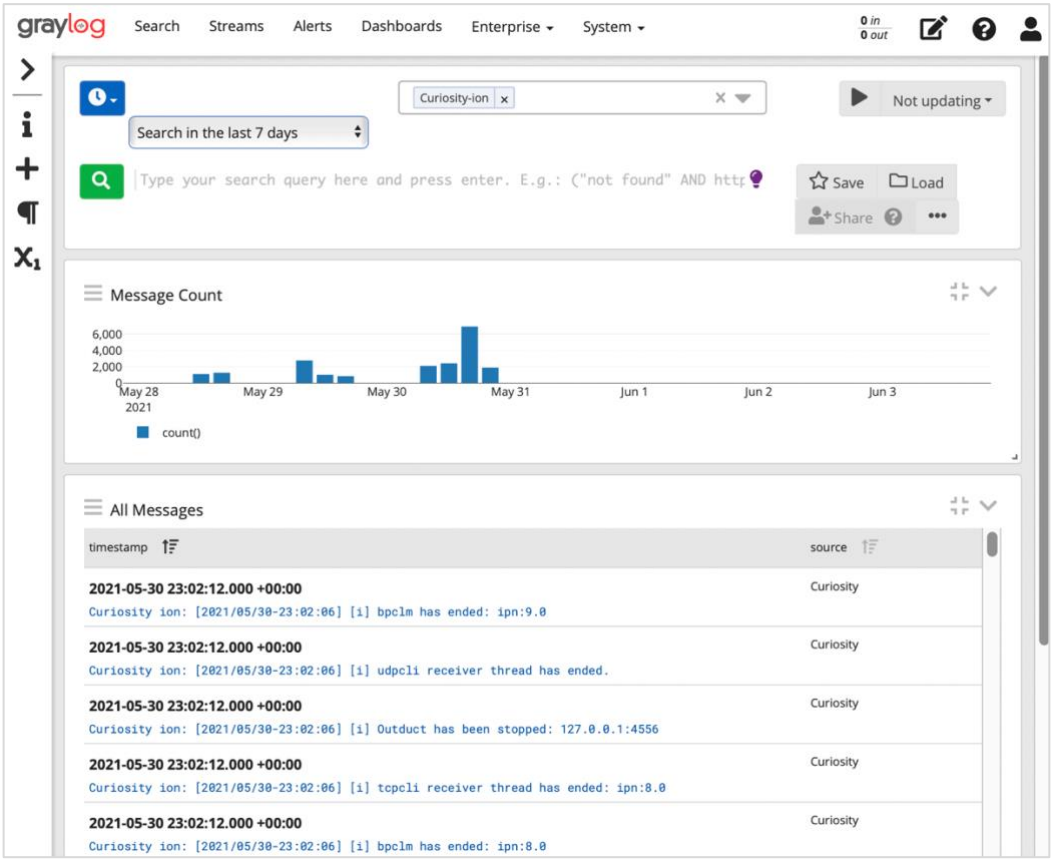
Los logs generados en cada uno de los nodos han sido centralizados en el servidor Graylog. Desde el servidor se pueden realizar búsquedas y filtros de la información almacenada para cada uno de los nodos de la red de pruebas.

Los mensajes se envían desde cada nodo utilizando el servicio *rsyslog*. En la ubicación /etc/rsyslog.d/ se añade el archivo ion.conf con el siguiente contenido:

```
$ModLoad imfile

$InputFilePollInterval 10

$InputFileName /home/user/ionnode/ion.log

$InputFileTag ion:

$InputFileStateFile stat-ion-info

$InputFileSeverity info

$InputFileFacility local3

$InputRunFileMonitor

local3.* @10.20.140.53:10010
```

El archivo de configuración indicará al servicio la vigilancia de una archivo en un intervalo de 10 segundos ubicado en /home/user/ionnode/ con el nombre ion.log. El servicio añadirá la etiqueta ‘ion:’ y establecerá los mensajes con el nivel ‘info’ para finalmente enviar los mensajes al servidor Graylog en la IP 10.20.140.53 y el puerto UDP 10010. En la figura 40 se muestra una captura de pantalla de algunas de las entradas para el nodo MSL Curiosity.

Figura 39: Captura de pantalla del servidor graylog.

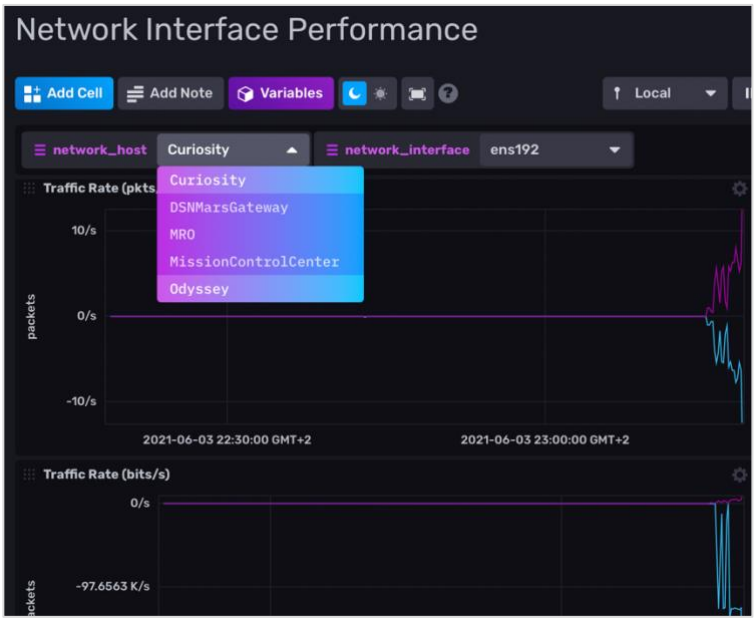


Anexo 6: InfluxDB y Telegraf

El seguimiento del tráfico generado en las redes de los nodos se ha seguido con Telegraf y InfluxDB. Telegraf es un agente que se ejecuta en cada nodo y que envía la información definida en telegraf.conf al servidor InfluxDB. InfluxDB es un servidor de base de datos que permite la consulta de la información y la representación de los datos.

Se ha utilizado una plantilla *Network Interface Performance Monitor*[72] para representar el rendimiento de la red. Permite seleccionar cada uno de los nodos y a su vez cada uno de los interfaces de red de los nodos (figura 41). Los datos representados hacen referencia a la red de transporte IP utilizada por la red DTN. No está representada la tasa de transferencia de los datos exclusivamente generados por la red DTN. Esta cuestión es interesante desarrollarla en futuros trabajos, como herramienta para monitorización de redes DTN.

Figura 40: Captura de pantalla de InfluxDB Network Interface Performance.



[72] https://www.influxdata.com/influxdb-templates/network-interface-performance-monitor/

Anexo 7: DTNperf3

DTNperf3 tiene tres modos de operación: cliente, servidor y monitor. El funcionamiento sigue las pautas de iperf/iperf3. El monitor puede recibir la información de las pruebas y generar un archivo .csv para su posterior análisis de datos.

Por otra parte, existen dos alternativas para el control de congestión: basado en ventanas o basado en tasa de transferencia. Las pruebas realizadas con el control de congestión en ventanas no han funcionado correctamente. En cambio, las pruebas basadas en una tasa de transferencia constante se han ejecutado de forma óptima.

Seguidamente se reproduce un ejemplo de uso, donde el cliente es el nodo que representa MRO como cliente y el nodo DSN como servidor. Ambos nodos están configurados sin modificación en la tasa de transferencia, ni otros parámetros tc.

Servidor:

```
user@DSNMarsGateway:~/dtnperf$ ./dtnperf_vION --server --debug

ION found.

regid 0x0

[DEBUG_L2] choosing bundle destination type... done. Bundles will be saved into [file/memory]: file

[DEBUG_L2] entering infinite loop...

[DEBUG_L2] initiating memory for the next bundle... done

[DEBUG_L2] waiting for the next bundle...

 bundle received

[DEBUG_L2] calculating bundle payload size... done

[DEBUG_L2] marking time... done

Thu Jun  3 22:17:08 2021

 : 1000000 bytes from ipn:8.17912

[DEBUG_L2]      getting bundle header and options... done.

[DEBUG_L2]      getting source eid... done:

        bundle_source_addr = ipn:8.17912
```

```
[DEBUG_L2]      getting destination eid... done:

        bundle_dest_eid =


[DEBUG_L2]      getting reply to eid... done:

        bundle_replyto_eid = dtn:none


[DEBUG_L2]      getting bundle creation timestamp... done:

        bundle creation timestamp:

        secs = 676073815

        seqno= 27


[DTNperf L2] no METADATA blocks specified.

[DEBUG_L2]      getting bundle payload filename... done:

[DEBUG_L1] saved into: /tmp/ion_152125_0

[DEBUG_L2]      checking if this is a file transfer bundle... done.

        bundle is not a file transfer bundle


[DEBUG_L2] initiating memory for the next bundle... done

[DEBUG_L2] waiting for the next bundle...

 bundle received

[DEBUG_L2] calculating bundle payload size... done

[DEBUG_L2] marking time... done

Thu Jun  3 22:17:25 2021

 : 1000000 bytes from ipn:8.17912

[DEBUG_L2]      getting bundle header and options... done.

[DEBUG_L2]      getting source eid... done:

        bundle_source_addr = ipn:8.17912
```

```
[DEBUG_L2]      getting destination eid... done:
        bundle_dest_eid =

[DEBUG_L2]      getting reply to eid... done:
        bundle_replyto_eid = dtn:none

[DEBUG_L2]      getting bundle creation timestamp... done:
        bundle creation timestamp:
        secs = 676073816
        seqno= 28

[DTNperf L2] no METADATA blocks specified.
[DEBUG_L2]      getting bundle payload filename... done:
[DEBUG_L1] saved into: /tmp/ion_152125_0
[DEBUG_L2]      checking if this is a file transfer bundle... done.
        bundle is not a file transfer bundle

...

[DEBUG_L2] initiating memory for the next bundle... done
[DEBUG_L2] waiting for the next bundle...
 bundle received
[DEBUG_L2] calculating bundle payload size... done
[DEBUG_L2] marking time... done
Thu Jun  3 22:17:44 2021
 : 1000000 bytes from ipn:8.17912
[DEBUG_L2]      getting bundle header and options... done.
```

```
[DEBUG_L2]      getting source eid... done:
        bundle_source_addr = ipn:8.17912

[DEBUG_L2]      getting destination eid... done:
        bundle_dest_eid =

[DEBUG_L2]      getting reply to eid... done:
        bundle_replyto_eid = dtn:none

[DEBUG_L2]      getting bundle creation timestamp... done:
        bundle creation timestamp:
        secs = 676073817
        seqno= 29

[DTNperf L2] no METADATA blocks specified.
[DEBUG_L2]      getting bundle payload filename... done:
[DEBUG_L1] saved into: /tmp/ion_152125_0
[DEBUG_L2]      checking if this is a file transfer bundle... done.
        bundle is not a file transfer bundle
[DEBUG_L2] initiating memory for the next bundle... done
[DEBUG_L2] waiting for the next bundle...
```

Para el nodo cliente se utilizan varios parámetros:

-d ipn:7.2000	destino
-D10MB	transmisión de datos: 10Mbytes
-P1MB	cada bundle es de 1 Mbyte
-R1b	ratio de transferencia de 1 bundle por segundo
-l200	tiempo de vida del bundle de 200 segundos

Cliente:

```
user@MRO:~/dtnperf$ dtnperf_vION --client -d ipn:7.2000  -D10MB -P1MB -R1b -l200  --debug

ION found.

[DEBUG_L2] opening connection to local BP daemon...done

[DEBUG_L1] Destination: ipn:7.2000

[DEBUG_L1] Reply-to   : dtn:none


[DEBUG_L1] regid 0x0

[DEBUG_L2] dtnperf header length: 24

[DEBUG_L2] dtnperf payload length: 999976.000000

[DEBUG_L1] Transmission mode = D

      requested Transmission of 10000000.000000 bytes of data

[DEBUG_L1] Transmitting data using = a file

[DEBUG_L1] Congestion control = rate:

      rate is 1.000000 b

[DEBUG_L1] Payload is 1000000.000000 byte

[DEBUG_L1] calculating how many bundles are needed... n_bundles = 10

[DEBUG_L2] creating the bundle object... done

[DEBUG_L2] number of blocks: 0

[DEBUG_L2] entering in the transmission loop

[DEBUG_L2 cong ctrl] congestion control started, mode = R

[DEBUG_L2 send thread] initializing timer... start.tv_sec = 1622758615 s

[DEBUG_L2 send thread] entering loop...

[DEBUG_L2] creating file /tmp/dtnperfbuf_17912... done
```

```
[DEBUG_L2] filling payload... done

[DEBUG_L1] Payload prepared

        [DEBUG_L2 send thread] window = 0

[DEBUG_L1 cong ctrl] wait time for each bundle: 1.0000 s

[DEBUG_L2 cong ctrl] increased free window size = 1

passing the bundle to BP...

[DTNperf L2] printing BUNDLE...

[DTNperf L2] id-creation timestamp: 676073815   sequence number: 27

[DTNperf L2] id-source: ipn:8.17912

[DTNperf L2] id-original length: 0      fragmentation offset: 0

[DTNperf L2] spec-creation timestamp: 0 sequence number: 0

                expiration time: 200

[DTNperf L2] spec-delivery register id: 0       delivery options: 0

[DTNperf L2] spec-source: ipn:8.17912   dest:   replyto:

[DTNperf L2] spec-is critical:  priority: 1,0   is unreliable:

[DTNperf L2] spec-flow label: 0

[DTNperf L1] no METADATA blocks specified.

[DTNperf L1] no EXTENSION blocks specified.

[DTNperf L2] done.

bundle passed to BP

        [DEBUG_L2 send thread] bundle timestamp: 676073815.27

        [DEBUG_L2 send thread] now bundle passed is 1

        [DEBUG_L2 send thread] window = 0


...

[DEBUG_L2 cong ctrl] increased free window size = 1

passing the bundle to BP...

[DTNperf L2] printing BUNDLE...
```

```
[DTNperf L2] id-creation timestamp: 676073824   sequence number: 36

[DTNperf L2] id-source: ipn:8.17912

[DTNperf L2] id-original length: 0      fragmentation offset: 0

[DTNperf L2] spec-creation timestamp: 0 sequence number: 0

             expiration time: 200

[DTNperf L2] spec-delivery register id: 0       delivery options: 0

[DTNperf L2] spec-source: ipn:8.17912   dest:   replyto:

[DTNperf L2] spec-is critical:  priority: 1,0   is unreliable:

[DTNperf L2] spec-flow label: 0

[DTNperf L1] no METADATA blocks specified.

[DTNperf L1] no EXTENSION blocks specified.

[DTNperf L2] done.

bundle passed to BP

      [DEBUG_L2 send thread] bundle timestamp: 676073824.36

      [DEBUG_L2 send thread] now bundle passed is 10

[DEBUG_L2 send thread] ...out from sending loop

[DEBUG_L2] out from loop

[DEBUG_L2] getting total end-time... end.tv_sec = 1622758624 s


Bundles sent = 10 total data sent = 10.000 Mbyte

Total execution time = 9.5

Throughput = 8.418 Mbit/s

[DEBUG_L1] Bundle stop not needed, mon_eid = dtn:none

[DEBUG_L2] closing DTN handle... done

[DEBUG_L2] removed file /tmp/dtnperfbuf_17912
```

Anexo 8: plan de contactos

El plan de contactos define los intervalos de tiempo en lo que existen un contacto entre vecinos. En este anexo se muestra el plan de contactos principal para las pruebas realizadas.

```
# FILE: contacts.ionrc

# CONTENT: ION network contacts commands file

# FOR: ionadmin

# NETWORK: Mars-Earth-Moon

# DATE: 2021-05-01

#

# CONTACT_REL_REL_TIME

# [1] relative start time

# [2] relative stop time

# [3] source node

# [4] destination node

# [5] transmission data rate (bytes per sec)

# [6] contact confidence (fraction, optional)

# a a contact +0 +1000 1 2 1000000



# Host                Location        Node           Test/Emu

# ------------------------------------------------------------

# MBASE               (Moon)          node 1         CORE

# MLAB                (Moon)          node 2         CORE

# MGATEWAY            (Moon)          node 3         CORE

# MSAT                (Moon)          node 4         CORE

# DSNMOONGATEWAY      (Earth)         node 5         CORE

# MCC                 (Earth)         node 6         Testbed
```

```
# DSNMARSGATEWAY        (Earth)        node 7         Testbed

# MRO                   (Mars)         node 8         Testbed

# ODYSSEY               (Mars)         node 9         Testbed

# MSL                   (Mars)         node 10        Testbed



# Test time:  24m --> 1440s


#

#       RANGES

#


# For simplicity, OWLT is always 1s

a range +0 +1440 1 1 1

a range +0 +1440 2 2 1

a range +0 +1440 3 3 1

a range +0 +1440 4 4 1

a range +0 +1440 5 5 1

a range +0 +1440 6 6 1

a range +0 +1440 7 7 1

a range +0 +1440 8 8 1

a range +0 +1440 9 9 1

a range +0 +1440 10 10 1


# MBASE [node 1] <-> MLAB [node 2] (Moon-Net OWLT 10ms)

a range +0 +1440 1 2 0.01

a range +0 +1440 2 1 0.01
```

```
# MBASE [node 1] <-> MGATEWAY [node 3] (Moon-Net OWLT 10ms)

a range +0 +1440 1 3 0.01

a range +0 +1440 3 1 0.01


# MGATEWAY [node 3] <-> MSAT [node 4] (Moon-Net OWLT 500ms --> 1s for simplicity)

a range +0 +1440 3 4 1

a range +0 +1440 4 3 1


# MSAT [node 4] <-> DSNMOONGATEWAY [node 5] (Moon-Earth OWLT 1.36s --> 1s for simplicity)

a range +0 +1440 4 5 1

a range +0 +1440 5 4 1


# DSNMOONGATEWAY [node 5] <-> MCC [node 6] (Earth-Net 100ms)

a range +0 +1440 6 5 0.1

a range +0 +1440 5 6 0.1


# DSNMOONGATEWAY [node 5] <-> DSNMARSGATEWAY [node 7] (Earth-Net 100ms)

a range +0 +1440 5 7 0.1

a range +0 +1440 7 5 0.1


# DSNMARSGATEWAY [node 7] <-> MCC [node 6] (Earth-Net 100ms)

a range +0 +1440 7 6 0.1

a range +0 +1440 6 7 0.1


# MRO [node 8] <-> DSNMARSGATEWAY [node 7] (Mars-Earth OWLT at less 360s --> 10s for simplicity)

a range +0 +1440 7 8 10

a range +0 +1440 8 7 10
```

ODYSSEY [node 9] <-> DSNMARSGATEWAY [node 7] (Mars-Earth OWLT at less 360s --> 10s for simplicity)

a range +0 +1440 7 9 10

a range +0 +1440 9 7 10

MSL [node 10] <-> MRO [node 8] (Mars-Net delay 1s)

a range +0 +1440 8 10 1

a range +0 +1440 10 8 1

MSL [node 10] <-> ODYSSEY [node 9] (Mars-Net delay 1s)

a range +0 +1440 9 10 1

a range +0 +1440 10 9 1

#

CONTACTS

#

Nodes are always connected to themselves

[node 1] MBASE rate

a contact +0 +1440 7 7 1250000

[node 2] MLAB rate

a contact +0 +1440 8 8 1250000

[node 3] MGATEWAY rate

a contact +0 +1440 3 3 1250000

[node 4] MSAT rate

a contact +0 +1440 6 6 62500

[node 5] DSNMOONGATEWAY rate

```
a contact +0 +1440 5 5 1250000

# [node 6] MCC rate

a contact +0 +1440 6 6 1250000

# [node 7] DSNMARSGATEWAY rate

a contact +0 +1440 7 7 1250000

# [node 8] MRO rate

a contact +0 +1440 8 8 125000

# [node 9] ODYSSEY rate

a contact +0 +1440 9 9 15000

# [node 10] MSL rate

a contact +0 +1440 10 10 32000



# MSL [node 10] -> MRO [node 8] (rate 128 kbps -> 16000 bytes per second)

a contact +467 +475 10 8 16000

a contact +1193 +1202 10 8 16000



# MRO [node 8 -> MSL [node 10] (rate 32 kbps -> 4000 bytes per second)

a contact +467 +475 8 10 4000

a contact +1193 +1202 8 10 4000


# MSL [node 10] -> ODYSSEY [node 9] (rate 64 kbps -> 8000 bytes per second)

a contact +596 +599 10 9 8000

a contact +709 +719 10 9 8000

a contact +1365 +1376 10 9 8000


# ODYSSEY [node 9] -> MSL [node 10] (rate 16 kbps -> 2000 bytes per second)
```

```
a contact +596 +599 9 10 2000

a contact +709 +719 9 10 2000

a contact +1365 +1376 9 10 2000


# MRO [node 8] -> DSNMARSGATEWAY [node 7] (rate 500 kbps -> 62500 bytes per second)

a contact +49 +90 8 7 62500

a contact +161 +202 8 7 62500

a contact +273 +315 8 7 62500

a contact +386 +427 8 7 62500

a contact +498 +539 8 7 62500

a contact +610 +651 8 7 62500

a contact +722 +763 8 7 62500

a contact +834 +876 8 7 62500

a contact +946 +988 8 7 62500

a contact +1059 +1100 8 7 62500

a contact +1171 +1212 8 7 62500

a contact +1283 +1324 8 7 62500

a contact +1395 +1437 8 7 62500


# DSNMARSGATEWAY [node 7] -> MRO [node 8]  (rate 128 kbps -> 16000 bytes per second)

a contact +49 +90 7 8 16000

a contact +161 +202 7 8 16000

a contact +273 +315 7 8 16000

a contact +386 +427 7 8 16000

a contact +498 +539 7 8 16000

a contact +610 +651 7 8 16000

a contact +722 +763 7 8 16000

a contact +834 +876 7 8 16000
```

```
a contact +946 +988 7 8 16000

a contact +1059 +1100 7 8 16000

a contact +1171 +1212 7 8 16000

a contact +1283 +1324 7 8 16000

a contact +1395 +1437 7 8 16000


# ODYSSEY [node 9] -> DSNMARSGATEWAY [node 7] (rate 128 kbps -> 16000 bytes per second)

a contact +70 +92 9 7 16000

a contact +188 +210 9 7 16000

a contact +307 +329 9 7 16000

a contact +425 +447 9 7 16000

a contact +544 +566 9 7 16000

a contact +662 +684 9 7 16000

a contact +781 +803 9 7 16000

a contact +899 +922 9 7 16000

a contact +1018 +1040 9 7 16000

a contact +1137 +1159 9 7 16000

a contact +1255 +1277 9 7 16000

a contact +1374 +1396 9 7 16000


# DSNMARSGATEWAY [node 7] -> ODYSSEY [node 9] (rate 16 kbps -> 2000 bytes per second)

a contact +70 +92 7 9 2000

a contact +188 +210 7 9 2000

a contact +307 +329 7 9 2000

a contact +425 +447 7 9 2000

a contact +544 +566 7 9 2000

a contact +662 +684 7 9 2000

a contact +781 +803 7 9 2000
```

a contact +899 +922 7 9 2000

a contact +1018 +1040 7 9 2000

a contact +1137 +1159 7 9 2000

a contact +1255 +1277 7 9 2000

a contact +1374 +1396 7 9 2000

DSNMARSGATEWAY [node 7] -> MCC [node 6] (rate 10 Mbps -> 1250000 bytes per second)

a contact +0 +1440 7 6 1250000

MCC [node 6] -> DSNMARSGATEWAY [node 7] (rate 10 Mbps -> 1250000 bytes per second)

a contact +0 +1440 6 7 1250000

DSNMOONGATEWAY [node 5] -> DSNMARSGATEWAY [node 7] (rate 10 Mbps -> 125000 bytes per second)

a contact +0 +1440 5 7 1250000

DSNMARSGATEWAY [node 7] -> DSNMOONGATEWAY [node 5] (rate 10 Mbps -> 125000 bytes per second)

a contact +0 +1440 7 5 1250000

MSAT [node 4] -> DSNMOONGATEWAY [node 5] (rate 500 kbps -> 62500 bytes per second)

a contact +70 +80 4 5 62500

a contact +150 +160 4 5 62500

a contact +230 +240 4 5 62500

a contact +310 +320 4 5 62500

a contact +390 +400 4 5 62500

a contact +470 +480 4 5 62500

a contact +550 +560 4 5 62500

a contact +630 +640 4 5 62500

```
a contact +710 +720 4 5 62500

a contact +790 +800 4 5 62500

a contact +870 +880 4 5 62500

a contact +950 +960 4 5 62500

a contact +1030 +1040 4 5 62500

a contact +1110 +1120 4 5 62500

a contact +1190 +1200 4 5 62500

a contact +1270 +1280 4 5 62500

a contact +1350 +1360 4 5 62500

a contact +1430 +1440 4 5 62500


# DSNMOONGATEWAY [node 5] -> MSAT [node 4] (rate 500 kbps -> 62500 bytes per second)

a contact +70 +80 5 4 62500

a contact +150 +160 5 4 62500

a contact +230 +240 5 4 62500

a contact +310 +320 5 4 62500

a contact +390 +400 5 4 62500

a contact +470 +480 5 4 62500

a contact +550 +560 5 4 62500

a contact +630 +640 5 4 62500

a contact +710 +720 5 4 62500

a contact +790 +800 5 4 62500

a contact +870 +880 5 4 62500

a contact +950 +960 5 4 62500

a contact +1030 +1040 5 4 62500

a contact +1110 +1120 5 4 62500

a contact +1190 +1200 5 4 62500

a contact +1270 +1280 5 4 62500
```

```
a contact +1350 +1360 5 4 62500

a contact +1430 +1440 5 4 62500


# MSAT [node 4] -> MGATEWAY [node 3] (rate 500 kbps -> 62500 bytes per second)

a contact +20 +40 4 3 62500

a contact +100 +120 4 3 62500

a contact +140 +160 4 3 62500

a contact +220 +240 4 3 62500

a contact +260 +280 4 3 62500

a contact +340 +360 4 3 62500

a contact +380 +400 4 3 62500

a contact +460 +480 4 3 62500

a contact +500 +520 4 3 62500

a contact +580 +600 4 3 62500

a contact +620 +640 4 3 62500

a contact +700 +720 4 3 62500

a contact +740 +760 4 3 62500

a contact +820 +840 4 3 62500

a contact +860 +880 4 3 62500

a contact +940 +960 4 3 62500

a contact +980 +1000 4 3 62500

a contact +1060 +1080 4 3 62500

a contact +1100 +1120 4 3 62500

a contact +1180 +1200 4 3 62500

a contact +1220 +1240 4 3 62500

a contact +1300 +1320 4 3 62500

a contact +1340 +1360 4 3 62500

a contact +1420 +1440 4 3 62500
```

MGATEWAY [node 3] -> MSAT [node 4] (rate 500 kbps -> 62500 bytes per second)

a contact +20 +40 3 4 62500

a contact +100 +120 3 4 62500

a contact +140 +160 3 4 62500

a contact +220 +240 3 4 62500

a contact +260 +280 3 4 62500

a contact +340 +360 3 4 62500

a contact +380 +400 3 4 62500

a contact +460 +480 3 4 62500

a contact +500 +520 3 4 62500

a contact +580 +600 3 4 62500

a contact +620 +640 3 4 62500

a contact +700 +720 3 4 62500

a contact +740 +760 3 4 62500

a contact +820 +840 3 4 62500

a contact +860 +880 3 4 62500

a contact +940 +960 3 4 62500

a contact +980 +1000 3 4 62500

a contact +1060 +1080 3 4 62500

a contact +1100 +1120 3 4 62500

a contact +1180 +1200 3 4 62500

a contact +1220 +1240 3 4 62500

a contact +1300 +1320 3 4 62500

a contact +1340 +1360 3 4 62500

a contact +1420 +1440 3 4 62500

MGATEWAY [node 3] -> MBASE [node 1] (rate 10 Mbps -> 1250000 bytes per second)

```
a contact +0 +1440 3 1 1250000

# MBASE [node 1] -> MGATEWAY [node 3] (rate 10 Mbps -> 1250000 bytes per second)
a contact +0 +1440 1 3 1250000

# MBASE [node 1] -> MLAB [node 2] (rate 10 Mbps -> 1250000 bytes per second)
a contact +0 +1440 1 2 1250000

# MLAB [node 2] -> MBASE [node 1] (rate 10 Mbps -> 1250000 bytes per second)
a contact +0 +1440 2 1 1250000
```

Bibliografía

[1] Bryce Space and Technology, “Smallsats by the Numbers 2020”, 2020. [En línea]. Disponible en: https://brycetech.com/reports.

[2] NASA Ames Research Center, “Small Sats State of the Art”, 2020. [En línea]. Disponible en: https://www.nasa.gov/sites/default/files/atoms/files/2020soa_final.pdf.

[3] Bryce Space and Technology, “Bryce 2019 Global Space Economy”, 2020. https://brycetech.com/reports/report-documents/Bryce_2019_Global_Space_Economy.png. (acceso: 1 de mayo de 2021).

[4] P. Muri, “Delay Tolerant Networking for cubesat topologies and platforms”, Tesis doctoral, University of Florida, 2013. [En línea]. Disponible en: https://ufdc.ufl.edu/UFE0045864/00001.

[5] I. Komnios (SPICE), “SPICE : Space Internetworking Center Activity Summary”, 2014. [En línea]. Disponible en: http://www.spice-center.org/.

[6] The Consultative Committee for Space Data Systems, “Solar System Internetwork (SSI) Architecture,” no. July, 2014, [En línea]. Disponible en: https://evt.grc.nasa.gov/rfp-industry-briefing-2016/wp-content/blogs.dir/52/files/sites/10/CCSDS-730.1-G-1-Solar-System-Internetwork-SSI-Architecture.pdf.

[7] A. Alhilal, T. Braud, and P. Hui, “Future Architecture of the Interplanetary Internet”, October, 2018, [Online]. Disponible en: http://arxiv.org/abs/1810.01093.

[8] C. J. Krupiarz *et al.*, “Enabling the interplanetary internet,” *Johns Hopkins APL Tech. Dig. Applied Phys. Lab.*, vol. 30, no. 2, pp. 122–134, 2011. [En línea]. Disponible en: https://www.jhuapl.edu/Content/techdigest/pdf/V30-N02/30-02-Krupiarz.pdf.

[9] K. Fall, “A delay-tolerant network architecture for challenged internets,” *Comput. Commun. Rev.*, vol. 33, pp. 27–34, 2003, doi: 10.1145/863955.863960. [En línea]. Disponible en: http://kfall.com/papers/p27-fall.pdf.

[10] J. Kaiser, “Interplanetary Internet,” *Science*, vol. 281, no. 5379, p. 879, Aug. 1998, [En línea]. Disponible en: https://search.proquest.com/scholarly-journals/interplanetary-internet/docview/213564635/se-2?accountid=15299.

[11] ICANN, “Las funciones de la IANA”, p. 22, 2015, [En línea]. Disponible en: https://www.icann.org/es/system/files/files/iana-functions-18dec15-es.pdf.

[12] L. Torgerson *et al.*, “Delay-Tolerant Networking Architecture”, no. 4838. RFC Editor, Apr. 2007, doi: 10.17487/RFC4838. [En línea]. Disponible en: https://www.rfc-editor.org/rfc/rfc4838.html.

[13] K. Scott and S. C. Burleigh, “Bundle Protocol Specification”, no. 5050. RFC Editor, Nov. 2007, doi: 10.17487/RFC5050. [En línea]. Disponible en: https://www.rfc-editor.org/rfc/rfc5050.html.

[14] CCSDS, “Bundle Protocol Specification”, *Blue Book*, 2015. [En línea], Disponible en: https://public.ccsds.org/Pubs/734x2b1.pdf.

[15] CCSDS, “Licklider Transmission Protocol for CCSDS”, *Blue Book*, 2015. [En línea], Disponible en: https://public.ccsds.org/Pubs/734x1b1.pdf.

[16] J. F. Kurose and K. W. Ross, *Computer Networking: A Top-Down Approach*, 6th ed. Boston, MA: Pearson, 2013.

[17] A. Pereira da Silva, S. Burleigh, K. Obraczka, *Delay and Disruption Tolerant Networks: Interplanetary and Earth-Bound Architecture, protocols and applications*, 1st ed. CRC Press, 2020.

[18] CCSDS, “Schedule-aware bundle routing”, *Blue Book*, 2019. [En línea], Disponible en: https://public.ccsds.org/Pubs/734x3b1.pdf.

[19] F. Warthman, “Delay- and Disruption-Tolerant Networks (DTNs) - A Tutorial”, *Ipnsig*, pp. 1–35, 2015, [En línea]. Disponible en: http://ipnsig.org/wp-content/uploads/2015/09/DTN_Tutorial_v3.2.pdf.

[20] S. Farrell, H. Weiss, S. Symington, and P. Lovell, “Bundle Security Protocol Specification”, no. 6257. RFC Editor, May 2011, doi: 10.17487/RFC6257. [En línea], Disponible en: https://www.rfc-editor.org/rfc/rfc6257.html.

[21] E. J. Birrane and K. McKeever, “Bundle Protocol Security Specification”, Internet Engineering Task Force, Feb. 2021. [En línea]. Disponible en: https://datatracker.ietf.org/doc/html/draft-ietf-dtn-bpsec-27.

[22] S. Burleigh, K. Fall, and E. J. Birrane, “Bundle Protocol Version 7”, Internet Engineering Task Force, Jan. 2021. [En línea]. Disponible en: https://datatracker.ietf.org/doc/html/draft-ietf-dtn-bpbis-31.

[23] W. Eddy and E. B. Davies, “Using Self-Delimiting Numeric Values in Protocols”, no. 6256. RFC Editor, May 2011, doi: 10.17487/RFC6256. [En línea]. Disponible en: https://www.rfc-editor.org/rfc/rfc6256.html.

[24] C. Bormann and P. E. Hoffman, “Concise Binary Object Representation (CBOR)”, no. 8949. RFC Editor, Dec. 2020, doi: 10.17487/RFC8949. [En línea]. Disponible en: https://www.rfc-editor.org/rfc/rfc8949.html.

[25] S. C. Burleigh, S. Farrell, and M. Ramadas, “Licklider Transmission Protocol - Specification”, no. 5326. RFC Editor, Sep. 2008, doi: 10.17487/RFC5326. [En línea]. Disponible en: https://www.rfc-editor.org/rfc/rfc5326.html.

[26] M. Demmer, J. Ott, and S. Perreault, “Delay-Tolerant Networking TCP Convergence-Layer Protocol”, no. 7242. RFC Editor, Jun. 2014, doi: 10.17487/RFC7242. [En línea]. Disponible en: https://www.rfc-editor.org/rfc/rfc7242.html.

[27] B. Sipos, M. Demmer, J. Ott, and S. Perreault, “Delay-Tolerant Networking TCP Convergence Layer Protocol Version 4”, Internet Engineering Task Force, Feb. 2021. [En línea]. Disponible en: https://datatracker.ietf.org/doc/html/draft-ietf-dtn-tcpclv4-26.

[28] S. C. Burleigh, “Minimal TCP Convergence-Layer Protocol,” Internet Engineering Task Force, Apr. 2019. [En línea]. Disponible en: https://datatracker.ietf.org/doc/html/draft-ietf-dtn-mtcpcl-01.

[29] S. C. Burleigh, “Minimal TCP Convergence-Layer Protocol,” Internet Engineering Task Force, Apr. 2019. [En línea]. Disponible en: https://datatracker.ietf.org/doc/html/draft-ietf-dtn-mtcpcl-01.

[30] S. Burleigh, “Contact Graph Routing,” Internet Engineering Task Force, Jul. 2010. [En línea]. Disponible en: https://datatracker.ietf.org/doc/html/draft-burleigh-dtnrg-cgr-01.

[31] J. A. Fraire, O. De Jonckère, and S. C. Burleigh, “Routing in the Space Internet: A contact graph routing tutorial”, *J. Netw. Comput. Appl.*, vol. 174, no. December 2020, p. 102884, 2021, doi: 10.1016/j.jnca.2020.102884.

[32] CCSDS, “Schedule-Aware Bundle Routing”, *Blue Book*, 2019. [En línea], Disponible en: https://public.ccsds.org/Pubs/734x3b1.pdf.

[33] J.A. Fraire., “Enrutando la Internet espacial con Contact Graph Routing”, IEEE COMSOC Argentina Open Webinar. Buenos Aires, Argentina, 2020. [En línea]. Disponible en: https://argentina.chapters.comsoc.org/enrutando-la-internet-espacial-con-contact-graph-routing/.

[34] P. G. Madoery, “Congestión y Escalabilidad en Redes Espaciales”, IEEE COMSOC Argentina Open Webinar. Buenos Aires, Argentina, 2020. [En línea]. Disponible en: https://argentina.chapters.comsoc.org/congestion-y-escalabilidad-en-redes-espaciales/.

[35] G. Araniti *et al.*, “Contact graph routing in DTN space networks: Overview, enhancements and performance”, *IEEE Commun. Mag.*, vol. 53, no. 3, pp. 38–46, 2015, doi: 10.1109/MCOM.2015.7060480. [En línea]. Disponible en: https://ieeexplore.ieee.org/document/7060480.

[36] A. Lindgren, A. Doria, E. B. Davies, and S. Grasic, “Probabilistic Routing Protocol for Intermittently Connected Networks”, no. 6693. RFC Editor, Aug. 2012, doi: 10.17487/RFC6693. [En línea]. Disponible en: https://www.rfc-editor.org/rfc/rfc6693.html.

[37] S. Pal and S. Misra, “Contact-based routing in DTNs”, *ACM IMCOM 2015 - Proc.*, 2015, doi: 10.1145/2701126.2701145. [En línea]. Disponible en: https://dl.acm.org/doi/10.1145/2701126.2701145.

[38] M. Madoery, “Modelos, Algoritmos y Protocolos para Redes de Comunicaciones Tolerantes a Interrupciones con Alto Grado de Predecibilidad”, Tesis doctoral, University Nacional de Córdoba, Argentina, 2019. [En línea]. Disponible en: https://rdu.unc.edu.ar/handle/11086/11460.

[39] M. Demmer, E. Brewer, K. Fall, M. Ho, and R. Patra, “Implementing Delay Tolerant Networking,” 2004, [Online]. Available: http://dtnrg.org/docs/docs/papers/demmer-irb-tr-04-020.pdf.

[40] M. Feldmann and F. Walter, “µpCN - A bundle protocol implementation for microcontrollers”, *2015 Int. Conf. Wirel. Commun. Signal Process. WCSP 2015*, 2015, doi: 10.1109/WCSP.2015.7341252. [En línea]. Disponible en: https://ieeexplore.ieee.org/document/7341252.

[41] A. Penning, L. Baumgärtner, J. Höchst, A. Sterz, M. Mezini, and B. Freisleben, “DTN7: An Open-Source Disruption-Tolerant Networking Implementation of Bundle Protocol 7”, *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 11803 LNCS, pp. 196–209, 2019, doi: 10.1007/978-3-030-31831-4_14. [En línea]. Disponible en: https://arxiv.org/abs/1908.10237.

[42] A. Jenkins, S. Kuzminsky, K. K. Gifford, R. L. Pitts, and K. Nichols, “Delay/disruption-tolerant networking: Flight test results from the international space station”, *IEEE Aerosp. Conf. Proc.*, 2010, doi: 10.1109/AERO.2010.5446948. [En línea]. Disponible en: https://ieeexplore.ieee.org/document/5446948.

[43] A. Schlesinger, B. M. Willman, L. Pitts, S. R. Davidson, and W. A. Pohlchuck, “Delay/Disruption Tolerant Networking for the International Space Station (ISS)”, *IEEE Aerosp. Conf. Proc.*, 2017, doi: 10.1109/AERO.2017.7943857. [En línea]. Disponible en: https://ieeexplore.ieee.org/document/7943857.

[44] S. Grasic, “Development and Deployment of Delay Tolerant Networks: An Artic Village Case”, Tesis doctoral, Lulea University of Technology, Suecia, 2014. [En línea]. Disponible en: http://ltu.diva-portal.org/smash/get/diva2:989908/FULLTEXT01.pdf.

[45] NASA, “Lunar Laser Communication Demonstration NASA’s First Space Laser Communication System Demonstration”, 2013, [En línea]. Disponible en: https://www.nasa.gov/sites/default/files/llcdfactsheet.final_.web_.pdf.

[46] D. Israel, B. Edwards, J. Hayes, W. Knopf, A. Robles, and L. Braatz, “The benefits of delay/disruption tolerant networking (DTN) for future NASA science missions”, *Proc. Int. Astronaut. Congr. IAC*, vol. 2019-October, pp. 21–25, 2019. [En línea]. Disponible en: https://www.nasa.gov/sites/default/files/atoms/files/the_benefits_of_dtn_for_future_nasa_science_missions.pdf.

[47] C. Caini, A. D’Amico, and M. Rodolfi, “DTNperf-3 at work: Aims and use”, *Proc. Annu. Int. Conf. Mob. Comput. Networking, MOBICOM*, pp. 53–55, 2013, doi: 10.1145/2505494.2505508. [En línea]. Disponible en: https://dl.acm.org/doi/10.1145/2505494.2505508.

[48] C. Caini, A. Amico, and M. Rodolfi, “DTNperf _ 3 : a Further Enhanced Tool for Delay- / Disruption- Tolerant Networking Performance,” pp. 3009–3015, 2021. [En línea]. Disponible en: https://ieeexplore.ieee.org/document/6831533.

[49] J. Taylor, “Deep space communications”, Descanso, Jet Propulsion Laboratory, California Institute of Technology, 2014. [En línea]. Disponible en: https://descanso.jpl.nasa.gov/monograph/mono.html.

[50] J. Taylor, D. Lee, S. Shambayati, “Mars Reconnaissance Orbiter Telecommunications”, Descanso, Jet Propulsion Laboratory, California Institute of Technology, 2006. [En línea]. Disponible en: https://descanso.jpl.nasa.gov/DPSummary/MRO_092106.pdf.

[50] A. Makovsky, Pl Ilott, J. Taylor, “Mars Science Laboratory Telecommunications System Design”, Descanso, Jet Propulsion Laboratory, California Institute of Technology, 2009. [En línea]. Disponible en: https://descanso.jpl.nasa.gov/DPSummary/Descanso14_MSL_Telecom.pdf.

[51] Al Makovsky, A. Barbieri, R. Tung, “Odyssey Telecommunications”, Descanso, Jet Propulsion Laboratory, California Institute of Technology, 2002. [En línea]. Disponible en: https://descanso.jpl.nasa.gov/DPSummary/odyssey_telecom.pdf.

[52] S. K. Sharma, S. Chatzinotas, and B. Ottersten, “Mars to Earth communications through orbiters”, *2013 Futur. Netw. Mob. Summit, Futur. 2013*, 2013, pp. 127–140, 2013, doi: 10.1002/sat. [En línea]. Disponible en: https://www.researchgate.net/publication/259542825_Mars_to_Earth_communications_through_orbiters_Delay-TolerantDisruption-Tolerant_Networking_performance_analysis.

[53] J. A. Fraire and J. M. Finochietto, “Design challenges in contact plans for disruption-tolerant satellite networks”, *IEEE Commun. Mag.*, vol. 53, no. 5, pp. 163–169, 2015, doi: 10.1109/MCOM.2015.7105656. [En línea]. Disponible en: https://ieeexplore.ieee.org/document/7105656.

[54] J. A. Fraire, “Introducing contact plan designer: A planning tool for dtn-based space-terrestrial networks”, *Proc. - 6th IEEE Int. Conf. Sp. Mission Challenges Inf. Technol. SMC-IT 2017*, vol. 2017-Decem, no. September 2017, pp. 124–127, 2017, doi: 10.1109/SMC-IT.2017.28. [En línea]. Disponible en: https://ieeexplore.ieee.org/document/8227551.

[55] C. Caini and V. Fiore, “Moon to earth DTN communications through lunar relay satellites”, *2012 6th Adv. Satell. Multimed. Syst. Conf. ASMS 2012 12th Signal Process. Sp. Commun. Work. SPSC 2012*, pp. 89–95, 2012, doi: 10.1109/ASMS-SPSC.2012.6333112. [En línea]. Disponible en: https://ieeexplore.ieee.org/document/6333112.