

Universidad Nacional de Córdoba

Facultad de Ciencias Exactas Físicas y Naturales

Doctorado en Ciencias de la Ingeniería

Tesis Doctoral



Modelos, Algoritmos y Protocolos para Redes de Comunicaciones Tolerantes a Interrupciones con Alto Grado de Predecibilidad

Autor: Ing. Pablo G. Madoery

Director: Dr. Jorge M. Finochietto

Marzo 2019

Modelos, Algoritmos y Protocolos para Redes de Comunicaciones
Tolerantes a Interrupciones con Alto Grado de Predecibilidad

por

Ing. Pablo G. Madoery

Dr. Jorge M. Finochietto

Director

COMISIÓN ASESORA

Dr. Jorge M. Finochietto

UNC-CONICET

Dra. Graciela Corral Briones

UNC-CONICET

Dr. Pablo A. Ferreyra

UNC

Esta Tesis fue enviada a la Facultad de Ciencias Exactas Físicas y Naturales de la Universidad Nacional de Córdoba para cumplimentar los requerimientos de obtención del grado académico de Doctor en Ciencias de la Ingeniería.

Córdoba, Argentina



UNIVERSIDAD NACIONAL DE CORDOBA
Facultad de Cs. Exactas, Físicas y Naturales

ACTA DE EXAMENES

Libro: 00001 Acta: 04649 Hoja 01/01

LLAMADO: 1 18/03/2019

CATEDRA - MESA:

DI002 TESIS DOCTORADO EN CIENCIAS DE LA INGENIERIA

NÚMERO	APELLIDO Y NOMBRE	DOCUMENTO	INGRESO	COND.	NOTA	FIRMA
33106097	MADOERY, PABLO GUSTAVO	DNI: 33106097	2014	T	Aprobado	

MICOLINI, ORLANDO - AGUIRRE, Nazareno - CASTIÑEIRA, Jorge - REY VEGA, Leonardo - CORRAL BRIONES

Observaciones:

Córdoba, ___/___/___.

Certifico que la/s firma/s que ha/n sido puesta/s en la presente Acta pertenece/n a: Graciela Corral Briones

1	0	1	0	1
Inscriptos	Ausentes	Examinados	Reprobados	Aprobados

13/03/2019 13:44:16

Libro/Acta: 0000104649 Hoja: 01/01

*“Si quieres construir un barco
no empieces por buscar madera, cortar tablas o distribuir el trabajo;
primero has de evocar en los hombres y mujeres el anhelo de la
interminable inmensidad del mar”*

Antoine de Saint-Exupéry

UNIVERSIDAD NACIONAL DE CÓRDOBA

Resumen en Español

Universidad Nacional de Córdoba
Facultad de Ciencias Exactas Físicas y Naturales

Doctorado en Ciencias de la Ingeniería

Modelos, Algoritmos y Protocolos para Redes de Comunicaciones Tolerantes a Interrupciones con Alto Grado de Predecibilidad

por Pablo Gustavo

MADOERY

La industria de las comunicaciones satelitales ha mostrado un avance limitado en las últimas décadas en comparación con la evolución de las redes terrestres tales como Internet. La razón principal consiste en que el entorno espacial es radicalmente diferente al terrestre, lo cual impacta considerablemente en la estabilidad de las conexiones y en el hecho de que los protocolos de comunicación utilizados en Tierra resulten inadecuados y/o ineficientes cuando se tratan de adaptar al espacio. Sin embargo, recientemente, y producto de un esfuerzo conjunto de diferentes agencias espaciales (NASA, ESA, CONAE, etc), se ha comenzado a estudiar y experimentar con estrategias de comunicaciones en red que son capaces de tolerar retardos e interrupciones mediante un cambio paradigmático en la forma de realizar dichas comunicaciones. En particular, se ha propuesto una arquitectura de protocolos llamada *Delay/Disruption Tolerant Networking (DTN)*, y que dada la predecibilidad de las trayectorias satelitales, permite aprovechar un plan de contactos compuesto por las oportunidades de comunicación en el futuro, lo que permite a los satélites tomar decisiones eficientes sobre cómo y cuándo transmitir el tráfico generado o recibido desde otros satélites.

En los últimos años, se han llevado a cabo numerosos avances en la implementación de DTN y se han realizado experimentos en órbita que avalan el potencial beneficio de esta arquitectura. Además, se ha estudiado el problema del diseño del plan de contactos que constituye el proceso de configurar y elegir apropiadamente las oportunidades de comunicación con el objetivo de optimizar el rendimiento y realizar una adecuada gestión de los limitados recursos que se disponen en este tipo de redes.

Sin embargo, esta arquitectura se encuentra aún en una etapa de maduración y son numerosos los desafíos que deben ser superados. En particular, en esta tesis se pone el foco en el problema de la congestión que ocurre cuando la planificación referida a la utilización de los recursos no resulta apropiada. Básicamente, se proponen mecanismos que actúan, por un lado, sobre el algoritmo de enrutamiento ejecutado de manera distribuida por los satélites de la red, y por otro, sobre el diseño automático del plan de contactos. Adicionalmente, se realizan aportes referidos a los problemas de incertidumbre y de escalabilidad e integración de la red. De esta manera, en esta investigación contribuimos con enfoques originales en los que se aprovecha la predictibilidad de las comunicaciones satelitales para proveer mecanismos de gestión y toma de decisiones de manera automatizada, solucionando de esta forma problemas cuya complejidad aumenta drásticamente con la cantidad de satélites y tiempos de evaluación.

Cabe destacar además, que si bien los aportes realizados pueden ser utilizados de forma general en la arquitectura DTN, los mismos encuentran aplicación dentro del *Plan Espacial Argentino*, ya que esta arquitectura resulta particularmente adecuada para ser aplicada en las nuevas misiones satelitales distribuidas que propone CONAE para la observación terrestre. Por lo tanto, los aportes brindados en esta tesis tienen alcance hacia protocolos desarrollados mediante colaboraciones internacionales y podrían además ser utilizados en nuestro país para resolver desafíos de interés local.

Palabras clave: Redes Satelitales; DTN; Disrupciones; Enrutamiento; Congestión.

UNIVERSIDAD NACIONAL DE CÓRDOBA

Resumen en Inglés

-Abstract-

Universidad Nacional de Córdoba
Facultad de Ciencias Exactas Físicas y Naturales

PhD. in Engineering Sciences

Models, Algorithms and Protocols for Disruption Tolerant Communication Networks with High Degree of Predictability

by Pablo Gustavo

MADOERY

The satellite communications industry has shown limited progress in the last decades compared to the evolution of terrestrial networks such as Internet. The main reason is due to the fact that the space environment is radically different from the terrestrial one, which have a considerable impact on the stability of the connections and on the fact that the communication protocols used on Earth are inadequate and/or inefficient when trying to adapt them to space. However, recently, and as a result of a joint effort of different space agencies (NASA, ESA, CONAE, etc.), there have been studies and experiments with network communication strategies that are capable of tolerating delays and disruptions through a paradigmatic change in the way of carrying out such communications. In particular, a protocol architecture called *Delay/Disruption Tolerant Networking (DTN)* has been proposed, and given the predictability of satellite trajectories, it allows to take advantage of a contact plan composed of communication opportunities in the future, which in turn allows satellites to make efficient decisions about how and when to transmit the traffic generated or received from other satellites.

In recent years, numerous advances have been made in the implementation of DTN and in-orbit experiments have been carried out in order to support the potential benefits of this architecture. In addition, there have been studies regarding the contact plan design, which is the process of configuring and appropriately choosing the communication opportunities with the aim of optimizing performance and properly managing the limited resources available in this type of networks.

However, this architecture is still at a maturation stage and there are numerous challenges that must be overcome. In particular, this thesis focuses on the problem of congestion, which occurs when the planning of resources utilization results in inappropriate. Basically, we propose mechanisms that act, on the one hand, on the routing algorithm executed in a distributed manner by the satellites of the network, and on the other hand, on the automatic contact plan design process. Furthermore, we make additional contributions regarding the problems of uncertainty and scalability/integration of the network. In this way, we contribute with original approaches, where the predictability of satellite communications is exploited to provide automated mechanisms of management, control, and decision making, in order to solve problems whose complexity increases drastically with the number of satellites and evaluation times.

It should be also noted that, although the contributions made in this thesis have general application to the DTN architecture, they also result appealing within the *Argentinean Space Plan*, given that it is particularly suitable to be applied in the novel distributed satellite missions proposed by CONAE for terrestrial observation. Therefore, the contributions provided in this thesis embrace protocols developed through international collaborations and could also be used in our country to solve challenges of local interest.

Keywords: Satellite Networks; DTN; Disruptions; Routing; Congestion.

UNIVERSIDAD NACIONAL DE CÓRDOBA

Resumen en Portugués

-Resumo-

Universidad Nacional de Córdoba
Facultad de Ciencias Exactas Físicas y Naturales

PhD. em Ciências da Engenharia

Modelos, Algoritmos e Protocolos para Redes de Comunicação Interrupção-Tolerante com Alto Grau de Previsibilidade

por Pablo Gustavo

MADOERY

O sector das comunicações por satélite tem registado progressos limitados nas últimas décadas em comparação com a evolução das redes terrestres como Internet. A principal razão é que o ambiente espacial é radicalmente diferente do ambiente terrestre, o que tem um impacto considerável na estabilidade das ligações e no facto de os protocolos de comunicação utilizados na Terra serem inadequados e/ou ineficientes quando se trata de adaptação ao espaço. No entanto, recentemente, e produto de um esforço conjunto de diferentes agências espaciais (NASA, ESA, CONAE, etc.), começou a estudar e experimentar estratégias de comunicação em rede que são capazes de tolerar atrasos e interrupções através de uma mudança de paradigma na forma de realizar tais comunicações. Em particular, foi proposta uma arquitectura de protocolo denominada *Delay/Disruption Tolerant Networking (DTN)* e, dada a previsibilidade das trajectórias dos satélites, permite tirar partido de um plano de contacto composto por oportunidades de comunicação no futuro, o que permite aos satélites tomar decisões eficientes sobre como e quando transmitir o tráfego gerado ou recebido de outros satélites.

Nos últimos anos, foram feitos inúmeros avanços na implementação da DTN e foram realizadas experiências em órbita que endossam o potencial benefício desta arquitectura. Além disso, foi estudado o problema do desenho do plano de contactos, que constitui o processo de configuração e escolha das oportunidades de comunicação, com o objectivo de otimizar o desempenho e efectuar uma gestão adequada dos limitados recursos disponíveis neste tipo de redes.

No entanto, esta arquitetura ainda se encontra numa fase de maturação e há muitos desafios que têm de ser ultrapassados. Em particular, esta tese centra-se no problema do congestionamento que ocorre quando o planeamento da utilização dos recursos não é adequado. Basicamente, são propostos mecanismos que atuam, por um lado, sobre o algoritmo de roteamento executado de forma distribuída pelos satélites de rede e, por outro, sobre o desenho automático do plano de contatos. Adicionalmente, são feitas contribuições sobre os problemas de incerteza e escalabilidade e integração da rede. Desta forma, nesta pesquisa contribuímos com abordagens originais no que tira partido da previsibilidade das comunicações por satélite para fornecer mecanismos de gestão e tomada de decisão de forma automatizada, resolvendo desta forma problemas cuja complexidade aumenta drasticamente com o número de satélites e tempos de avaliação.

Deve-se também destacar que, embora as contribuições realizadas possam ser utilizadas de maneira geral na arquitetura da DTN, elas são aplicadas no âmbito do Plano Espacial Argentino, já que esta arquitetura é particularmente adequada para ser aplicada nas novas missões de satélite distribuído propostas pela CONAE para a observação terrestre. Portanto, as contribuições oferecidas nesta tese têm escopo para protocolos desenvolvidos através de colaborações internacionais e também podem ser usadas em nosso país para resolver desafios de interesse local.

Palavras-chave: Redes de Satélite; DTN; Interrupções; Roteamento; Congestionamento.

Agradecimientos

A Jorge Finochietto, por compartirme su visión y motivación para aprender a pensar de manera curiosa, y por contagiarme el gusto por descubrir y comprender los problemas, mas allá de la mera obtención de resultados. A Pablo Ferreyra, por su humor y aliento cotidiano. A la gente del Laboratorio de Comunicaciones Digitales por su interminable apoyo a lo largo de este proceso: Graciela Briones, Mario Hueda, Carmen Rodriguez, Juan Fraire, Fernando Raverta, Nehuén Gonzalez, Diego Murature, Juan Leal Licudis, Luis Rodriguez, Sergio Baudino, Martín Ayarde, Fede Baldoni, Marcos, Renato Cherini, Horacio Medoza, Guillermo Riva, Carlos Zerbini, Gabriel Maggio, Mariano Lizarraga, Toni Abdala, Jairo Trad, Nico, Maxi, ET, Mati, Fede, César, Rubén, Leo, Jere, Ramiro, Gabi, Ariane y Adrián. A la gente de STI y CONAE por generar el contexto de motivación para que este proyecto tuviera sentido. Al Conicet y Fulbright por el aporte técnico y financiero que posibilitó este logro. A Scott Burleigh, Leigh Torgerson y Marc Sanchez del JPL por la ayuda en una experiencia única.

A mis padres Gustavo Madoery e Inés Simón, por brindarme apoyo emocional, ADN, enseñarme sus valores, soportar, inspirar, promover y a veces evadir mis constantes cuestionamientos, y por brindarme los recursos necesarios para que pueda estudiar. A mis hermanos Guillermo y Romina, por su amistad invaluable y el aguante eterno. A mis abuelos Coco, Cora, Tuchi y Lucha, por sus sabios y valiosos consejos. A mis tíos y primos, por su compañía constante. A Roxana Farías por sus iluminadoras e inextinguibles charlas, su amor y comprensión.

A los Compamigos, por el espíritu desmedido y las anécdotas irreproducibles de toda una vida: Fede Jover, Adrian Giorgio, Joaquin Pellegrini, Ezequiel Arguello, Charly Alvarez, Martín Albano, Cristian Benolol, Panchi Cecarelli, Mauricio Bustamante, Ale Bonetto, Diego Mattos. A la gente de la Casona, por las inspiradoras salidas oscurantistas. A Edel y Yúsef, por las aventuras compartidas. A mis amigos del Chaco, por la infancia insuperable. A mis profesores y amigos del Colegio Sarmiento y del Instituto Aeronáutico, por las innumerables enseñanzas. A la familia Roldán y Piedrabuena, por el camino compartido. A mis profesores y amigos de Natación, Karate, Taekwondo, Ajedrez, Básquet, Voley, Tenis, Órgano, Guitarra, Ping Pong y Tai Chi Chuan, por la diversión inagotable. A los Olímpicos del Conicet, por su garra y compañerismo.

A todos aquellos quienes aportaron su granito de arena para que hoy tenga la insania necesaria para poder doctorarme y para que sea una persona tanto íntegra como desintegrada, y a quienes no alcanzo a nombrar por cuestiones de espacio u olvido premeditado. Sin dudas ha sido un camino arduo y serpenteante pero también entretenido y enriquecedor. Gracias Totales !

Índice General

Resumen en Español	III
Resumen en Inglés	V
Resumen en Portugués	VII
Agradecimientos	IX
Índice General	XI
Índice de Figuras	XV
Índice de Tablas	XVII
Índice de Algoritmos	XVIII
Glosario	XIX
1. Introducción	1
1.1. Problema de Investigación	2
1.1.1. Relevancia del Problema	3
1.2. Hipótesis y Objetivos	4
1.2.1. Metodología de Trabajo	4
1.3. Estructura General de la Tesis	5
1.4. Contribuciones	6
1.4.1. Congestión	6
1.4.2. Incertidumbre	7
1.4.3. Escalabilidad	7
1.4.4. Trabajos Adicionales	8
1.5. Marco Teórico	10
1.5.1. Evolución de las Comunicaciones	10
1.5.2. DTN: Redes Tolerantes a Retardos e Interrupciones	12
1.5.2.1. Origen y Evolución de DTN	12
1.5.2.2. El Protocolo Bundle	14
1.5.2.3. Esquemas de Enrutamiento	17
1.5.2.4. Implementaciones	18
1.5.2.5. Experimentos en Órbita	19

1.5.2.6.	Frontera del Estado del Arte	22
1.5.3.	La Arquitectura Segmentada para la Observación Terrestre	23
1.5.3.1.	La Teledetección	23
1.5.3.2.	El Entorno Espacial	23
1.5.3.3.	Los Pequeños Satélites	24
1.5.3.4.	La Arquitectura Segmentada	25
2.	Comunicaciones Planificadas y Mecanismos de Enrutamiento	29
2.1.	Introducción	29
2.2.	Comunicaciones Planificadas	30
2.2.1.	Definición de Nodo e Identificador	30
2.2.2.	Definición de Contacto	30
2.2.3.	Definición de Topología y Plan de Contactos	31
2.2.4.	Restricciones de Recursos	32
2.2.4.1.	Restricciones de Tiempo y Zona	32
2.2.4.2.	Restricciones de Recursos Concurrentes	33
2.2.5.	Modelado de la Topología y el Plan de Contactos	35
2.2.5.1.	Modelado como Lista de Contactos	36
2.2.5.2.	Modelado como Máquina de Estados Finitos	36
2.2.6.	Diseño del Plan de Contactos	36
2.3.	Rutas Temporales	39
2.4.	Flujo de la Información	40
2.5.	Mecanismos de Enrutamiento	41
2.5.1.	Contact Graph Routing (CGR)	42
2.5.1.1.	Mejoras Realizadas	42
2.5.1.2.	Descripción del Algoritmo	43
3.	Congestión	50
3.1.	Introducción	50
3.1.1.	Congestión en Internet	51
3.1.2.	Congestión en DTN	51
3.2.	Congestión con un Único Tráfico	52
3.3.	Local Path Aware CGR (LPA-CGR)	53
3.3.1.	Concepto	53
3.3.2.	Descripción Algorítmica	55
3.3.3.	Consideraciones sobre Envío al Nodo Previo	56
3.4.	Congestión con Múltiples Tráficos	57
3.5.	Modelo de Flujo Óptimo de Múltiples Tráficos	58
3.5.1.	Coefficientes y Variables de Decisión	59
3.5.2.	Función Objetivo y Restricciones	60
3.5.3.	Ejemplo	61
3.6.	Multi Graph CGR (MG-CGR)	62
3.6.1.	Concepto	62
3.6.2.	Descripción Algorítmica	63
3.6.3.	Implementabilidad	65
3.6.3.1.	Problema de Encolamiento por Nodo	65
3.6.3.2.	Problema de Múltiples Tráficos por Nodo	66

3.7. Global Path Aware CGR (GPA-CGR)	68
3.7.1. Concepto	68
3.7.2. Descripción Algorítmica	69
3.8. Optimización Evolutiva de GPA-CGR (EGPA-CGR)	71
3.8.1. Concepto	71
3.8.2. Descripción Algorítmica	72
3.9. Evaluación de Rendimiento	74
3.9.1. Caso de Estudio	74
3.9.2. Herramienta de Simulación: Dtnsim	76
3.9.3. Análisis de Resultados	77
3.9.4. Análisis de Algoritmos	81
3.10. Conclusiones Parciales	83
4. Incertidumbre	84
4.1. Introducción	84
4.2. Fallas en Sistemas Espaciales	85
4.3. Estudio de Cotas de Rendimiento	87
4.3.1. Extensión de DtnSim	87
4.3.2. Escenarios Simulados	87
4.3.3. Análisis de Resultados	88
4.4. Estudio de Replicación	91
4.4.1. Esquemas de Enrutamiento	91
4.4.1.1. Oportunísticos	92
4.4.1.2. Planificados (tiempo de entrega)	92
4.4.1.3. Planificados (número de saltos)	92
4.4.1.4. Planificados (2 copias)	93
4.4.2. Métricas de Rendimiento	94
4.4.3. Escenarios Simulados	94
4.4.3.1. Redes Aleatorias	94
4.4.3.2. Constelación Walker	95
4.4.4. Análisis de Resultados	96
4.4.4.1. Caso de Estudio I, Redes Aleatorias, Baja Congestión	96
4.4.4.2. Caso de Estudio II, Redes Aleatorias, Alta Congestión	98
4.4.4.3. Caso de Estudio III, Constelación Walker	99
4.5. Discusión	100
4.5.0.4. Modelo de Fallas	100
4.5.0.5. Re-cálculo de Ruta	101
4.5.0.6. CGR Probabilístico	101
4.6. Conclusiones Parciales	102
5. Escalabilidad e Integración	103
5.1. Introducción	103
5.2. Enrutamiento y Limitaciones de Escalabilidad	104
5.3. Modelo de sistema	105
5.3.1. Global Full Knowledge (GFK)	105
5.3.2. Definición de Regiones	105
5.3.3. Global Regionalized Knowledge (GRK)	107

5.3.4. Local Regionalized Knowledge (LRK)	108
5.3.4.1. Planes de Contactos Regionales	108
5.3.4.2. Enrutamiento Jerárquico	108
5.3.4.3. Esquema Inter-RR	109
5.3.4.4. Algoritmo Inter-RR	110
5.3.4.5. Utilización de Inter-RR	112
5.4. Evaluación	112
5.4.1. Entorno de Simulación	112
5.4.2. Análisis de Resultados	114
5.5. Conclusiones Parciales	116
6. Conclusiones	117
Bibliografía	119

Índice de Figuras

1.1. Evolución temporal de las tecnologías de comunicación	10
1.2. Integración de redes heterogéneas en una red DTN.	14
1.3. Protocolo Bundle como capa de superposición.	15
1.4. Campos y formato de un Bundle de acuerdo a RFC5050.	16
1.5. Misiones de observación de la Tierra de CONAE	23
1.6. Beneficios de la Arquitectura Segmentada.	26
2.1. Contactos variables en el tiempo	31
2.2. Interferencia ocasionada a un satélite GEO	33
2.3. Arquitectura de plataformas satelitales con a) múltiples posibles vecinos, b) un conmutador de potencia, c) dos equipos de comunicaciones, y d) una antena direccionable electrónica o mecánicamente	34
2.4. Modelos de la topología y el plan de contactos: a) trayectorias orbitales, b) lista de contactos, c) máquina de estados finitos (FSM)	35
2.5. Rendimiento esperado de los mecanismos de diseño FCP, RACP y TACP	37
2.6. Dos posibles planes de contactos que priorizan a) máxima capacidad de transmisión y b) justicia de asignación de enlaces	38
2.7. Flujo de información en redes DTN planificadas	41
2.8. Ejemplo de plan de contactos en representación temporal con rutas desde un nodo A hacia un nodo D	46
2.9. Ejemplo de plan de contactos en representación como grafo de contactos desde un nodo A hacia un nodo D	47
3.1. Plan de contactos de caso de ejemplo con un flujo de tráfico	52
3.2. Flujo de un único tráfico: a) con congestión, b) sin congestión	53
3.3. Reacción a la congestión mediante re-ruteo hacia el nodo previo	56
3.4. Plan de contactos de caso de ejemplo con dos flujos de tráfico	57
3.5. Flujo de dos tráficos: a) con congestión, b) sin congestión	58
3.6. Solución proporcionada por modelo de flujos óptimos	62
3.7. Representación gráfica de esquema MG-CGR	63
3.8. Problema de encolamiento por nodo	65
3.9. Problema de múltiples tráficos por nodo	66
3.10. Alternativas para obtener soluciones óptimas e implementables	67
3.11. Representación gráfica del esquema GPA-CGR	68
3.12. Representación gráfica del esquema EGPA-CGR	71
3.13. Topología del caso de estudio de formación lineal	75
3.14. DtnSim: Simulador de redes DTN	77
3.15. Resultados de simulación	78

3.16. Flujos de tráfico para a) CGR y b) EGPA-CGR con $\rho = 1$	79
3.17. Resultados de variación de los tamaños de los buffers	81
3.18. Ejecuciones de los algoritmos	82
4.1. Enrutamiento en escenario planificado con fallas	85
4.2. Cantidad de rutas por contacto	86
4.3. Disponibilidad de rutas con inyección de fallas de forma aleatoria	89
4.4. Disponibilidad de rutas con inyección de fallas de forma dirigida	89
4.5. Rendimiento con inyección de fallas de forma aleatoria	90
4.6. Rendimiento con inyección de fallas de forma dirigida	90
4.7. Constelación Walker	95
4.8. Resultados de caso de estudio I) redes aleatorias con baja congestión	97
4.9. Resultados de caso de estudio II) redes aleatorias con alta congestión	98
4.10. Resultados de caso de estudio III) constelación Walker	99
5.1. Red espacial dividida en 6 regiones (A, B, C, D, E, F)	106
5.2. Regiones y nodos de la red de la Figura 1	107
5.3. Grafo estático conformado por el algoritmo de enrutamiento Inter-RR propuesto.	110
5.4. Captura de la red espacial de las Figuras 1 y 2	114
5.5. Resultados de simulación. a) Escalabilidad. b) Rendimiento	115

Índice de Tablas

2.1. Ejemplo de plan de contactos en forma de tabla	47
3.1. Parámetros del modelo de flujo óptimo de tráfico	59
3.2. Parámetros orbitales e intervalos de tiempo de caso de estudio de formación lineal	74
3.3. Tabla de comparación de algoritmos	82
4.1. Preferencia de esquemas de enrutamiento en constelación Walker	100
5.1. Parámetros satelitales	113

Índice de Algoritmos

1.	CGR Forward	43
2.	Identify Proximate Node List	45
3.	Load Route List	48
4.	LPA-CGR Forward	54
5.	LPA-CGR Identify Proximate Node List	55
6.	GPA-CGR	69
7.	EGPA-CGR	73
8.	Inter-RR	111

Glosario

AMS	A synchronous M essage S ervice
AOCS	A ttitude and O rbital C ontrol S ystem
AS	A rquitectura S egmentada
BER	B it E rror R ate
BP	B undle P rotocol
CCSDS	C onsultative C ommittee for S pace and D ata S ystems
CFDP	C CSDS F ile D elivery P rotocol
CGR	C ontact G raph R outing
CLA	C onvergence L ayers A dapters
CONAE	C OMisión N acional de A ctividades E spaciales
COTS	C ommertial O f T he S helf
CPD	C ontact P lan D esign
DINET	D eep I mpact N etwork E xperiment
DMC	D isaster M onitoring C onstellation
DSN	D eep S pace N etwork
DTN	D elay/ D isruption T olerant N etworks
DTNRG	D elay/ D isruption T olerant N etworks R earch G roup
DTNWG	D elay/ D isruption T olerant N etworks W orking G roup
EA	E volutionary A lgorithms
EBCGR	E xtension B lock C GR
EGPA-CGR	E volutionary G lobal P ath A ware - C GR
ESA	E uropean S pace A gency
ESL	E arth S atellite L ink
FSM	F inite S tate M achine
GEO	G eoestationary E arth O rbital

GFK	Global Full Knowledge
GRK	Global Regionalized Knowledge
GPA-CGR	Gobal Path Aware - CGR
IBR	Institut Betriebssysteme Rechnerverbund
ION	Interplanetary Overlay Network
IP	Internet Protocol
IPNSIG	Interplanetary Networking Special Interest Group
ISL	Inter Satellite Link
ISS	International Space Station
JHUAPL	Johns Hopkins University Applied Physics Laboratory
JPL	Jet Propoulsion Laboratory
LEO	Low Earth Orbit
LPA-CGR	Local Path Aware - CGR
LRK	Lobal Regionalized Knowledge
LTP	Licklider Transmission Protocol
MAC	Medium Access Control
MG-CGR	Multi Graph - CGR
LP	Linear Programming
MOC	Mission Operations and Control Center
NASA	National Aeronautic and Space Administration
PEN	Plan Espacial Nacional
POSIX	Portable Operating System Interfaces para Unix
RACP	Route Aware Contact Plan
RFC	Request For Comments
TACP	Traffic Aware Contact Plan
TCP	Transfer Control Protocol
UDP	User Datagram Protocol
UNC	Universidad Nacional de Córdoba

Capítulo 1

Introducción

Tradicionalmente, los sistemas de comunicación han buscado cumplir los objetivos de conectar a la mayor cantidad de personas y/o dispositivos por un lado (cobertura), y en incrementar los servicios y prestaciones ofrecidas por otro. El primero de estos objetivos se ha conseguido generalmente mediante el despliegue de una infraestructura, es decir elementos interconectados en red, de forma tal que se proporcionen los medios necesarios para que el transmisor de un mensaje pueda eventualmente entregar dicho mensaje a un receptor. Por otra parte, el segundo objetivo entra en juego una vez que ya se dispone de una infraestructura de red y está mas enfocado en el mejoramiento tecnológico de los elementos intervinientes de forma que se incrementen los anchos de banda y la eficiencia de las comunicaciones.

Tomando como ejemplo a Internet, es claro que los esfuerzos se han orientado a crear una red universal y ubicua que conecte a toda la población y que además sea eficiente en el intercambio de grandes volúmenes de datos. Es así como actualmente aproximadamente el 50 % de las personas del mundo tiene acceso a Internet. Si bien este es indudablemente un logro valioso, cabe cuestionarse que aspectos limitan el alcance de Internet al otro 50 % de la población mundial. Como ocurre generalmente, una de las causas resulta ser la falta de rentabilidad económica que implica establecer la infraestructura necesaria para proporcionar servicio a lugares remotos, aislados y poco poblados. Sin embargo, dicha falta de rentabilidad económica se basa, entre otros aspectos, en la suposición de que la infraestructura de red a ser instalada debe cumplir con ciertos estándares y metodologías, los que a su vez están apoyados por una larga herencia en la forma de

concebir las comunicaciones y sus aplicaciones. Particularmente, la suposición mas fuerte es que los nodos de la red deben poder establecer comunicaciones extremo a extremo (fuente-destino) de una forma contemporánea, síncrona, y altamente conversacional.

Sin embargo, ha surgido en los últimos años un enfoque conocido como *Delay/Disruption Tolerant Networking (DTN)*, el cual especifica una arquitectura en donde se proveen mecanismos para dar soporte a comunicaciones asíncronas y sin requerir de la existencia de una conversación contemporánea extremo a extremo. La relajación de este requerimiento, aunque no resulta trivial en cuanto a los desafíos involucrados, permite extender Internet para abarcar redes heterogéneas con una infraestructura, costos y servicios arbitrariamente reducidos. De esta manera, aunque es evidente que una mayor infraestructura conlleva mayores capacidades de transmisión de datos y calidades de servicio, el valor de esta nueva arquitectura radica en proveer la flexibilidad que permite el despliegue de aplicaciones que puedan tolerar demoras e interrupciones en una red con recursos limitados. Dichas aplicaciones se hacen presentes en un gran número de entornos. Ejemplos de las mismas se encuentran en las redes de sensores, terrestres móviles, satelitales, submarinas, militares, entre otras, y el hecho de poder integrarlas de forma mas transparente constituye una gran motivación para el desarrollo de nuestros sistemas de comunicaciones tanto actuales como futuros.

1.1. Problema de Investigación

La arquitectura de comunicaciones DTN resulta particularmente adecuada para ser aplicada dentro de la llamada *Arquitectura Segmentada*, un programa desarrollado por la Comisión Nacional de Actividades Espaciales (CONAE) con el objetivo de dar rápida respuesta a la creciente demanda de datos de observación de la Tierra desde el espacio. Esta arquitectura se basa en el desarrollo tecnológico de una nueva generación de satélites e instrumentos, mediante el uso de un conjunto de plataformas pequeñas (llamadas segmentos), que operadas coordinadamente podrían proporcionar respuestas equivalentes o superiores a las que es posible obtener actualmente mediante una única plataforma (satélite monolítico).

Aunque los beneficios potenciales de desarrollar esta arquitectura son enormes, se deben resolver numerosos desafíos antes de que la misma pueda ser implementada de forma

operativa. Entre estos se destacan la distribución de funciones entre segmentos (memoria, bajada de datos, procesamiento, etc.), la distribución de instrumentos (sistema óptico o de antena partida), y las comunicaciones en red entre los segmentos orbitantes. En particular, en esta tesis se hacen aportes a este último desafío, poniendo el foco en la planificación de las oportunidades de comunicación entre los segmentos, el mecanismo de enrutamiento de datos utilizado y la evitación de la congestión.

Una particularidad de las redes satelitales reside en el hecho de que las oportunidades de comunicación resultan predecibles, es decir, se puede predecir con exactitud en que momentos dos nodos satelitales podrán comunicarse en el futuro y en que momentos no. Dichas oportunidades de comunicación entre nodos se conocen en la literatura como contactos y existe la posibilidad de aprovisionar a los nodos con un conjunto de esos contactos con cierta antelación. Este conjunto de contactos recibe el nombre de plan de contactos y luego de recibirlo cada nodo puede enviar datos a otros nodos utilizando un protocolo y un algoritmo de enrutamiento apropiado. Este trabajo se centra en la interacción entre el plan de contactos provisto a cada nodo y el algoritmo de enrutamiento ejecutado de manera distribuida por todos los nodos de la red. Específicamente se estudiará el problema de la congestión de tráfico que se hace presente en esta interacción y se propondrán soluciones incrementales que busquen resolver el problema mejorando de esta manera el rendimiento de las comunicaciones. Adicionalmente, se harán aportes orientados a abordar otros problemas tales como las fallas transitorias y la escalabilidad.

1.1.1. Relevancia del Problema

El problema mencionado es de gran importancia tanto a nivel nacional como internacional. Por un lado, organismos internacionales de estandarización tales como *Internet Engineering Task Force (IETF)* y *Consultative Committee for Space Data Systems (CCSDS)* se encuentran diseñando y mejorando la arquitectura de protocolos DTN para que pueda ser utilizada tanto en redes espaciales como en redes terrestres. Por otro lado, el *Plan Espacial Nacional (PEN)* busca desarrollar misiones satelitales distribuidas que sean económicas, autónomas, eficientes y flexibles como la actualmente ideada para la observación de la Tierra por CONAE bajo el nombre SARE. Por lo tanto, los aportes

brindados en esta tesis tienen alcance hacia protocolos desarrollados mediante colaboraciones internacionales y podrían además ser utilizados en nuestro país para resolver desafíos de interés local y lograr una mejor gestión de nuestros recursos.

Mayores detalles sobre el alcance de DTN y de la arquitectura segmentada son brindados posteriormente en el Marco Teórico (1.5).

1.2. Hipótesis y Objetivos

La hipótesis en la que nos basaremos consiste en que *es posible mejorar el rendimiento de las comunicaciones en redes satelitales disruptivas mediante técnicas que actúen sobre la interacción entre el diseño del plan de contactos y el algoritmo de enrutamiento ejecutado de manera distribuida*. Para probar esta hipótesis se proponen los siguientes objetivos secundarios:

1. Comprender el estado del arte de las redes DTN.
2. Generar técnicas de modelado que nos permitan plantear formalmente el diseño del plan de contactos y el enrutamiento de datos en redes DTN.
3. Proponer e implementar mejoras a los actuales esquemas de diseño del plan de contactos haciendo foco en la congestión, incertidumbre y escalabilidad.
4. Proponer e implementar mejoras a los actuales esquemas de enrutamiento de datos haciendo foco en la congestión, incertidumbre y escalabilidad.
5. Desarrollar herramientas de simulación que nos permitan evaluar las mejoras propuestas y compararlas con las del estado del arte.

1.2.1. Metodología de Trabajo

Como metodología de trabajo se procederá en primer lugar a desarrollar representaciones que nos permitan describir redes DTN satelitales planificadas, cuya conectividad pueda variar en el tiempo. Dichas representaciones deberán resultar de utilidad tanto para aplicar algoritmos de planificación de las comunicaciones como de enrutamiento de datos.

En segunda instancia, y para cada problema planteado (congestión, incertidumbre y escalabilidad), se analizarán casos simples que permitan evidenciar claramente los compromisos involucrados. Se espera poder utilizar el entendimiento de los problemas observados para proponer estrategias alternativas a las actualmente utilizadas en el estado del arte.

Una vez divisadas las distintas estrategias, se buscará implementarlas en algoritmos que serán finalmente evaluados y comparados mediante simulaciones y análisis. Con este propósito se deberán desarrollar las herramientas de simulación necesarias que implementen tanto los modelos de red como los algoritmos mencionados. Adicionalmente, en los casos en que sea pertinente se desarrollarán modelos de programación lineal (*Linear Programming (LP)*) que nos permitan obtener soluciones óptimas a los problemas aquí tratados.

1.3. Estructura General de la Tesis

El doctorando es parte de un grupo de investigación que ha venido realizando aportes en DTN desde el año 2013. Cabe aclarar en este contexto que el doctorando ha tenido una participación y colaboración gradual, y que parte de esta investigación se solapa parcialmente con una tesis presentada por Fraire en [1]. Dicha investigación es explicada brevemente tanto en el Marco Teórico (1.5) como en el capítulo 2. Por otra parte, avances posteriores donde el doctorando ha tenido un rol más protagónico son plasmados en los capítulos sucesivos y constituyen el aporte original de este trabajo.

Esta tesis se estructura en 6 capítulos. Se procederá en primer lugar a explicar el estado del arte en dos etapas. En una primera etapa, contenida dentro del Marco Teórico (1.5), se detallará de manera general el origen y funcionamiento de la arquitectura DTN y los principios que motivan su aplicación al desarrollo de la Arquitectura Segmentada. En una segunda etapa, contenida en el capítulo 2, se describirán brevemente los aportes realizados por el grupo de investigación al diseño del plan de contactos y a los algoritmos de enrutamiento. Esta constituye la base sobre la que se realizan los aportes principales de esta tesis en particular. En el capítulo 3 se describen los aportes realizados en la evitación de la congestión tanto a nivel local como a nivel global de la red. En el capítulo 4 se trata el problema de la incertidumbre en las comunicaciones planificadas mientras

que en el capítulo 5 se tratan los problemas de escalabilidad e integración de la red. Finalmente se presentan las conclusiones generales en el capítulo 6.

1.4. Contribuciones

En esta tesis se presentan contribuciones respecto a tres problemas existentes en las redes DTN satelitales: congestión, incertidumbre y escalabilidad. Cada una de las contribuciones se ha plasmado en los trabajos que se detallan a continuación:

- Congestión: [2], [3], [4]
- Incertidumbre: [5], [6], [7], [8]
- Escalabilidad: [9], [10]

1.4.1. Congestión

El trabajo presentado en [2] es una extensión de [4], donde se propone un esquema de enrutamiento llamado *LPA-CGR*, capaz de evitar la congestión generada por tráfico local, y dos esquemas de planificación y enrutamiento llamados *GPA-CGR* y *EGPA-CGR*, capaces de evitar la congestión a nivel global. En el trabajo presentado en [3] se estudian los beneficios de utilizar un esquema llamado *MG-CGR* en un caso de constelación satelital para la observación de la tierra.

P. Madoery, J. Fraire, and J. Finochietto, “*Congestion management techniques for disruption-tolerant satellite networks*,” *International Journal of Satellite Communications and Networking*, May 2017. [2].

P. Madoery, J. Fraire, and J. Finochietto, “*Analysis of communication strategies for earth observation satellite constellations*,” *IEEE Latin America Transactions*, June 2016. [3].

J. Fraire, **P. Madoery**, J. Finochietto, and E. Birrane. “*Congestion modeling and management techniques for predictable disruption tolerant networks*,” In *2015 IEEE 40th Conference on Local Computer Networks (LCN)*, Clearwater Beach, FL, Oct 2015. [4].

1.4.2. Incertidumbre

El trabajo presentado en [5] propone tres esquemas de enrutamiento llamados *CGR-Hops*, *CGR-2Copies* y *CGR-2Copies-0.5*, los cuales son comparados junto a otros esquemas de enrutamiento tanto planificados como oportunistas en constelaciones satelitales aleatorias y realistas. Los trabajos [6] y [7] analizan cotas de rendimiento para el esquema *CGR* en redes aleatorias con distintos métodos de inyección de fallas. El trabajo [8] abre una nueva línea de investigación proponiendo un nuevo modelo para el análisis de fallas utilizando procesos de decisión de Markov.

P. Madoery, F. Raverta, J. Fraire, and J. Finochietto, “*Routing in space delay tolerant networks under uncertain contact plans*,” In 2018 IEEE International Conference on Communications (ICC), Kansas, MO, USA, May 2018. [5].

P. Madoery, F. Raverta, J. Fraire, and J. Finochietto, “*On the performance analysis of disruption tolerant satellite networks under uncertainties*,” In 2017 XVII Workshop on Information Processing and Control (RPIC), Mar del Plata, Argentina, Sept 2017. [6].

J. Fraire, **P. Madoery**, S. Burleigh, M. Feldmann, J. Finochietto, A. Charif, N. Zergainoh, and R. Velazco, “*Assessing contact graph routing performance and reliability in distributed satellite constellations*,” *Journal of Computer Networks and Communications*, 2017. [7].

F. Raverta, R. Demasi, **P. Madoery**, J. Fraire, J. Finochietto, and P. D’Argenio, “*A Markov Decision Process for Routing in Space DTNs with Uncertain Contact Plans*,” In 2018 IEEE International Conference on Wireless for Space and Extreme Environments (WiSEE) Huntsville, AL, USA, Dec 2018. [8].

1.4.3. Escalabilidad

El trabajo presentado en [9] fué realizado en el *Jet Propulsion Laboratory* de *NASA* y *Caltech*, gracias a la ayuda de una beca *Fulbright* que permitió consolidar y ampliar la colaboración con el grupo de trabajo del doctorando. Aquí se proponen dos esquemas de planificación y enrutamiento llamados *GRK* y *LRK* orientados a mejorar la escalabilidad mediante la disminución del esfuerzo de cálculo de rutas. El trabajo presentado en [10]

persigue el mismo objetivo mediante cambios propuestos en el algoritmo de cálculo de rutas de CGR.

P. Madoery, J. Fraire, F. Raverta, J. Finochietto, and S. Burleigh, “*Managing Routing Scalability in Space DTNs*,” In 5th IEEE International Workshop on Space-Terrestrial Internetworking (STINT), Huntsville, AL, USA, Dec 2018. [9].

P. Madoery, C. Zerbini, J. Fraire, and J. Finochietto, “*Enhancing contact graph routing forwarding. performance for segmented satellites architectures*,” In 1st IAA Latin American Symposium on Small Satellites: Advanced Technologies and Distributed Systems, Buenos Aires, Argentina, March 2017. [10].

1.4.4. Trabajos Adicionales

Cabe destacar además que el doctorando ha colaborado en los trabajos que son mencionados a continuación. Si bien no son plasmados explícitamente en la presente tesis, estos trabajos forman parte de una base sobre la cual se desarrollaron trabajos posteriores y se realizaron los aportes mencionados.

J. Fraire, **P. Madoery**, and J. Finochietto, “*On the design of fair contact plans for delay tolerant networks*,” In 2013 IEEE International Conference on Wireless for Space and Extreme Environments (WiSEE), Baltimore, Maryland, USA, 2013. [11].

J. Fraire, **P. Madoery**, and J. Finochietto, “*Leveraging routing performance and congestion avoidance in predictable delay tolerant networks*,” In 2014 IEEE International Conference on Wireless for Space and Extreme Environments (WiSEE), Noordwick, Netherlands, Oct 2014. [12].

J. Fraire, **P. Madoery**, and J. Finochietto, “*On the design and analysis of fair contact plans in predictable delay-tolerant networks*,” IEEE Sensors Journal, Nov 2014. [13].

J. Fraire, **P. Madoery**, J. Finochietto, and G. Leguizamón, “*Preliminary results of an evolutionary approach towards contact plan design for satellite dtns*,” In 2015 IEEE International Conference on Wireless for Space and Extreme Environments (WiSEE), Orlando, FL, USA, Dec 2015. [14].

J. Fraire, **P. Madoery**, and J. Finochietto, “*Traffic-aware contact plan design for disruption-tolerant space sensor networks*,” Ad Hoc Networks. [15].

J. Fraire, **P. Madoery**, J. Finochietto, P. Ferreyra, and R. Velazco, “*Internetworking approaches towards along-track segmented satellite architectures*,” In 2016 IEEE International Conference on Wireless for Space and Extreme Environments (WiSEE), Aachen, Germany, Sep 2016. [16]

J. Barrientos, A. Ferral, L. Cara, J. Fraire, R. Velazco, **P. Madoery**, and P. Ferreyra, “*A segmented architecture approach to provide a continuous, long-term, adaptive and cost-effective glaciers monitoring system based on dtn communications and cubesat platforms*,” In 2017 1st IAA Latin American Symposium on Small Satellites: Advanced Technologies and Distributed Systems, Buenos Aires, Argentina, March 2017. [17]

J. Finochietto, J. Fraire, F. Raverta, **P. Madoery**, and R. Cherini, “*Enhancing Successful Data Delivery in Small Satellites*,” In 2017 1st IAA Latin American Symposium on Small Satellites: Advanced Technologies and Distributed Systems, Buenos Aires, Argentina, March 2017. [18].

J. Fraire, **P. Madoery**, J. Finochietto, and G. Leguizamón, “*An evolutionary approach towards contact plan design for disruption-tolerant satellite networks*,” Applied Soft Computing, 2017. [19].

J. Fraire, **P. Madoery**, and J. Finochietto, “*Contact Plan Design for Predictable Disruption-tolerant Space Sensor Networks*,” chapter 7, June 2017. [20].

J. Fraire, **P. Madoery**, F. Raverta, J. Finochietto, and R. Velazco, “*DtnSim: Bridging the gap between simulation and implementation of space-terrestrial DTNs*,” In 2017 6th International Conference on Space Mission Challenges for Information Technology (SMC-IT), Alcala de Henares, Spain, Sep 2017. [21].

P. Madoery, J. Fraire, F. Raverta, and J. Finochietto, “*On the trade-off analysis of centralized and distributed routing in delay tolerant satellitenetworks*,” In IV School on Systems and Networks (SSN 2018), Valdivia, Chile, October 2018. [22].

F. Raverta, **P. Madoery**, J. Fraire, and J. Finochietto, “*Modeling routing in scheduled delay tolerant networks under uncertainties*,” In IV School on Systems and Networks (SSN 2018), Valdivia, Chile, October 2018. [23].

J. Fraire, **P. Madoery**, A. Charif, and J. Finochietto, “*On route table computation strategies in delay-tolerant satellite networks*,” Ad Hoc Networks, 2018. [24]

1.5. Marco Teórico

1.5.1. Evolución de las Comunicaciones

Tal como se ilustra en la Figura 1.1 extraída de [25] y [26], las comunicaciones han evolucionado a lo largo del tiempo siguiendo dos modelos claramente diferenciados: por un lado, un **modelo síncrono o conversacional** en el que la fuente de datos debe ser capaz de establecer una comunicación extremo a extremo con el destino. Por otro lado, un **modelo asíncrono o epistolar** en el que la fuente no requiere de dicha conexión contemporánea con el destino y puede por ejemplo utilizar *mensajeros* que sean capaces de recibir los datos y transportarlos hasta que tengan una comunicación con el destino o con otro mensajero.

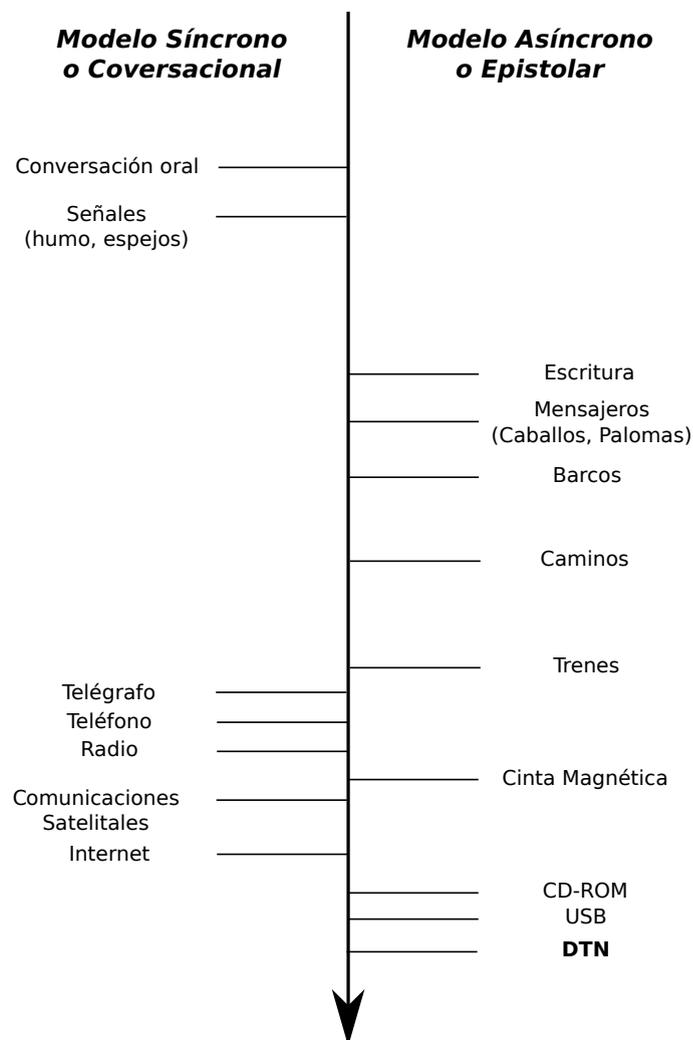


FIGURA 1.1: Evolución temporal de las tecnologías de comunicación

La comunicación síncrona surgió hace decenas o cientos de miles de años atrás mediante la manifestación de la comunicación oral y cara a cara entre los homínidos. En este tipo de comunicación, el ancho de banda es mucho mas grande que con cualquier otro sistema de comunicaciones. Sin embargo, el alcance tanto espacial como temporal resulta muy reducido. De esta manera y para superar estas limitaciones, se hizo necesario la utilización de un sistema asíncrono: la escritura. Esto permitió una mayor divulgación del conocimiento y no solo a distancias mas grandes, sino que también a lo largo del tiempo. En general, se puede decir que las condiciones cambiantes van afectando el uso en mayor o menor medida de estos distintos modelos de comunicación. Sin embargo, estos han existido siempre de forma complementaria en un balance que cambia continuamente de acuerdo a dichas condiciones:

- Con la expansión geográfica de una comunidad social, aumenta el tiempo de propagación de la señal (retardo) de un miembro de la comunidad a otro. Esto fomenta la utilización de sistemas de comunicación asíncronos.
- Con el despliegue de infraestructura de comunicación y el mejoramiento en la tecnología, se pueden reducir los retardos. Esto fomenta la utilización de sistemas de comunicación síncronos.

Continuando con nuestra descripción histórica, la domesticación del caballo resultó ser un avance sustancial en la comunicación asíncrona, ya que un mensajero podía transportar información a tasas de órdenes de magnitud superiores a las de un hombre a pié. Luego, la construcción de embarcaciones permitió extender el alcance abarcando de esta forma comunicaciones intercontinentales. Mas tarde, la invención de caminos pavimentados y de trenes generó un considerable incremento en el ancho de banda. Alrededor de 1833, la invención del telégrafo marcó el primer avance significativo en la tecnología de comunicaciones síncronas. Le siguió el teléfono en 1877 y la radio unos 20 años después. Nuevos avances en la comunicación asíncrona se produjeron en 1928 con la invención de la cinta magnética, el CD-ROM en la década del 80 y la memoria USB en el 2000. Mientras tanto, otros avances en la comunicación síncrona se producían gracias a los satélites de comunicaciones que permitían extender la radio telefonía a distancias intercontinentales y con anchos de banda mas grandes. Además, en 1971 Internet comenzó a brindar los servicios que conocemos uniendo a los rincones mas alejados del planeta.

Actualmente, nos encontramos con dos entornos problemáticos en términos de comunicaciones y que comparten algunas características en común:

- ❑ En primer lugar, estamos en una etapa de expansión geográfica mediante sondas, estaciones y satélites que tienen por objetivo explorar otros planetas y lugares del universo.
- ❑ En segundo lugar, hay redes y regiones del mundo que no están actualmente conectadas a Internet debido a la falta de rentabilidad económica que implica establecer la infraestructura de red necesaria.

Estos problemas pueden ser abordados utilizando la arquitectura DTN. En la próxima sección veremos su origen y fundamentos, y como puede aplicarse la misma en ambos entornos.

1.5.2. DTN: Redes Tolerantes a Retardos e Interrupciones

Resulta interesante el hecho de que una arquitectura de comunicaciones que fue inicialmente pensada para el espacio exterior pueda ser aplicada en comunicaciones terrestres. La relación entre ambos entornos quedará mas clara luego del breve resumen retrospectivo sobre DTN [27–29] que proporcionamos a continuación. Esto nos permitirá además sentar las bases para comprender las aplicaciones en nuestro contexto local y los avances propuestos por esta tesis.

1.5.2.1. Origen y Evolución de DTN

En 1973, Vint Cerf y Robert Kahn escribieron un trabajo sobre el protocolo *Transmission Control Protocol* (TCP) [30]. Años mas tarde este protocolo permitió la interconexión de dispositivos de comunicación alrededor del mundo en una red de redes llamada Internet. Luego, en 1997, Cerf junto a otros científicos del *Jet Propulsion Laboratory* (JPL) de la NASA comenzaron a pensar en el futuro y a visualizar como nos comunicaríamos si tuvieramos redes en distintos planetas, es decir, en una red interplanetaria. De esta manera, desde finales de los 90 comenzaron a colaborar en la definición de una arquitectura que llamaron *Internet Interplanetaria* (IPN) para adaptar los protocolos de

Internet a las misiones espaciales. Estos trabajos fueron realizados a través de un grupo de la *Internet Research Task Force* (IRTF) llamado *Interplanetary Internet Research Group* (IPNRG) que comenzó a considerar algunos problemas propios de este entorno. Las comunicaciones interplanetarias tienen la particularidad de estar sujetas a grandes retardos de propagación del orden de minutos, horas o días debidos a las enormes distancias que tienen que viajar las ondas electromagnéticas de radiofrecuencia. Además, deben enfrentar frecuentes interrupciones ocasionadas por la rotación de los planetas y el ocultamiento de las aeronaves que siguen una determinada dinámica orbital. Otras condiciones que complican aún más el panorama se deben a las altas tasas de error (BER), a los enlaces asimétricos y a la limitada energía que disponen las aeronaves para comunicarse. Estas condiciones llevaron a la conclusión general de que no bastaba con simples extensiones de la suite de protocolos TCP/IP, sino que era necesario el desarrollo de nuevas técnicas que serían luego concretadas en un protocolo llamado *Bundle Protocol* (BP).

Durante los años 2001 y 2002, se investigaron otras situaciones sujetas a condiciones similares de retardos e interrupciones en las comunicaciones. El propósito era extender el alcance para abarcar otras redes tales como las redes de sensores, satelitales, militares, submarinas, e incluso de comunidades viviendo en ambientes extremos o alejados [31]. De esta manera, en 2002 el IPNRG generalizó la arquitectura IPN y la rebautizó como *Delay/Disruption Tolerant Networking* (DTN). Cabe aclarar que la literatura utiliza la sigla DTN haciendo referencia tanto a la arquitectura de protocolos propuestos como a las redes que son vinculadas mediante esta arquitectura (*Delay/Disruption Tolerant Networks*). La Figura 1.2 ilustra esta integración de redes heterogéneas.

Además de esta generalización se creó un nuevo grupo llamado DTNRG que realizó entre sus primeros trabajos el refinamiento y la definición de la arquitectura DTN [32] y el protocolo Bundle [33]. Más tarde, en 2008, se creó *Licklider Transmission Protocol* (LTP), un protocolo punto a punto que provee confiabilidad mediante un mecanismo de retransmisión en enlaces caracterizados por grandes retardos. Actualmente, tanto el *Consultative Committee for Space Data Systems* (CCSDS), como la *Internet Engineering Task Force* (IETF), a través de un nuevo grupo llamado *Delay Tolerant Networking Working Group* (DTN WG), buscan estandarizar DTN de forma internacional para lograr acuerdos y compatibilidad entre las distintas implementaciones.

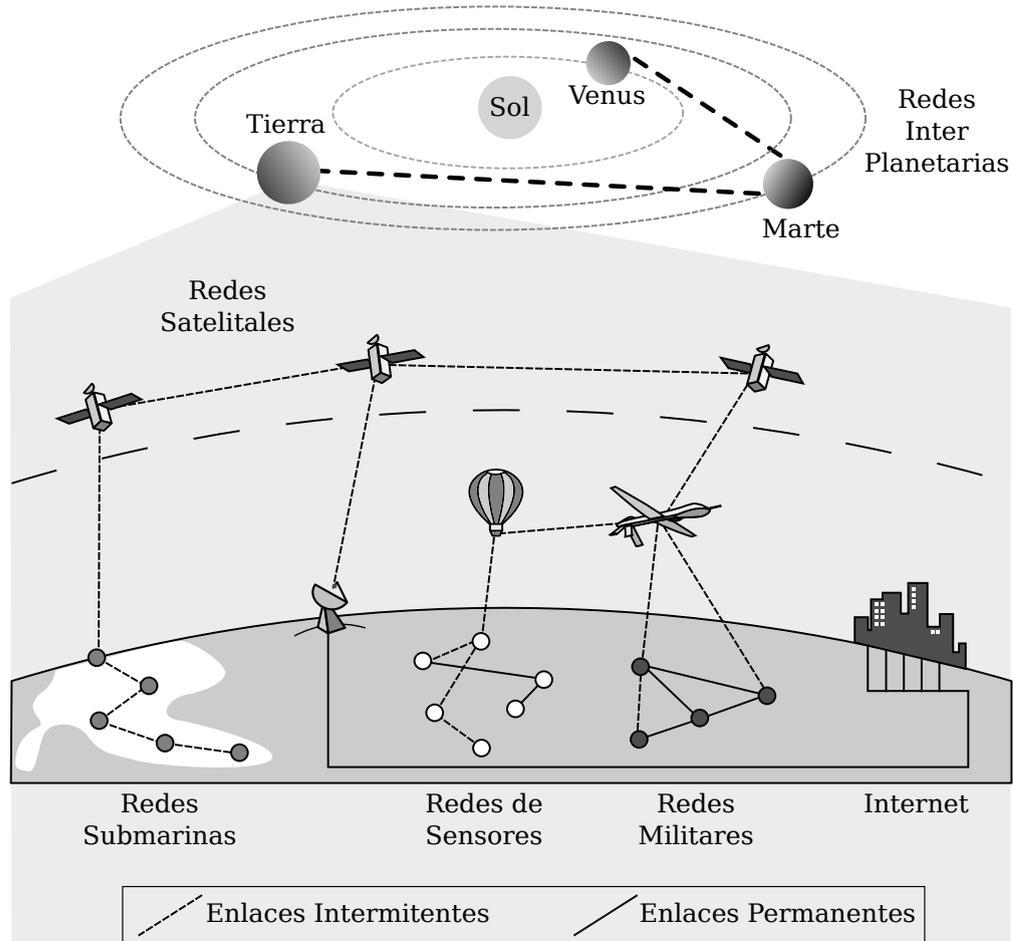


FIGURA 1.2: Integración de redes heterogéneas en una red DTN.

1.5.2.2. El Protocolo Bundle

El protocolo Bundle (BP) constituye la pieza fundamental de la arquitectura DTN. Es el encargado de adaptar los retardos e interrupciones entre y dentro de redes de distinta naturaleza, realizando traducciones entre los protocolos subyacentes de dichas redes. Para realizar esta tarea BP actúa como una capa de superposición (*overlay* en Inglés) por encima de otros protocolos de transporte, red y enlace, y utiliza un mecanismo conocido como almacenar-transportar-enviar (*store-carry-and-forward* en Inglés) que le permite almacenar los datos en una memoria persistente durante los momentos en que no hay conexión y realizar envíos cuando si la hay. Este mecanismo se basa en el modelo de comunicación asíncrona o epistolar que describimos en la Sección 1.5.1. Además, BP incorpora las llamadas capas de convergencia (Layer Adapters o CLAs en Inglés) [34, 35] que son las encargadas de adaptar los *paquetes bundle* (unidad de datos del protocolo BP) a las interfaces de los protocolos subyacentes.

La Figura 1.3 ilustra el funcionamiento de BP en una red satelital de 4 nodos compuesta por dos satélites y una Estación Terrena que tienen comunicaciones intermitentes, y un Centro de Control de Misión (MOC) que tiene comunicación permanente a través de Internet con la misma Estación Terrena. Aquí se visualiza como un paquete bundle es enviado desde el satélite 1 hasta el MOC atravesando diferentes capas de convergencia gracias al funcionamiento de BP como una capa de superposición sobre las mismas.

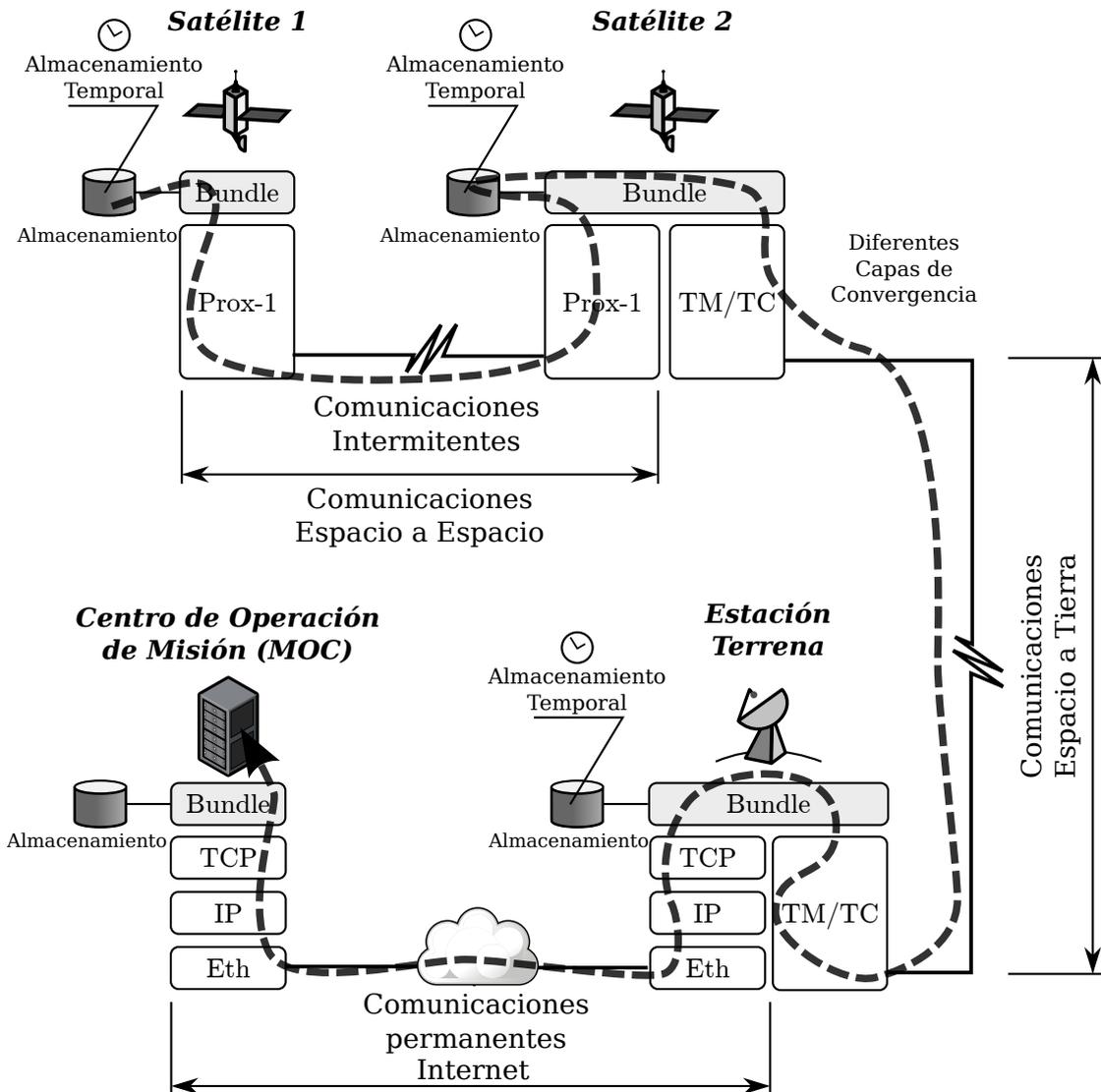


FIGURA 1.3: Protocolo Bundle como capa de superposición.

Tal como se describe en [33] y se ilustra en la Figura 1.4, los bundles pueden tener una longitud variable y están formados por al menos dos bloques de datos. El bloque primario contiene información sobre los identificadores del nodo origen y el nodo destino, el tiempo de vida y algunas banderas de procesamiento entre otros campos. En segundo

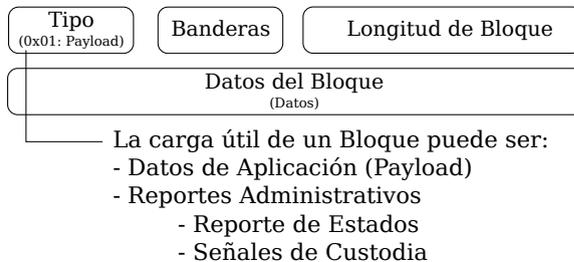
Bloque Primario



Banderas de Procesamiento de Bundle

- General:**
- Bundle es Fragmento
 - ADU es Administrativo
 - No Fragmentar
 - Transferencia con Custodia Requerida
 - El Destino es único
 - Confirmación Solicitada
- Clases de Servicios:**
- Bulk
 - Normal
 - Expedited
- Reportes de Estado:**
- Pedido de Reporte de Recepción
 - Pedido de Reporte de Aceptación de Custodia
 - Pedido de Reporte de Envío de Bundle
 - Pedido de Reporte de Entrega de Bundle
 - Pedido de Reporte de Borrado de Bundle

Bloque Canónico (Dato)



Banderas de Procesamiento de Bloque

- Replicar Bloque en cada Fragmento
- Transmitir Reporte de Estado si no se puede Procesar
- Borrar Bundle si no se puede Procesar
- Último Bloque
- Borrar Bloque si no se puede Procesar
- Bloque enviado sin Procesar
- El Bloque contiene referencia de EID

Bloque Extensión (Opcional)

FIGURA 1.4: Campos y formato de un Bundle de acuerdo a RFC5050.

lugar se puede tener uno o mas bloques canónicos que contienen datos de aplicación y otras banderas de procesamiento. Por último se dispone de la posibilidad de incluir bloques de extensión que cumplan con funciones específicas adicionales.

Dado que los bundles pueden ser enviados sobre enlaces con demoras significativas, la especificación del protocolo no obliga a utilizar confirmaciones de recepción (ACKs), sino que las mismas resultan opcionales. Por otra parte, la especificación permite la retransmisión de información perdida salto a salto por medio del uso de *transferencia con custodia*. Estas transferencias se acuerdan de antemano entre los nodos DTN. Si el

nodo receptor es capaz de hacerse responsable del dato recibido (es decir, transformarse en custodio), este envía una confirmación de aceptación de custodia liberando al nodo origen de mantener una copia local del bundle. En caso de que el nodo receptor no acepte la custodia, el nodo origen puede retransmitir el bundle a otro nodo. De esta manera, la transferencia con custodia permite mejorar la confiabilidad de entrega de datos, aunque no necesariamente la garantiza.

Además, BP provee 3 clases de servicio (Bulk, Normal y Expedited) que permiten encolar y desencolar bundles de acuerdo a estas prioridades. Una vez que un bundle es encolado para ser transmitido por medio de un enlace, el protocolo subyacente se podría encargar de fragmentarlo de manera acorde al enlace en cuestión. Sin embargo, también existe la posibilidad de que el agente Bundle fragmente los bundles proactivamente con el fin de adaptar los datos y confirmaciones de recepción de forma que la comunicación sea mas eficiente.

1.5.2.3. Esquemas de Enrutamiento

En general, las oportunidades de comunicación entre los nodos de una red DTN se pueden clasificar en *oportunisticas*, *probabilísticas* y *planificadas* [32], y existen distintas estrategias de enrutamiento para cada una de ellas.

Las comunicaciones oportunisticas suceden de forma aleatoria, por ejemplo cuando dos nodos entran dentro de su rango de cobertura de forma no planificada. Debido a la gran incertidumbre que se tiene en este tipo de comunicaciones, se han ideado estrategias de enrutamiento del tipo inundación (*flooding o epidémicos*) que realizan multiples copias de los bundles y los envían por cada comunicación posible [36]. Este tipo de mecanismos resulta ineficiente y hasta perjudicial en redes con recursos limitados y propensas a congestionarse, por lo que también se han desarrollado esquemas que restringen el número de copias tales como Spray-and-Wait [37] y que obtienen tasas de entrega aceptables con un uso de recursos arbitrariamente reducido de forma paramétrica.

Por otra parte, las comunicaciones probabilísticas son aquellas que exhiben una cierta distribución y que se pueden predecir con cierto nivel de confianza basándose en un historial de encuentros previos entre los nodos. Los esquemas de enrutamiento propuestos para aprovechar este tipo de comunicaciones buscan aprovechar el conocimiento de estas

distribuciones para enviar los paquetes de forma que tengan las mayores chances de llegar a sus respectivos destinos [38–40].

Finalmente, las comunicaciones interplanetarias y satelitales resultan predecibles y planificables con una gran exactitud gracias a la existencia de propagadores orbitales [41] y al control de la orientación de los nodos. Esquemas de enrutamiento como los desarrollados en [42, 43] permiten aprovechar dicha planificación para tomar decisiones mucho más eficientes que en otro tipo de redes. De hecho, el esquema *Contact Graph Routing* (CGR) especificado en [43], se encuentra en la frontera del estado del arte para redes planificadas debido a que ha recibido un gran número de aportes de la comunidad [44–47], y resulta de particular interés como veremos a lo largo de este trabajo.

1.5.2.4. Implementaciones

Actualmente existe un gran número de implementaciones de BP con diferentes características y objetivos operativos. Entre los más destacados, DTN2 [48] es la implementación de referencia de la arquitectura DTN, por lo que tiene el objetivo de incluir todos los componentes detallados en la misma. En consecuencia, DTN2 provee una arquitectura de software flexible para poder realizar experimentos, modificaciones, y eventualmente dar soporte a aplicaciones de BP en entornos operativos. Actualmente este software es portable a sistemas operativos basados en Linux y MAC OS X.

Por otra parte, el instituto *Institut für Betriebssysteme und Rechnerverbund* (IBR) desarrolló IBR-DTN [49] como una aplicación liviana, portable y altamente modular para uso en plataformas limitadas en procesamiento tales como los teléfonos celulares. En consecuencia, IBR-DTN incluye adaptadores de capas de convergencia (CLAs) para una importante variedad de protocolos tales como TCP, UDP, IEEE 802.15.4, etc. El código está diseñado para trabajar sobre Linux incluyendo distribuciones para dispositivos portátiles basados en Android y puede ser obtenido bajo licencia GNU de [50].

Recientemente, *Micro Planetary Communication Network* (μ PCN) [51] fue desarrollado en la *Universidad de Dresden* y está pensado para ser utilizado en redes satelitales con escasos recursos. Está especialmente diseñado en lenguaje C para sistemas operativos de tiempo real que se ejecuten utilizando la familia de microcontroladores Cortex STM32F4. Implementa además un protocolo de descubrimiento de vecinos (IPND) y un

adaptador de capa de convergencia para el protocolo de enlace AX.25 que es utilizado frecuentemente en pequeños satélites.

Además, existen otras implementaciones de menor relevancia en la comunidad como Postellation [52] o Java-DTN [53], que si bien están disponibles, en su mayoría no han pasado de un estadio de prototipo.

Finalmente, existe una implementación de suma relevancia para el uso de DTN en el entorno espacial llamada *Interplanetary Overlay Network* (ION) [54]. El software ION fue desarrollado por el *Jet Propulsion Laboratory* (JPL) de NASA en colaboración con la *Universidad de Ohio*, el *Johns Hopkins University Applied Physics Laboratory* (JHUAPL) y la *Universidad de Colorado* bajo la licencia *Berkeley Software Distribution* (BSD). Es probablemente el desarrollo del protocolo Bundle mas maduro de la comunidad para aplicaciones espaciales e incluye implementaciones de otros protocolos DTN tales como *Licklider Transmission Protocol* (LTP), *CCSDS File Delivery Protocol* (CFDP), y *Asynchronous Message Service* (AMS). Está diseñado para operar en sistemas embebidos de uso espacial soportando despliegues de alta velocidad y con pocos recursos como aquellos usados en sistemas robóticos en el espacio exterior. Sin embargo, también se desempeña adecuadamente en computadoras de escritorio tradicionales [55]. Está completamente escrito en lenguaje C, con comunicaciones entre tareas por medio de memoria compartida e incluye un sistema de memorias y bases de datos. Es portable a sistemas operativos como Linux, OS/X, FreeBSD, Solaris, uClibc, VxWorks, RTEMS, y Windows. Actualmente, la versión 3.6.1 se encuentra disponible en [56] con funcionalidades como Contact Graph Routing (CGR), compresión de cabecera de Bundle (Compressed Bundle Header Encoding (CBHE)) [57], protocolo de seguridad de bundle (Streamlined Bundle Security Protocol (SBSP)) [58], funciones de gestión de red (Delay Tolerant Network Management Protocol (DTNMP)) [59] y numerosos adaptadores de capas de convergencia a protocolos subyacentes tales como TCP [60], UDP y LTP [61], entre otros.

1.5.2.5. Experimentos en Órbita

A pesar del reciente nacimiento del concepto de redes DTN y la especificación e implementaciones del Protocolo Bundle, así como del elevado costo que caracteriza a las

misiones espaciales, al día de la fecha existen algunos experimentos del uso de esta tecnología en satélites reales cuyos resultados se describen a continuación.

Misión DINET

En Octubre y Noviembre del 2008 el JPL de NASA llevó a cabo la misión *Deep Impact Network Experiment* (DINET) [44] en la que se validó el uso de DTN con ION a bordo de la sonda de espacio profundo *EPOXY*, la cual tenía el objetivo inicial de estudiar el cometa *Tempel 1* para luego sobrevolar el cometa *Hartley 2* [62]. Una vez terminada la tarea de observación de EPOXY, la misión DINET tomó lugar sin interferir con el objetivo principal a una distancia aproximada de 21 millones de Km (70 segundos luz aproximadamente) de la Tierra.

Debido al tamaño de las antenas y la limitada energía disponible, la tasa de datos esperada a esta distancia rondaba desde los *256Kbps* a los *6Mbps* de bajada (ciencia y telemetría) y entre *1Kbps* y *2Kbps* de subida de datos (comandos). Con el fin de no interferir con la misión principal, ION fue cargado en la memoria de una computadora de a bordo secundaria la cual completaba una topología de un satélite orbital del tipo repetidor (relay) y dos objetos en la superficie de Marte (*Mars* y *Phobos*) simulados en tierra [44].

Los resultados fueron positivos al lograr mantener el sistema operativo por un total de 4 semanas totalizando 8 pasadas sobre la red de espacio profundo (DSN) de NASA traficando un total de 292 imágenes (por un total de *14,5MBytes*) desde los nodos con cargas útiles de adquisición de imagen. Se validaron capacidades de prioridades de tráfico y se verificó la entrega del mismo sin corrupciones. El uso de Bundle permitió operar sobre protocolos LTP en el caso de espacio profundo y TCP en las redes internas de NASA sin problemas de interoperabilidad, verificando y validando así la arquitectura DTN.

Misión UK-DMC

En el contexto de órbitas bajas (LEO) se destaca la misión *Disaster Monitoring Constellation* (DMC), llevada a cabo por *Surrey Satellite Technology Ltd.* (SSTL) de Reino Unido (UK) [63]. En particular, la misión está compuesta por varios satélites que tienen el objetivo de tomar imágenes de la superficie de la Tierra para luego bajarlas por medio del protocolo IP y un protocolo adaptado de TCP llamado *Saratoga* [64]. Saratoga actúa

como una capa de convergencia por debajo de BP y proporciona resultados similares a los brindados por LTP para el espacio profundo [65]. Sin embargo, Saratoga no ha tenido mayor trascendencia ya que la comunidad que se ha inclinado más por otras capas de convergencia.

Cronológicamente, la misión UK-DMC fue la primera prueba del protocolo Bundle desde el espacio ya que la misma fue lanzada en 2003 con un router CISCO a bordo (*Cisco router in Low Earth Orbit* o CLEO) [66]. La plataforma provee una tasa de transferencia de 8,1Mbps para bajada de ciencia y telemetría y una tasa de 9,6Kbps para subida de comandos. Sin embargo, dado que el espacio de memoria para el software del equipo de CISCO era limitado, se usó el stack DTN2 [48] en lugar del software ION [54]. Los científicos que realizaron este experimento documentaron algunos problemas en el planteo de BP [63] que fueron luego tratados en nuevas versiones del protocolo. Se propuso además una alternativa basada en el protocolo de transferencia de contenido web HTTP [67]. Sin embargo esta propuesta data del 2008, momento desde el cual no ha logrado captar un interés significativo en la comunidad DTN.

Misión ISS

Los experimentos previos en DINET y UK-DMC demostraron la factibilidad de utilizar BP como una capa de superposición para vincular las redes terrestres y espaciales. Esto motivó a que desde el 2008 hubiera un esfuerzo por utilizar DTN de una forma más intensiva y permanente. De esta manera, en junio del 2009 se instaló ION en la *Estación Espacial Internacional* (ISS) con el objetivo de comunicar datos de ciencia desde su cámara de experimentos CGBA5 al Centro de Control de la Misión en Colorado. La ISS se comunica con el Centro de Control por medio de relevos (handovers) de satélites geostacionarios (TDRSS). A pesar de las frecuentes interrupciones, BP ha demostrado la capacidad de transmitir datos de forma eficiente, por medio de ACKs selectivos, y confiable, gracias a la transferencia con custodia. Asimismo, este tipo de experimentaciones permite afinar detalles de implementación que resultan de suma utilidad en el proceso actual de estandarización en el que se encuentran estos protocolos.

1.5.2.6. Frontera del Estado del Arte

El campo de estudio de este trabajo encuentra aplicación en las redes DTN planificadas conformadas por constelaciones satelitales. A pesar de que al día de la fecha se cuenta con numerosas implementaciones en software y se han llevado a cabo las misiones experimentales descritas, no se han desarrollado aún misiones satelitales que aprovechen las ventajas potenciales de la arquitectura en un entorno donde los paquetes puedan realizar múltiples saltos para llegar a su destino. Misiones de este tipo podrían habilitar el uso de enlaces intersatelitales que permitirían a su vez optimizar el uso de las comunicaciones espaciales. La falta de experimentación en estos aspectos derivó en que no se hayan tenido en cuenta hasta ahora algunos aspectos que resultan claves. Dentro de las pruebas de emulación realizadas en la Tierra [54, 68, 69], todas determinan y utilizan directamente el plan de contactos asumiendo que los nodos siempre contarán con los recursos necesarios para implementar dichas comunicaciones. Sólo existen a nivel académico algunos aportes en mecanismos para optimizar estos planes en término de costo [70] o confiabilidad [71], pero ninguno de ellos contempla las limitaciones de recursos en los nodos.

En consecuencia, el grupo de investigación del cual forma parte el doctorando contribuyó en primer lugar con una serie de mecanismos de diseño del plan de contactos (resumidos en el Capítulo 2 a partir de [1]) que contemplan la limitación de recursos hasta ahora no considerada por otros trabajos.

Por otra parte, el desarrollo de estos trabajos destacó un nuevo aspecto hasta ahora no tenido en cuenta y que constituye el aporte principal de esta tesis: *la interacción que existe entre el plan de contactos y el esquema de enrutamiento de datos influye directamente en la congestión de la red, en la incertidumbre y en la escalabilidad*. Estos aportes constituyen un avance en el estado del arte de las redes DTN planificadas en general y encuentran aplicación particular en la Arquitectura Segmentada, que constituye nuestra principal motivación y que describimos continuación.

1.5.3. La Arquitectura Segmentada para la Observación Terrestre

1.5.3.1. La Teledetección

La observación de la Tierra desde el espacio resulta ser un mecanismo sumamente eficiente para lograr una mejor gestión de nuestros recursos. Las principales ventajas con respecto a otros métodos se ven plasmadas en una mayor resolución espacial, temporal, radiométrica y espectral, lo que deriva en un acceso a información adecuada y de manera oportuna. Estas capacidades han sido entendidas por la Comisión Nacional de Actividades Espaciales (CONAE), que luego de elaborar un *Plan Espacial Nacional* (PEN) [72], ha llevado con éxito numerosas misiones satelitales de observación terrestre tales como la serie SAC y Saocom. Sin embargo, como se puede observar en la Figura 1.5, la necesidad de un mayor número de instrumentos y de mayores resoluciones deriva en la construcción de satélites mas complejos y de mayor peso, lo que a su vez impacta en un incremento de los costos y en la extensión de los ciclos de desarrollo.

1.5.3.2. El Entorno Espacial

Debido a la agresividad del entorno espacial, se deben considerar numerosos aspectos que pueden desencadenar tanto fallas transitorias como permanentes. Entre ellos se destacan las colisiones con objetos orbitantes, la rotura de componentes debido a las vibraciones

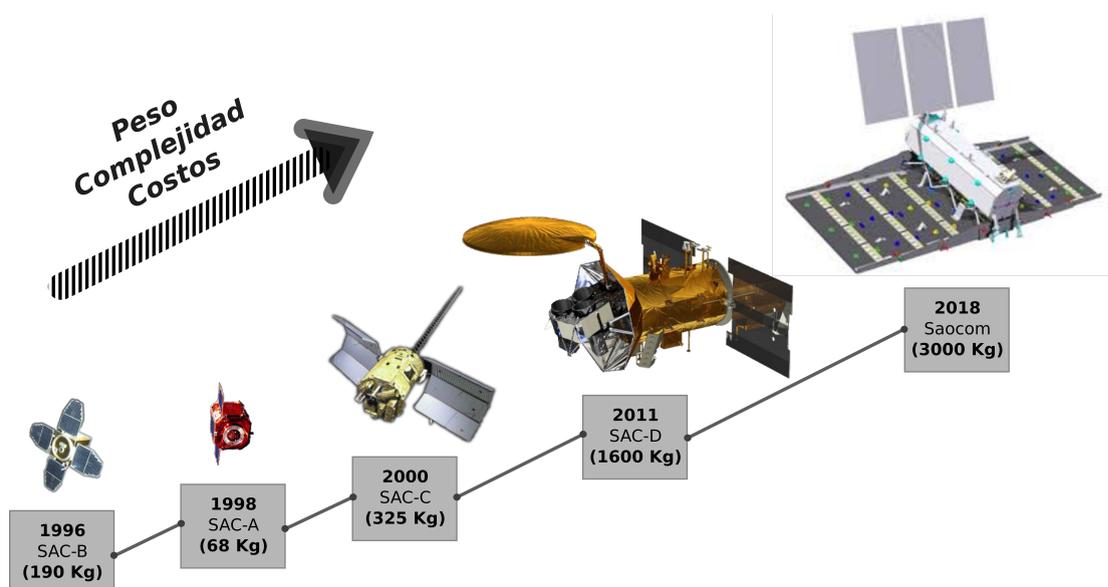


FIGURA 1.5: Misiones de observación de la Tierra de CONAE

del lanzador, la alta variabilidad de temperatura y los efectos de la radiación. Este último fenómeno resulta de particular interés ya que los satélites se basan en gran medida en circuitos electrónicos, por lo cual los efectos de la radiación deben ser especialmente considerados para garantizar los grados de confiabilidad y seguridad requeridos en cada misión [73].

Para hacer frente a este problema el enfoque usualmente adoptado por la industria espacial consiste en la inclusión de componentes redundantes tanto a nivel de software como de hardware. Sin embargo, dicha redundancia genera un aumento en la complejidad del sistema, lo que a su vez deriva en nuevos modos potenciales de fallas y que resultan muy difíciles de anticipar debido a la compleja interacción entre las partes del sistema. Como consecuencia de este aumento en la complejidad, se tienden a imponer mayores márgenes de confiabilidad para evitar pérdidas catastróficas, lo que ocasiona nuevamente la necesidad de redundar componentes para cumplir con dichos márgenes. Este fenómeno es conocido en la literatura como espiral de la muerte de costo-complejidad [74] y una manera de evitarlo consiste en distribuir dicha complejidad en partes independientes del sistema. Esto es resultado de que la complejidad no sea en realidad una función del número de componentes de un sistema, sino que mas bien de la densidad de los mismos. Si tomamos como ejemplos ilustrativos a *Internet* y al *transbordador espacial* como los productos de ingeniería mas complejos desarrollados por el hombre, podemos apreciar que aunque el primero contiene un mayor número de componentes, el segundo cuenta con un mayor registro de fallas catastróficas.

1.5.3.3. Los Pequeños Satélites

En la actualidad las constelaciones de pequeños satélites compiten o incluso superan las capacidades de los tradicionales satélites monolíticos aunque a una fracción del costo. Sin embargo, las tecnologías utilizadas no son fundamentalmente diferentes en uno y otro caso. La diferencia radica principalmente en la velocidad de adopción de nuevas tecnologías y especialmente de los componentes comerciales (COTS) para lograr ciclos rápidos de desarrollo con una alta utilidad y a un bajo costo [75]. Los pequeños satélites se han convertido en una nueva tendencia y resultan ser catalizadores de nuevas aplicaciones y modelos de negocio, de la misma forma que lo han hecho en su momento las computadoras y teléfonos portátiles. En particular, estan cambiando el sensado remoto

de la Tierra y la industria de las comunicaciones digitales a través de un incremento en la vinculación de Internet de las Cosas (IoT) con grandes almacenes de datos (big data) y la extracción de información relevante a partir del uso de inteligencia artificial (AI). Empresas como Planet, Boeing, SpaceX, Telesat y OneWeb constituyen ejemplos de emprendimientos que ya han lanzado o buscan lanzar grandes constelaciones de pequeños satélites y cuyo éxito se verá reflejado en un futuro próximo.

Por otra parte, el mayor impedimento para el rápido desarrollo de este sector lo constituye el costo y la disponibilidad de un lanzador orbital. El lanzamiento constituye uno de los costos mas elevados que tiene una misión. Además, resulta fundamental para los modelos de negocio de los pequeños satélites poder ajustarse a un cronograma determinado.

1.5.3.4. La Arquitectura Segmentada

La CONAE ha diseñado el nuevo Plan Espacial Nacional de forma tal de solucionar los problemas actuales (peso, complejidad, costos) y de aprovechar las nuevas tendencias (sistemas de pequeños satélites) adaptándolas a un contexto local. Los dos pilares fundamentales de dicho plan lo constituyen la Arquitectura Segmentada y el proyecto Tronador.

La serie de cohetes Tronador se corresponde con etapas de un lanzador nacional pequeño y de bajo costo capaz de depositar satélites en órbitas bajas y polares, utilizadas para observar territorios en altas latitudes como es el caso argentino. Al día de la fecha, el proyecto está en vías de desarrollo y ya se han lanzado tres vehículos experimentales denominados VEx. Se espera que una vez constituido como tal, el programa Tronador sea capaz de realizar entre 5 y 10 lanzamientos de satélites al año de entre 250 y 350 Kg.

Por otra parte, la Arquitectura Segmentada busca articular e integrar una red cooperativa de satélites pequeños (llamados segmentos) puestos en órbita por medio de lanzamientos sucesivos de cohetes Tronador. De esta manera, cada segmento podría compartir recursos por medio de una red inalámbrica en órbita. El eje de la idea es que varios segmentos pequeños, más fáciles y rápidos de fabricar, logren generar en conjunto un producto equivalente o mejor que el de un gran satélite monolítico. Por ejemplo,

varios segmentos con cámaras ópticas de baja resolución podrían cooperar para generar una única imagen de mayor resolución similar a la que podría obtenerse con un satélite de gran tamaño y peso. Además de cumplir el objetivo de acceder de manera rápida y autónoma a la información espacial, el conjunto Tronador-Segmentada aporta otras ventajas importantes que son ilustradas en la Figura 1.6 y descriptas a continuación:

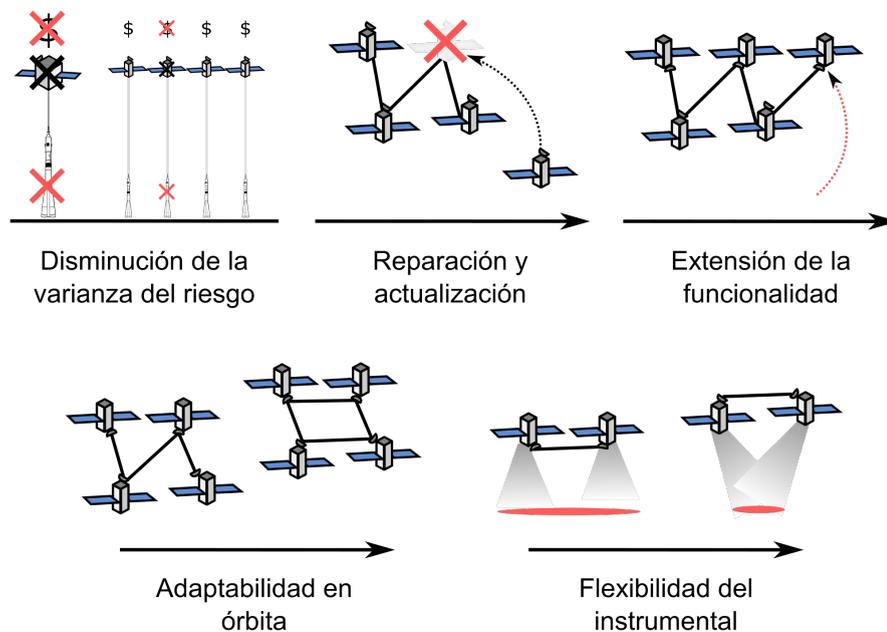


FIGURA 1.6: Beneficios de la Arquitectura Segmentada.

- *Disminución de la varianza del riesgo en el lanzamiento:* Dividir un lanzamiento mayor en mas lanzamientos de menor magnitud permite que un fallo no implique la pérdida de la misión, si no que solo una degradación de su rendimiento.
- *Reparación y actualización:* El fallo de un satélite tampoco supone el fin de la misión y podría ser rápidamente reemplazado. Por otro lado, segmentos obsoletos podrían igualmente ser actualizados con otros de mayores capacidades. La vida útil del sistema resulta así tan extensa como se lo requiera.
- *Extensión de funcionalidad:* La funcionalidad del sistema puede extenderse con el agregado de un nuevo segmento a la red que incorpore un nuevo instrumento mejorando la capacidad de observación u otros recursos de valor para la red tales como memoria, procesamiento o capacidad de descarga.

- *Adaptabilidad en órbita:* En principio, y dependiendo de las capacidades de propulsión y orientación de los segmentos lanzados, la constelación podría acomodarse y adaptarse a nuevas formaciones para satisfacer diferentes necesidades de misión.
- *Flexibilidad del instrumental:* La Arquitectura Segmentada posibilita configuraciones inexistentes en satélites monolíticos. Un mismo objetivo puede ser observado desde diferentes ángulos (simulando mayor apertura) o bien se puede cubrir una mayor superficie de observación (mayor cobertura). Además, la distribución espacial de segmentos incrementa las tasas de revisita a un mismo punto sobre la tierra.

Si bien la Arquitectura Segmentada proporciona grandes beneficios, se deben resolver también numerosos desafíos antes de poder implementarla en el espacio:

- *Cargas útiles segmentadas:* Es fundamental desarrollar la inteligencia que permita integrar observaciones de pequeños instrumentos para obtener una de mayor calidad. El caso particular de pequeños radares cooperando en varios segmentos (a.k.a. antena partida) supone un gran desafío que hoy motiva a muchos científicos locales. Aproximaciones similares deberán ser aplicadas para el caso de servicios de telecomunicaciones.
- *Control de vuelo en formación:* Los diferentes segmentos necesitarán mantener una formación en el espacio. Esto puede implicar el uso de pequeños propulsores disparados autónomamente por medio de una inteligencia distribuida basada en datos muy precisos de sensores de posición y orientación. Evitar colisiones resulta fundamental.
- *Chatarra espacial:* La Argentina es miembro activo de comités internacionales que acuerdan políticas para controlar la generación de chatarra espacial. Para evitar dejar segmentos orbitando la Tierra por años se deberá contar con sistemas que lleven a los mismos a desintegrarse controladamente en la atmósfera hacia el fin de su vida útil.
- *Distribución de tareas:* Los segmentos deberán coordinar entre ellos tareas de adquisición de datos de observación así como de transmisión, almacenamiento y procesamiento. Estas decisiones requieren del desarrollo de un sistema de software en red adaptado al espacio.

- ***Comunicaciones en red:*** El sistema se basará en tecnología de comunicaciones inalámbricas en red entre los segmentos. Como se introdujo en la Sección 1.5.2, DTN resulta un enfoque adecuado para ser aplicado en la Arquitectura Segmentada. Sin embargo, el mismo se encuentra aún en una etapa de maduración y estandarización por lo que se hacen indispensables estudios como los desarrollados en esta investigación. Particularmente, nos centraremos en la planificación de las comunicaciones y en los esquemas de enrutamiento. Como veremos en el Capítulo 2, la planificación de las comunicaciones resulta necesaria para poder gestionar adecuadamente la limitada energía que es recolectada por medio de paneles solares. Además, resulta fundamental para evitar las interferencias, conflictos en el uso de recursos y optimizar la entrega de datos. Por otra parte, dicha planificación está íntimamente ligada con los esquemas de enrutamiento que son ejecutados de manera distribuida, por lo que resulta clave diseñar ambos mecanismos y su interacción de manera tal que se evite la congestión, se superen las incertidumbres y se logren comunicaciones eficientes y escalables.

Capítulo 2

Comunicaciones Planificadas y Mecanismos de Enrutamiento

2.1. Introducción

Las comunicaciones de datos han sido tradicionalmente adaptadas a la infraestructura permanentemente conectada que proporciona Internet. En este tipo de redes, el uso de información tal como enlaces activos, rutas o direcciones es usualmente descubierta de forma conversacional por los nodos cuando ésta es requerida. Sin embargo, la obtención de la misma información puede resultar imposible o ineficiente en una red con retardos y/o interrupciones como las que presentan las redes satelitales. El enfoque propuesto para estas DTNs planificadas consiste en configurar de antemano a los nodos con la información necesaria, como así también con los tiempos en que dicha información resulta relevante.

En este capítulo repasaremos el funcionamiento de las redes DTN planificadas haciendo especial hincapié en como utilizan la información sobre la red. En particular, definiremos los conceptos de topología de contactos, plan de contactos y algoritmos de enrutamiento, que serán de suma utilidad para comprender los mecanismos de evitación de congestión propuestos posteriormente.

2.2. Comunicaciones Planificadas

2.2.1. Definición de Nodo e Identificador

Se denomina nodo DTN a toda entidad capaz de enviar y recibir bundles, es decir, que debe disponer de una implementación de la capa Bundle Protocol (BP). Los nodos pueden ser miembros de los llamados *endpoints*. De esta manera, un endpoint es un conjunto de nodos, y un bundle es considerado entregado a un endpoint cuando un mínimo subconjunto de los nodos registrados en ese endpoint ha recibido el bundle sin error. Además, los endpoints se identifican según los llamados *endpointIDs* (EIDs) que son nombres expresados según la sintaxis URI definida en [76].

Por otra parte, si bien un satélite podría implementar un número variable de nodos DTN, en este trabajo haremos uso de un único nodo por satélite, por lo que los términos nodo y satélite serán intercambiables.

2.2.2. Definición de Contacto

Dentro del amplio abanico de redes DTN existentes, aquellas formadas por satélites tienen la particularidad de que sus trayectorias y orientaciones resultan predecibles por medio de propagadores orbitales [41] y sistemas de control de órbita y actitud (Attitude and Orbit Control System (AOCS)). Además, se puede sumar información sobre los sistemas de comunicaciones y antenas presentes en cada satélite que ayuden a determinar de antemano las oportunidades de comunicación entre los nodos orbitantes.

En general, cada oportunidad de comunicación se conoce en la literatura como *contacto* y está definida por un intervalo de tiempo durante el cual un nodo transmisor puede enviar datos a un nodo receptor. Notar que según la definición, un contacto es directivo, por lo que para modelar una comunicación *full duplex* necesitaríamos de dos contactos, uno en cada sentido. La Figura 2.1 ilustra dos pares de contactos. El primer par entre dos satélites (*Inter Satellite Link o ISL*) y el segundo entre un satélite y una estación terrena (*Earth Satellite Link o ESL*). Cada contacto se caracteriza por un identificador de nodo transmisor (i), un identificador de nodo receptor (j), un tiempo de inicio (t_{ini}), un tiempo de finalización (t_{fin}), y otros datos de relevancia que varían según la implementación particular. Por ejemplo, en el caso del *stack* de protocolos implementados en

el software ION el contacto también incluye la tasa de datos promedio que se puede lograr en ese intervalo. Sin embargo también se podría considerar información adicional tal como los identificadores de las antenas, las modulaciones a utilizar, potencias de transmisión necesarias, la distancia promedio entre los nodos, etc. En general, utilizaremos la notación $cnt(i, j, t_{ini}, t_{fin})$, un identificador numérico (id) o directamente la capacidad para referirnos a un contacto específico.

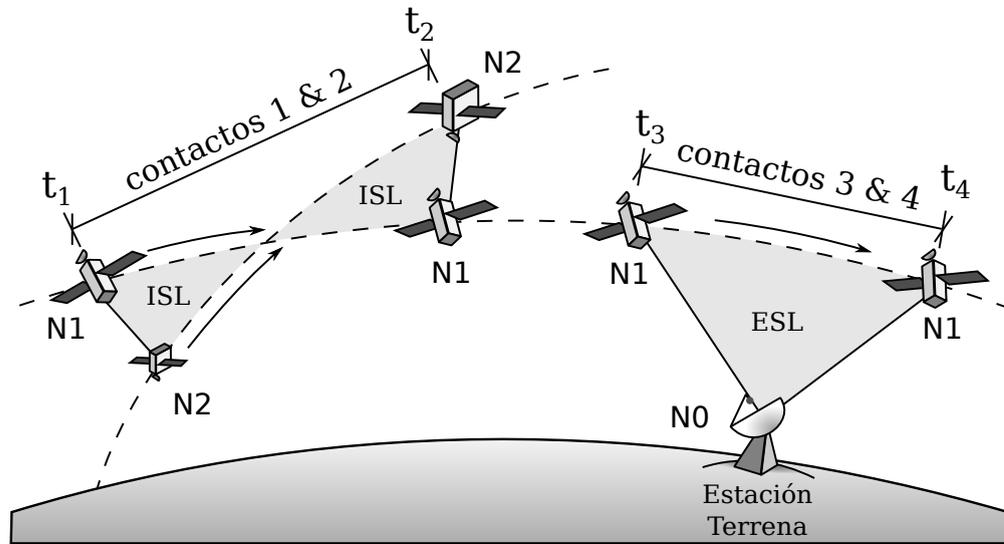


FIGURA 2.1: Contactos variables en el tiempo

2.2.3. Definición de Topología y Plan de Contactos

Llamamos *Topología de Contactos* (TC) a la agrupación de todos los contactos que resultan *físicamente factibles* dentro de un determinado horizonte de tiempo T . Dicho horizonte de tiempo está en el orden de los días o semanas y se define usualmente en función de la precisión de los modelos de propagación y de la anticipación requerida por el Centro de Control de Misión (MOC).

Por otra parte, llamamos *Plan de Contactos* (PC) a un subconjunto de contactos de la topología de contactos que es escogido de forma tal de cumplir con ciertas restricciones de recursos que son explicadas en la siguiente sección. El PC constituye el conjunto de oportunidades de comunicación que los nodos intentarán llevar a cabo en los momentos especificados. La información del PC se utilizará además como entrada para calcular las rutas que deberán seguir los datos generados localmente o en tránsito.

Además, llamamos *Diseño del Plan de Contactos* al procedimiento de obtener el PC a partir de la TC mediante la adaptación de la misma para cumplir con restricciones adicionales que no son tenidas en cuenta en primer lugar. A continuación proporcionamos una breve descripción de las restricciones de recursos mas importantes.

2.2.4. Restricciones de Recursos

En general, las plataformas satelitales se diseñan con el objetivo de optimizar el peso, volumen y consumo de manera tal de minimizar los altos costos asociados al lanzamiento y operación de las mismas. En consecuencia, los satélites suelen estar restringidos en cantidad de transponders (equipos de comunicaciones), disponibilidad de potencia (obtenida a través de paneles solares), cantidad y tamaño de antenas, combustible disponible para propulsión, entre otros que limitan la posibilidad de concretar contactos que resultan físicamente factibles.

De acuerdo a [77], se pueden clasificar estas restricciones en dos grupos: uno representando aquellas que hacen que un contacto particular en un tiempo dado se vuelva directamente inviable, y otra que limita la cantidad de contactos que un nodo puede mantener simultáneamente. Las primeras se denominan *Restricciones de Tiempo y Zona*, y las segundas, *Restricciones de Recursos Concurrentes*.

2.2.4.1. Restricciones de Tiempo y Zona

En general las restricciones de tiempo y zona prohíben la emisión de radio-frecuencia en un área geográfica determinada y en un tiempo determinado ya sea para evitar interferencias no deseadas o por razones políticas o diplomáticas específicas de las agencias espaciales. Dado que las constelaciones de baja órbita (LEO) orbitan prácticamente sobre toda la superficie terrestre, cumplir con todas las regulaciones internacionales puede resultar un desafío complejo. La Unión Internacional de Telecomunicaciones (ITU) [78] es el organismo encargado de establecer las normativas correspondientes. De acuerdo a las normativas de la ITU, las interferencias no sólo se limitan por la cantidad de potencia interferente que llega al satélite interferido, si no que también por el porcentaje de tiempo mensual que esta puede superar un determinado valor umbral [79]. En consecuencia, en lugar de anular contactos directamente, también es válida la opción de elegir

posibles combinaciones de utilización de los mismos de manera tal de no infringir las leyes internacionales.

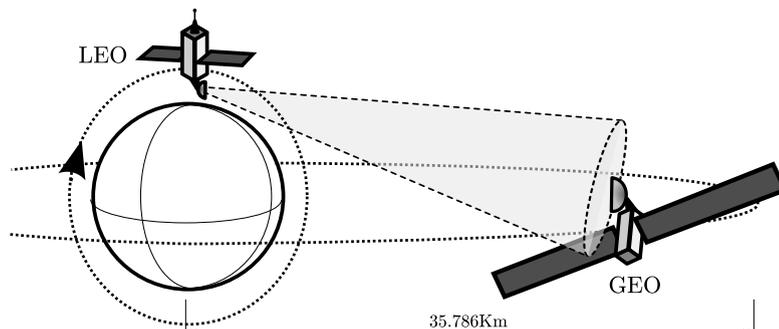


FIGURA 2.2: Interferencia ocasionada a un satélite GEO

2.2.4.2. Restricciones de Recursos Concurrentes

Las restricciones de recursos concurrentes se refieren a limitaciones por arquitectura o potencia de la plataforma satelital. Probablemente la restricción mas común de este tipo es aquella que surge cuando el patrón de radiación de una antena de un satélite permite alcanzar dos o mas nodos destino como se ilustra en la Figura 2.3 a). En este caso, los esquemas de múltiple acceso podrían automáticamente negociar y compartir el acceso al medio de las comunicaciones. Sin embargo, estos resultan ineficientes para las distancias en que se requieren efectuar las comunicaciones satelitales (del orden de los 1000 Km). Por el contrario, un correcto diseño del plan de contactos permite el uso de enlaces punto a punto mas eficientes que realmente utilicen todo el espectro disponible sin perder capacidad en negociaciones innecesarias y que resultan además evitables en comunicaciones predecibles/planificables como es el caso de las redes espaciales.

Por otro lado, cuando un satélite se diseña para poder utilizar enlaces intersatelitales desde diferentes direcciones, suele ser necesario colocar mas de una antena sobre la estructura del satélite. Por ejemplo, considerando la Figura 2.3 b), un conmutador de potencia podría permitir la utilización de diferentes elementos radiantes en distintos tiempos sin la necesidad de multiplicar los transponders. En consecuencia, si bien en la topología de contactos se encontrarían presentes todos los contactos potenciales (uno por cada antena), el diseño del plan de contactos implicaría tomar una decisión de por cual antena transmitir con ese único transponder en un tiempo dado.

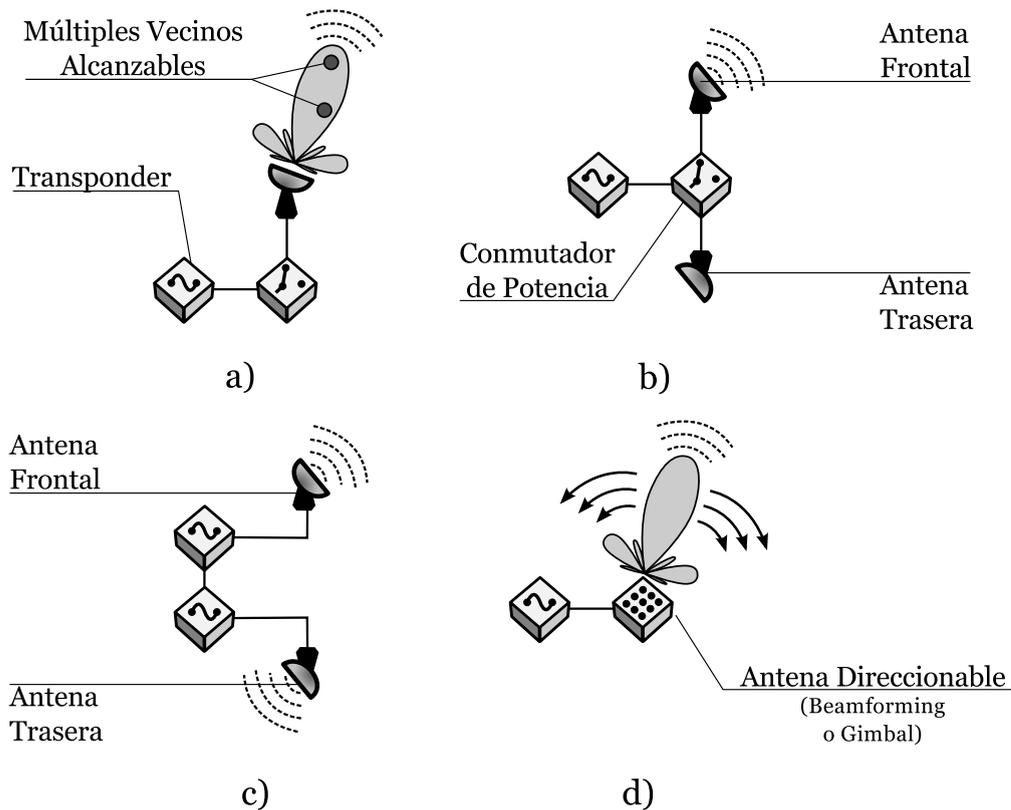


FIGURA 2.3: Arquitectura de plataformas satelitales con a) múltiples posibles vecinos, b) un conmutador de potencia, c) dos equipos de comunicaciones, y d) una antena direccional electrónica o mecánicamente

Una arquitectura mas compleja se ilustra en la Figura 2.3 c), donde dos contactos simultáneos pueden ser implementados en el plan de contactos siempre y cuando la potencia de la plataforma lo permita.

Por último, al considerar antenas direccionables como se muestra en la Figura 2.3 d) se abre un interesante panorama que da lugar al diseño del plan de contactos para antenas con haz dinámico. En general, existen antenas que se pueden apuntar mecánicamente (*gimbal*) con el fin de obtener la máxima ganancia del lóbulo de radiación en una zona de interés. Por otro lado, también existen antenas dirigibles electrónicamente (*beamforming*) que evitan realizar un movimiento mecánico y por ende el desgaste y limitación de su vida útil. En general, este tipo de antenas implica necesariamente una selección de contacto en caso de que mas de uno sea factible. En efecto, una estrategia válida para la conformación de la topología de contacto de estos sistemas es considerar que la antena tiene la máxima ganancia en todas las direcciones dentro del rango de operación de la misma. Luego, el diseño del plan de contactos debería definir una dirección específica de apuntamiento en función de la conveniencia de un contacto particular a nivel sistema.

En resumen, las restricciones de recursos concurrentes implican un proceso de selección de contactos que deriva en problemas combinatorios poco triviales para grandes constelaciones y/o largos intervalos de topología.

2.2.5. Modelado de la Topología y el Plan de Contactos

Con el objetivo de realizar el diseño del plan de contactos a partir de la topología, resulta útil contar con modelos que permitan representarla adecuadamente y ejecutar algoritmos que permitan modificarla de forma eficiente. A continuación se describen dos formas de modelado que resultan equivalentes en cuanto a la información que proporcionan, pero cuya conveniencia depende del contexto dado.

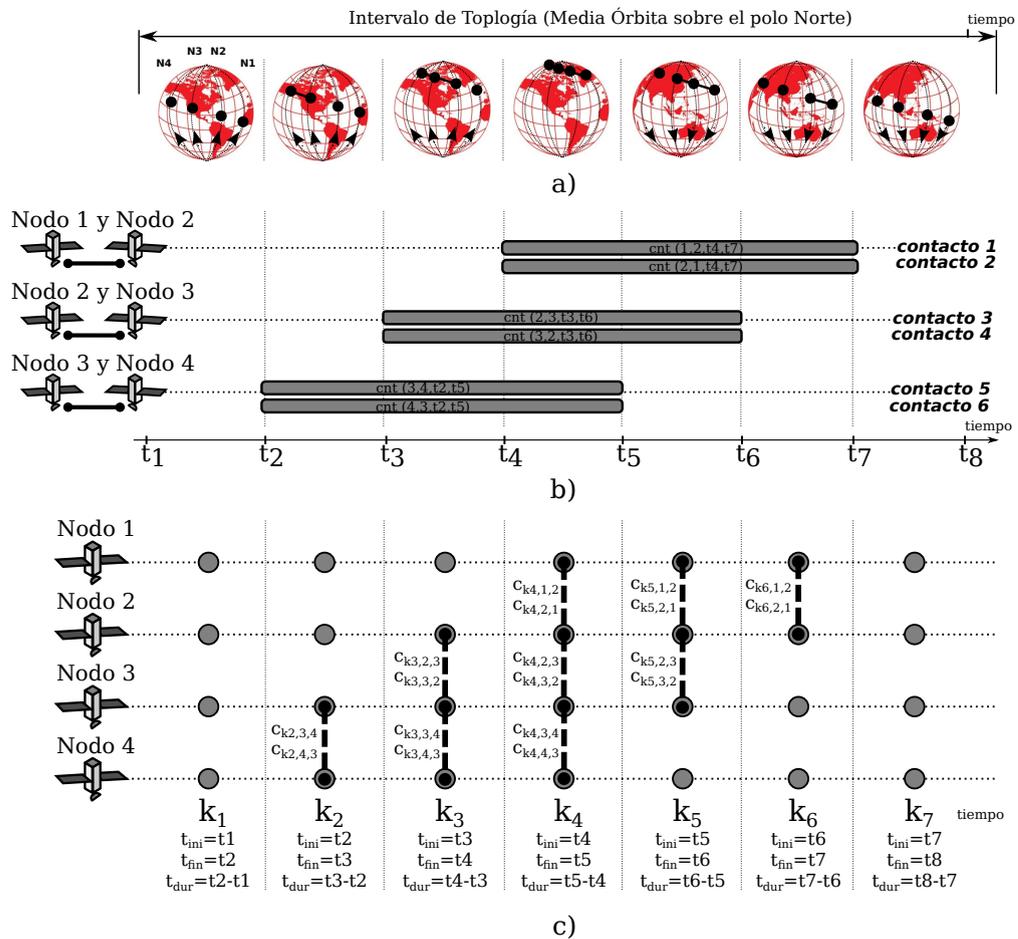


FIGURA 2.4: Modelos de la topología y el plan de contactos: a) trayectorias orbitales, b) lista de contactos, c) máquina de estados finitos (FSM)

2.2.5.1. Modelado como Lista de Contactos

La Figura 2.4 a) ilustra posibles contactos entre satélites que siguen distintas trayectorias orbitales. En 2.4 b) se ilustra una forma de modelar estos contactos que consiste simplemente en listarlos y proporcionarles ciertos atributos tales como los identificadores de los nodos intervinientes y los tiempos de inicio y fin. Esta forma de representación resulta eficiente, por un lado, cuando los nodos utilizan el plan de contactos con el objetivo de llevar a cabo dichos contactos, y por otro, cuando los nodos necesitan ejecutar los algoritmos de enrutamiento. Un ejemplo en donde los nodos reciben el plan de contactos con esta representación se da en el software ION [56].

2.2.5.2. Modelado como Máquina de Estados Finitos

Por otra parte, algunos algoritmos de diseño del plan de contactos requieren tomar decisiones cada vez que hay un inicio o final de cualquier contacto, o incluso en puntos intermedios de los contactos. Esto lleva a la necesidad de modelar la topología como una máquina de estados finitos (*Finite State Machine (FSM)*) en donde el intervalo de la topología es discretizada en K estados. De esta manera, cada estado resulta válido durante un intervalo de tiempo $[t_{ini}, t_{fin}]$, tiene una duración t_{dur} , y se representa con un grafo cuyos vértices simbolizan nodos y cuyos arcos simbolizan porciones de contactos o contactos completos. Como resultado, un solo contacto puede abarcar varios estados k_n , y por cada inicio y/o final de un contacto hay una evolución de un estado k_n a un estado k_{n+1} . Notar además que la notación $C_{k_n, i, j}$ representa la capacidad del arco entre los nodos i y j solo durante el estado k_n . Además, en caso de que resulte necesario que los algoritmos de diseño tomen decisiones en puntos intermedios de un estado, es decir con una granularidad mas fina, éste puede fácilmente subdividirse en dos o mas estados.

2.2.6. Diseño del Plan de Contactos

Como mencionamos anteriormente, el diseño del plan de contactos es el procedimiento algorítmico utilizado para seleccionar (o descartar) un subconjunto de contactos de la topología, de forma tal que se cumpla con restricciones de tiempo y zona, y restricciones de utilización concurrente de recursos, entre otras restricciones posibles.

Tradicionalmente, el problema de diseño del plan de contactos ha recibido escasa atención de la comunidad dado que en general se ha asumido que todos los contactos de la topología pueden pertenecer al plan de contactos. Sin embargo, el hecho de considerar configuraciones más complejas de hardware (transponders, antenas, etc), deriva en que el diseño del plan de contactos, además de ser necesario, resulte determinante en el rendimiento futuro de la red una vez que los contactos son llevados a cabo por los nodos. En trabajos anteriores hemos demostrado que el criterio con el cual se concreta esta elección puede considerar diferentes fuentes de información tales como la topología [13], las rutas entre los nodos [80] y el tráfico planificado [15]. Esto puede observarse en la ilustración de la Figura 2.5, donde a medida que se utiliza mayor información de la red se obtienen mayores rendimientos a costa también de una mayor complejidad computacional en el proceso de diseño.

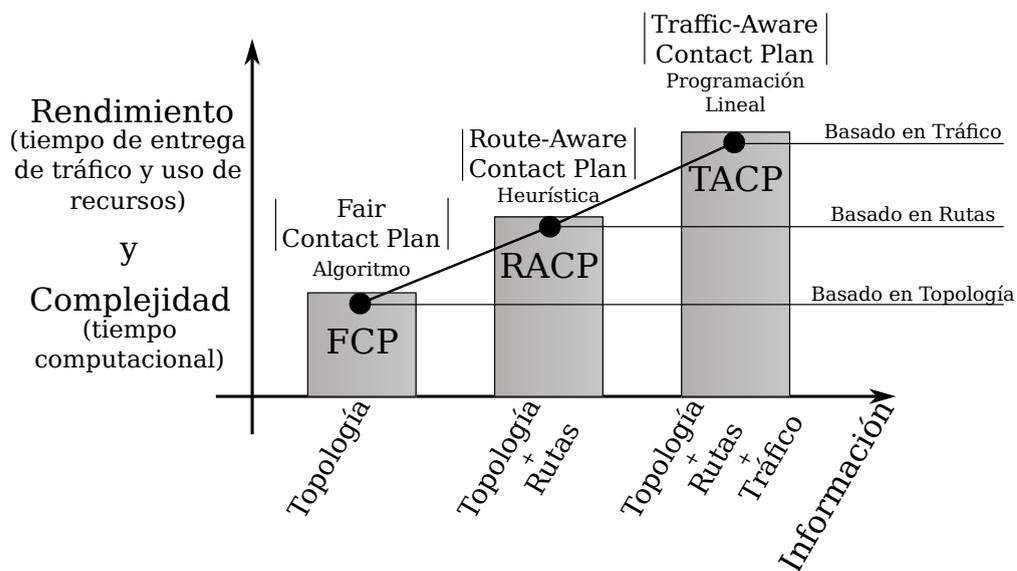


FIGURA 2.5: Rendimiento esperado de los mecanismos de diseño FCP, RACP y TACP

Para ilustrar el concepto de diseño del plan de contactos, retomamos la topología que utilizamos en la sección 2.2.5. Consideraremos que cada satélite dispone de un único transponder y dos antenas que puede seleccionar mediante un conmutador de potencia. De esta forma, cada satélite solo podrá comunicarse con un único satélite al mismo tiempo, por lo que es necesario tomar una decisión en los estados k_3 , k_4 , y k_5 .

En la Figura 2.6 a) y b) se pueden apreciar dos posibles planes de contactos. Si se elige el primer plan de contacto, la red tendrá habilitada la máxima capacidad de transmisión dado que los contactos suman el mayor tiempo de comunicación posible. Sin embargo,

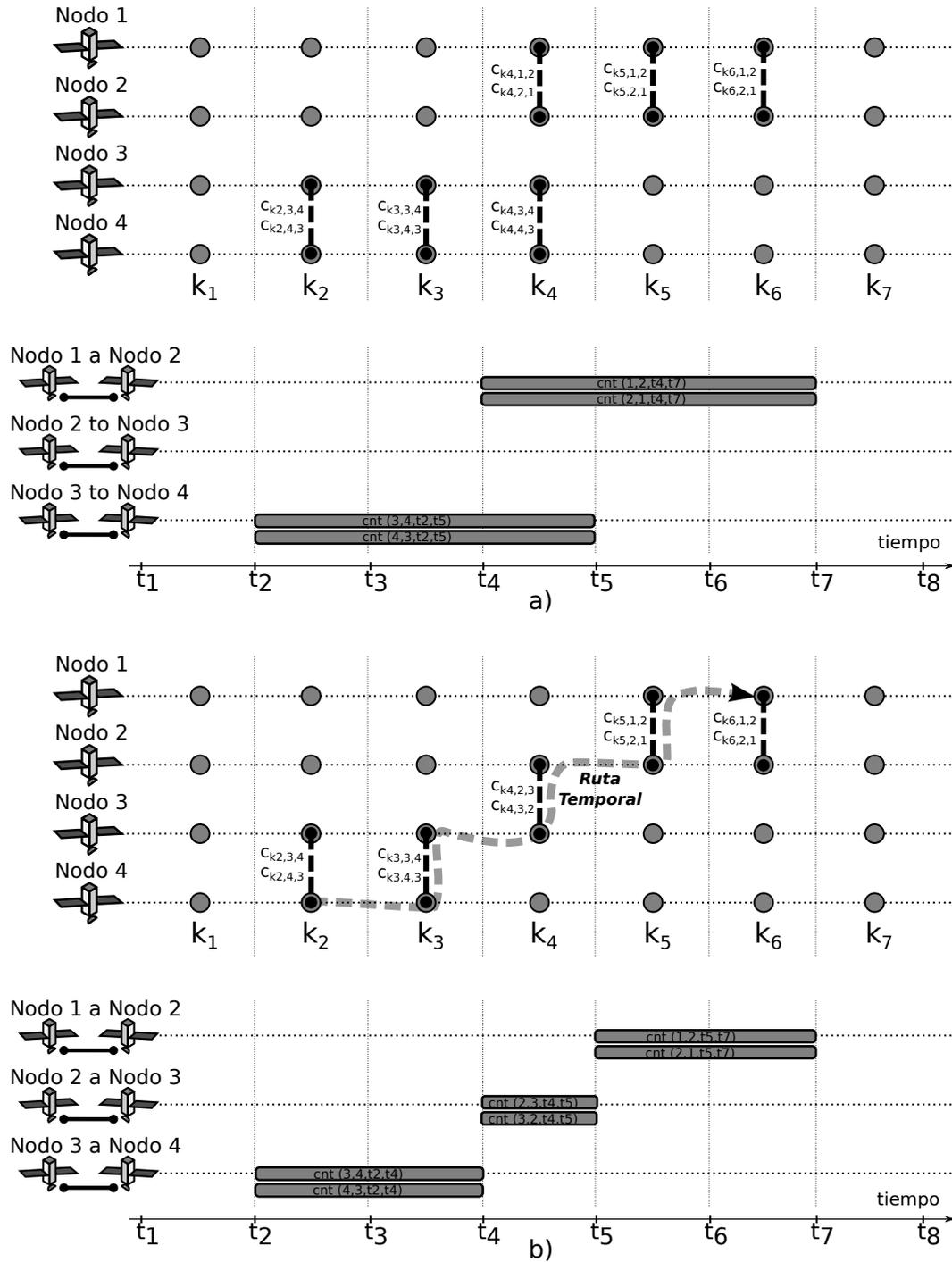


FIGURA 2.6: Dos posibles planes de contactos que priorizan a) máxima capacidad de transmisión y b) justicia de asignación de enlaces

dicha elección implica obtener una red disconexa, donde los nodos N_3 y N_4 no se podrían comunicar con los nodos N_1 y N_2 .

En este sentido, el plan de contactos mostrado en la Figura 2.6 b) favorece una distribución mas equitativa y justa de los enlaces dando lugar además a una red mas comunicada

a través de *rutas temporales*. En otras palabras, ambos planes de contactos son factibles y satisfacen las restricciones de recursos del sistema, sin embargo estos responden a criterios de selección diferentes: máxima capacidad de sistema o justicia en la asignación de los enlaces.

2.3. Rutas Temporales

En redes estáticas permanentemente conectadas las rutas o caminos entre dos nodos se pueden expresar sin ambigüedad mediante una secuencia de nodos. De esta forma, un paquete generado en un nodo origen puede llegar al nodo destino mediante los envíos sucesivos descritos por una ruta.

Por otra parte, tal como se describió en la sección 1.5.2.2, dado que las redes DTN están sujetas a interrupciones en los enlaces, y en el caso general es posible que no exista nunca un camino extremo a extremo para llegar al destino, las mismas deben utilizar el mecanismo de almacenar paquetes en los nodos, y transmitirlos cuando ocurren los contactos (*store-carry-and-forward*). Por lo tanto, para que no exista ambigüedad sobre el camino a seguir por dichos paquetes, las rutas se deben expresar teniendo en cuenta tanto los nodos como la dimensión temporal. En este contexto se define una *ruta* desde un nodo X a un nodo D como una secuencia de contactos $cnt(i, j, t_{ini}, t_{fin})$, donde:

- El nodo transmisor (i) del primer contacto debe ser X
- El nodo receptor (j) del último contacto debe ser D
- El nodo receptor (j) para el contacto n debe ser igual al nodo transmisor (i) para el contacto $n + 1$
- El tiempo de finalización (t_{fin}) del contacto $n + 1$ debe ser mayor al tiempo de comienzo (t_{ini}) del contacto n

Analizando nuevamente la Figura 2.6 b), se puede notar que existe una ruta temporal desde N_4 a N_1 dada por la secuencia [$cnt(4, 3, t_2, t_4)$, $cnt(3, 2, t_4, t_5)$, $cnt(2, 1, t_5, t_7)$]. Notar que esta ruta no es bidireccional a pesar de que existen contactos bidireccionales entre los nodos, es decir, N_1 no puede utilizar esta ruta para llegar a N_4 .

Cabe destacar además que en este tipo de rutas tiene sentido calcular atributos tales como:

- El tiempo de terminación (*termination time*), que constituye el tiempo límite hasta el cual una ruta puede ser utilizada, y que está dado por el menor tiempo de finalización entre todos los contactos de la ruta. En este caso t_4 .
- El tiempo de entrega (*best delivery time*), que constituye el tiempo mas optimista en el cual un bundle podría ser entregado a destino utilizando la ruta y que está dado por el tiempo de comienzo del último contacto. En este caso t_5 .
- La capacidad (*capacity*), que constituye el volumen de tráfico total que puede ser entregado al destino, y que en el caso en que los contactos no se superpongan temporalmente, está limitado por el contacto con menor capacidad.
- La cantidad de saltos (*hops number*), que está dada por la cantidad de nodos que existe hasta llegar al destino. Este número coincide con la cantidad de contactos de la ruta. En este caso 3 saltos/hops.
- El costo, que puede ser cualquiera de los atributos anteriores y que se utiliza por los algoritmos de enrutamiento para decidir que una ruta es mejor que otra.

2.4. Flujo de la Información

Dado que en redes DTN planificadas se puede aprovechar la información sobre el PC y sobre las rutas, en general la utilización de dicha información sigue un flujo tal y como se muestra en la Figura 2.7. En una primera instancia, una entidad central tal como una estación terrena o un MOC calcula tanto la TC como el PC a partir de propagadores orbitales, modelos de comunicación y el proceso de diseño del plan de contactos. Opcionalmente, se pueden calcular en este punto las rutas temporales. En una segunda etapa, se distribuye esta información a todos los nodos satelitales de la red ya sea utilizando comunicaciones en banda o fuera de banda. Finalmente, en una tercera etapa, los nodos utilizan esta información para *i*) calcular las rutas en caso de que las mismas no hayan sido computadas previamente o que las rutas calculadas ya no sean válidas y *ii*) realizar el envío del tráfico de acuerdo a dichas rutas.

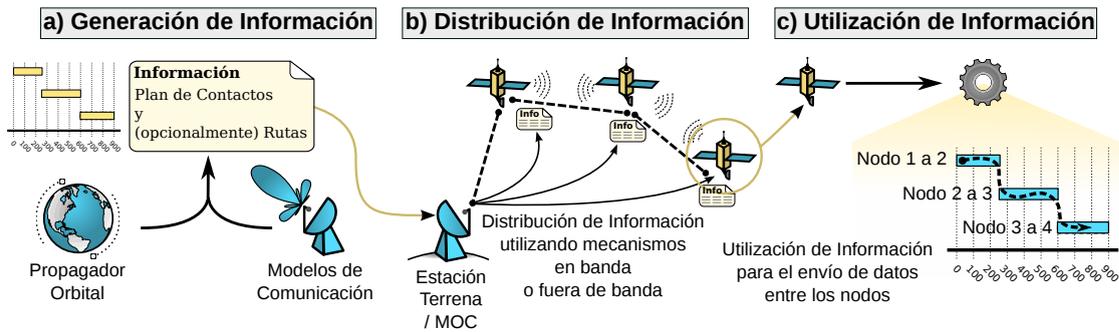


FIGURA 2.7: Flujo de información en redes DTN planificadas

2.5. Mecanismos de Enrutamiento

El enrutamiento de datos es el proceso en el cual se calcula y utiliza la información de rutas temporales para realizar el envío de datos. Este puede a su vez dividirse en tres etapas:

- Enrutamiento (encaminamiento o ruteo) propiamente dicho. En este proceso se toma como entrada el PC y se calcula una tabla de rutas.
- Encolamiento. En este proceso se selecciona una ruta de la tabla y se coloca el bundle en una cola correspondiente al nodo receptor del primer contacto de dicha ruta.
- Envío. Una vez que comienza un contacto, el nodo transmisor realiza el envío de los bundles encolados para el nodo receptor de dicho contacto.

Existen diferentes enfoques o esquemas para realizar el enrutamiento de datos en redes DTN planificadas:

Como propuso Merugu en [42], una alternativa es centralizar el cálculo de las rutas en la misma entidad que calcula el PC. Luego de que esta información de rutas es enviada junto al PC a los nodos de la red, estos solo deben efectuar el encolamiento y envío de bundles. La principal ventaja de este esquema es que los nodos satelitales no necesitan realizar el cómputo de rutas en sus limitadas computadoras de a bordo. La desventaja radica en que para hacer frente a posibles fallas en la planificación de la topología y/o el tráfico, se deben calcular rutas adicionales que probablemente no se terminen utilizando.

Por otra parte, existe otro enfoque que se basa en un cálculo de rutas distribuido, donde los nodos utilizan el PC recibido de antemano para calcular las rutas requeridas a medida que las necesitan. Si bien de esta forma los nodos necesitan tener una capacidad de cómputo superior, esto posibilita un esquema más robusto en cuanto a tolerancia a fallas y una mayor utilización de la información calculada. El enfoque distribuido es el más utilizado actualmente en redes DTN planificadas y es el que ha recibido la mayor cantidad de aportes de la comunidad. Por lo tanto, en esta tesis nos centramos en algoritmos de enrutamiento tales como *Contact Graph Routing (CGR)* [43, 81] que han sido especialmente diseñados para ser aplicados de forma distribuida.

2.5.1. Contact Graph Routing (CGR)

2.5.1.1. Mejoras Realizadas

CGR se documentó inicialmente en 2009 y luego se actualizó en 2010 como un borrador experimental de la IETF [43]. A fines de ese mismo año, Segui et al. [47] propusieron el menor tiempo de entrega (*best delivery time*) como una métrica conveniente de optimización que al ser monótonamente decreciente evita los bucles de enrutamiento y permite un uso adaptado del algoritmo Dijkstra para el cálculo de rutas [82]. Esta mejora se introdujo luego en la versión oficial de CGR y se incluyó en la implementación ION desarrollada por el *Jet Propulsion Laboratory* [56, 83]. En 2012, Birrane et al. [84] propusieron una extensión a CGR llamada *Extención Block CGR (EB-CGR)* con el objetivo de reducir los cálculos en nodos intermedios a expensas de una sobrecarga en el encabezado del paquete. La idea de EB-CGR consiste en que solo el nodo generador de un bundle sea el encargado de calcular la ruta. Luego, la misma es agregada en la cabecera del bundle y es extraída en nodos intermedios. Más tarde, en 2014, los autores estudiaron la implementación de listas de rutas como medio para minimizar las ejecuciones de CGR [12]. En el mismo año, Bezirgiannidis et al. [45] sugirieron que se supervisen las colas de transmisión dentro de CGR para calcular el tiempo de entrega del bundle de forma más precisa (*CGR-ETO*). Propusieron además un mecanismo llamado *overbooking management* que permite el re-enrutamiento proactivo de bundles previamente encolados en situaciones en que el encolamiento de nuevos bundles con mayor prioridad cambia las condiciones asumidas para los primeros. Con respecto al manejo de la congestión se propusieron extensiones adicionales en los trabajos [85–87]. Más recientemente, en 2016,

Burleigh et al. [88] buscaron ampliar la aplicabilidad de CGR para abarcar también redes terrestres oportunistas. Actualmente, CGR se está estandarizando formalmente en un documento de la CCSDS llamado *Schedule-Aware Bundle Routing (SABR)* [89]. Además, todas estas modificaciones se han implementado en el software ION. Como resultado, ION se convierte en una referencia importante para los últimos mecanismos de enrutamiento y envío para DTNs espaciales.

2.5.1.2. Descripción del Algoritmo

A continuación proporcionamos una descripción sobre el algoritmo CGR que es extraída del trabajo [7] publicado por el grupo de investigación.

Algoritmo 1: CGR Forward

input : bundle to forward B , contact plan Cp , route list Rl , excluded nodes En

output: bundle B is enqueued in the corresponding queue

```

1 if  $Cp$  changed since last  $Rl$  calculation then
2   |  $Rl \leftarrow \emptyset$ ;
3 if  $B$  forbids return to sender then
4   |  $En \leftarrow B$  sender node;
5  $Pn \leftarrow \text{identifyProxNodes}(B, Cp, Rl, En)$ ;
6 if  $B$  is critical then
7   | enqueue a copy of  $B$  to each node in  $Pn$ ;
8   | return
9 set  $nextHop$  to empty;
10 for  $pn \in Pn$  do
11   | if  $nextHop$  is empty then
12     |  $nextHop = pn$ 
13   | else if  $pn.arrivalTime < nextHop.arrivalTime$  then
14     |  $nextHop = pn$ 
15   | else if  $pn.arrivalTime > nextHop.arrivalTime$  then
16     | continue
17   | else if  $pn.hops < nextHop.hops$  then
18     |  $nextHop = pn$ 
19   | else if  $pn.hops > nextHop.hops$  then
20     | continue
21   | else if  $pn.id < nextHop.id$  then
22     |  $nextHop = pn$ 
23 if  $nextHop$  is not empty then
24   | enqueue  $B$  to  $nextHop$ ;
25   |  $\text{manageOverbook}(B, nextHop)$ 
26 else
27   | enqueue  $B$  to limbo
28 return

```

La rutina *CGR Forward* descrita en el Algoritmo 1 se invoca cada vez que un nodo necesita enviar un bundle. Recibe como entrada el bundle B , el plan de contactos Cp , una lista de rutas Rl para un destino dado $B.dst$ y una lista de nodos excluidos En . Inicialmente, en las líneas 1 y 2 se comprueba si el Cp se modificó desde el último cálculo de Rl . En caso afirmativo, Rl se borra para forzar una actualización de rutas para el destino $B.dst$. Como se explicó anteriormente, las rutas en DTN se expresan en función del tiempo y solo son válidas por un período determinado. Por lo tanto, su validez debe ser corroborada por CGR para cada bundle.

A continuación, el procedimiento *identifyProximateNodes* descrito en el Algoritmo 2 obtiene una lista de nodos vecinos Pn con todos los nodos posibles que, de acuerdo con la lista de rutas Rl , tienen asociada una ruta válida hacia el destino $B.dst$ (línea 5). En este paso, se utiliza como entrada una lista de nodos excluidos En para evitar la consideración de nodos prohibidos administrativamente (por ejemplo, nodos que no responden) o para evitar bucles que podrían ocurrir si se envía el bundle al mismo nodo desde el cual proviene (líneas 3 y 4). Una vez obtenida la lista Pn , esta se puede utilizar para enviar el bundle a uno o mas nodos vecinos de acuerdo al tipo de bundle. Si el bundle es crítico, este es clonado y encolado a todos los nodos de Pn (líneas 6 a 8). En este caso, concluye aquí el procedimiento de enrutamiento y encolamiento. Por otro lado, si el bundle es de tipo normal, se debe elegir un único nodo *nextHop* donde el bundle será encolado (líneas 10 a 22). Este nodo se escoge de acuerdo a las rutas asociadas con cada nodo vecino de Pn . En primer lugar se busca el nodo cuya ruta tiene el menor tiempo de entrega al destino. En caso de empate, se escoge el nodo cuya ruta tiene el menor número de saltos. Finalmente, si el empate continúa se da prioridad al nodo con identificador mas pequeño.

Luego, si se encuentra un *nextHop*, el bundle B se inserta en la cola de salida correspondiente (línea 24) antes de ejecutar el procedimiento *manageOverbook* (línea 25). En caso contrario, el bundle se almacena en un espacio de memoria especial llamado *limbo* (línea 27), donde un proceso de nivel superior puede borrarlo o intentar un encolamiento posterior (por ejemplo, después de una actualización del Cp). Como resultado, después de completar esta rutina de CGR, uno o más bundles pueden almacenarse en la memoria local esperando el contacto con el nodo vecino correspondiente.

Algoritmo 2: Identify Proximate Node List**input** : bundle to forward B , contact plan Cp , route list Rl , excluded nodes En ,**output**: proximate nodes list Pn

```

1 if  $Rl$  is empty then
2   |  $Rl \leftarrow \text{loadRouteList}(B, Cp)$ ;
3    $Pn \leftarrow \emptyset$ ;
4 for  $route \in Rl$  do
5   | if  $route.toTime \leq currentTime$  then
6     | continue (ignore past route)
7   | if  $route.arrivalTime \geq B.deadline$  then
8     | continue (route arrives late)
9   | if  $route.capacity < B.bitLength$  then
10    | continue (not enough capacity)
11   | if  $route.nextHop \in En$  then
12    | continue (next hop is excluded)
13   | if  $localQueue(route.firstContact) < B.bitLength$  then
14    | continue (bundle does not fit in the route first contact)
15   | for  $pn \in Pn$  do
16     | if  $pn = route.nextHop$  then
17       | if  $pn.arrTime > route.arrTime$  then
18         | replace  $pn$  with  $route.nextHop$ 
19       | else if  $pn.arrTime < route.arrTime$  then
20         | continue (previous route was better)
21       | else if  $pn.hops > route.hops$  then
22         | replace  $pn$  with  $route.nextHop$ 
23       | else if  $pn.hops < route.hops$  then
24         | continue (previous route was better)
25       | break
26   | if  $route.nextHop \notin Pn$  then
27     |  $pn \leftarrow route.nextHop$ ;
28     |  $Pn \leftarrow pn$ ;
29 return  $Pn$ 

```

Por otra parte, como mencionamos anteriormente, la rutina *identifyProximateNodes* representada en el Algoritmo 2 explora las rutas existentes para derivar la lista Pn . Esta lista está formada por un conjunto de estructuras con nodos vecinos (y atributos adicionales asociados a la mejor ruta por cada vecino) que no se repiten y que pueden eventualmente alcanzar el destino del bundle. Si la lista de rutas Rl está vacía se invoca la función *Load Route List* para encontrar todas las rutas hacia el destino de B (líneas 1 y 2). Luego, cada una de estas rutas debe ser evaluada para llenar Pn . Inicialmente, aquellas rutas que no satisfacen ciertas condiciones son descartadas (líneas 4 a 14). En particular, rutas con un tiempo de terminación (*toTime*) en el pasado, un tiempo de entrega posterior al tiempo límite de vida del bundle, una capacidad menor que el

tamaño del bundle, un nodo vecino dentro de los nodos excluidos o que tienen un contacto inicial sin capacidad residual para acomodar el bundle, son ignoradas. Las rutas restantes se consideran válidas y el nodo vecino correspondiente en P_n se sustituye por el de una ruta mejor (líneas 15 a 24) o se agrega directamente a la lista (líneas 26 y 27). El criterio de reemplazo es coherente con el de selección del Algoritmo 1. En primer lugar se considera el menor tiempo de entrega y luego el número de saltos. Durante este proceso, las métricas de ruta como el tiempo de entrega y el número de saltos también se almacenan en cada estructura de datos de nodo vecino contenida en P_n .

En general, para buscar todas las rutas posibles desde una fuente a un destino, CGR utiliza una representación del PC llamada *grafo de contactos*. Se trata de un grafo conceptual acíclico y dirigido cuyos vértices corresponden a contactos mientras que los arcos representan episodios de retención de datos (es decir, almacenamiento) en un nodo [89]. Además, se agregan dos vértices ficticios: el vértice raíz, que es un contacto del nodo fuente a si mismo, y un vértice terminal, que es un contacto del destino a si mismo. Aunque la estructura del grafo de contactos puede parecer contraintuitiva, es una representación estática conveniente de una topología evolutiva en el tiempo y que puede utilizarse para ejecutar algoritmos de grafos tradicionales tales como las búsquedas de Dijkstra. Las distintas representaciones de un CP de ejemplo se pueden observar en forma temporal en la Figura 2.8, en forma de tabla en 2.1, y en forma de grafo de contactos desde un nodo fuente A hacia un nodo destino D en la Figura 2.9.

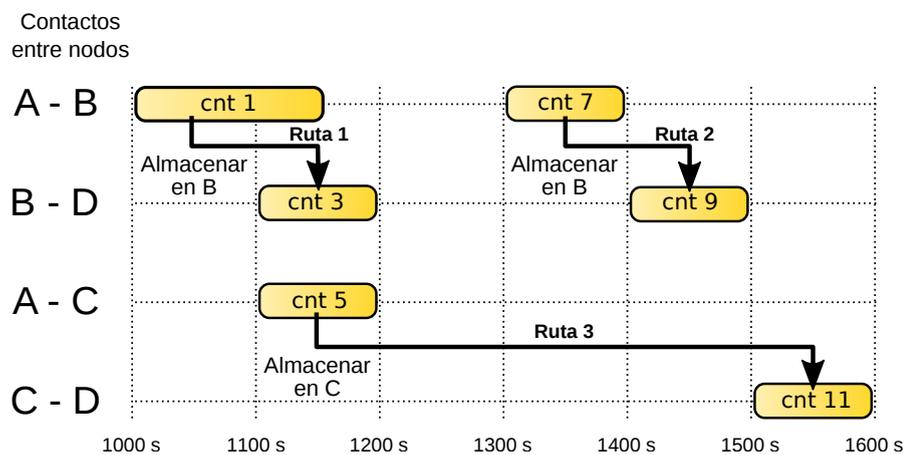


FIGURA 2.8: Ejemplo de plan de contactos en representación temporal con rutas desde un nodo A hacia un nodo D

contacto #	Tx	Rx	ini	fin	tasa
1	A	B	1000	1150	1000
2	B	A	1000	1150	1000
3	B	D	1100	1200	1000
4	D	B	1100	1200	1000
5	A	C	1100	1200	1000
6	C	A	1100	1200	1000
7	A	B	1300	1400	1000
8	B	A	1300	1400	1000
9	B	D	1400	1500	1000
10	D	B	1400	1500	1000
11	C	D	1500	1600	1000
12	D	C	1500	1600	1000

TABLA 2.1: Ejemplo de plan de contactos en forma de tabla

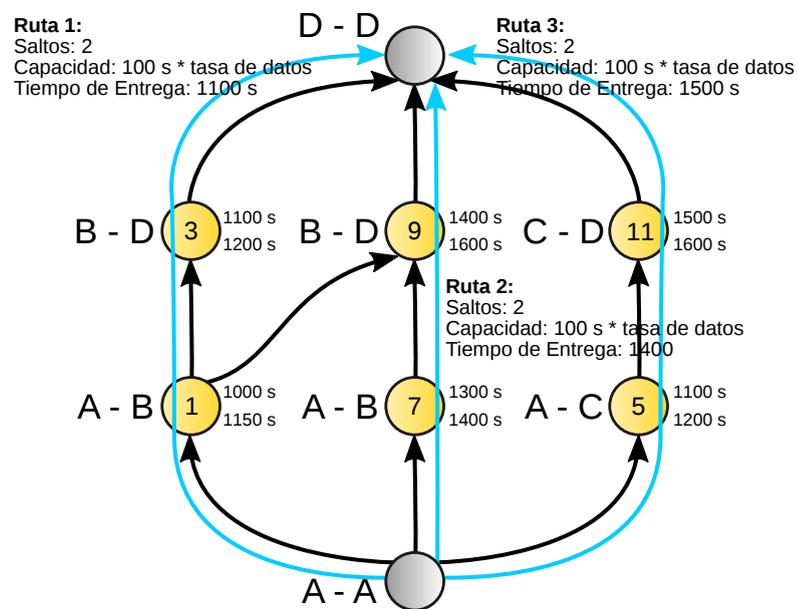


FIGURA 2.9: Ejemplo de plan de contactos en representación como grafo de contactos desde un nodo A hacia un nodo D

Para encontrar todas las rutas posibles hacia un destino, la rutina *Load Route List* descrita en el Algoritmo 3 realiza una serie de búsquedas de Dijkstra a través del grafo de contactos.

Algoritmo 3: Load Route List

```

input : bundle to forward  $B$ , contact plan  $Cp$ 
output: route list  $Rl$ 
1  $Rl[B.destination] \leftarrow \emptyset$ ;
2 for  $contact \in Cp$  do
3    $contact.work.arrivalTime \leftarrow \text{inf}$ ;
4    $contact.work.capacity \leftarrow 0$ ;
5    $contact.work.predecessor \leftarrow 0$ ;
6    $contact.work.visited \leftarrow false$ ;
7    $contact.work.suppressed \leftarrow false$ ;
8 set  $anchorContact$  to empty;
9 while 1 do
10   $route \leftarrow \text{Dijkstra}(B.dst, Cp, rootContact)$ ;
11  if  $route$  is empty then
12    break (no more routes in contact graph)
13   $firstContact \leftarrow route.firstHop$ ;
14  if  $anchorContact$  not empty then
15    if  $anchorContact \neq firstContact$  then
16      for  $contact \in Cp$  do
17         $contact.work.arrivalTime \leftarrow \text{inf}$ ;
18         $contact.work.predecessor \leftarrow 0$ ;
19         $contact.work.visited \leftarrow false$ ;
20        if  $contact.source \neq local\ node$  then
21           $contact.work.suppressed \leftarrow false$ ;
22         $anchorContact.work.suppressed \leftarrow true$ ;
23        set  $anchorContact$  to empty;
24        continue (go to next Dijkstra's search)
25   $Rl[B.destination] \leftarrow route$ ;
26  if  $route.toTime = firstContact.endTime$  then
27     $limitContact \leftarrow firstContact$ ;
28  else
29     $anchorContact \leftarrow firstContact$ ;
30    for  $contact \in route.hops$  do
31      if  $contact.toTime = route.toTime$  then
32         $limitContact \leftarrow contact$ ;
33        break (limit contact found)
34   $limitContact.work.suppressed \leftarrow true$ ;
35  for  $contact \in Cp$  do
36     $contact.work.arrivalTime \leftarrow \text{inf}$ ;
37     $contact.work.predecessor \leftarrow 0$ ;
38     $contact.work.visited \leftarrow false$ ;
39 return

```

Un área de trabajo *work* es reservada dentro de cada contacto en el Cp para ejecutar las búsquedas de Dijkstra. Inicialmente, el algoritmo borra todos los parámetros requeridos en cada área de trabajo de todos los contactos (líneas 2 a 7). Luego, la rutina se repite para encontrar diferentes rutas usando el algoritmo de Dijkstra (líneas 9 y 10). La métrica

que impulsa la búsqueda de ruta más corta es el tiempo de entrega (*arrivalTime*), que debe calcularse a partir del inicio y el retraso esperado de cada contacto en la ruta explorada. Para garantizar que cada ejecución de Dijkstra proporcione una ruta distinta, se elimina el *contacto limitante* (*limitContact*) de cada ruta encontrada. El contacto limitante de una ruta se define como el contacto que tiene su final más temprano en el tiempo [89]. En general, el contacto limitante a menudo es el primer contacto de la ruta (por ejemplo, observar los contactos 1, 7 y 5 en la Figura 2.8). Por lo tanto, la eliminación del contacto limitante permite proporcionar la siguiente mejor ruta hacia el destino. Sin embargo, en algunos casos especiales, el nodo transmisor está detrás de un contacto muy largo (por ejemplo, un contacto de Internet). Para estos casos, un mecanismo llamado *anchoring* permite que el algoritmo encuentre varias rutas a través de contactos largos. Este mecanismo entra en juego cuando el algoritmo detecta que el contacto limitante no corresponde al primer contacto. En esta etapa, se almacena un *anchorContact* que permite encontrar y suprimir el contacto limitante (línea 29 a 34). Después de limpiar las áreas de trabajo (línea 35 a 38), se ejecuta una nueva búsqueda, ahora con el contacto limitante eliminado del grafo de contactos. Tan pronto como se encuentra la primera ruta sin el *anchorContact* como primer contacto, la búsqueda anclada termina suprimiendo el *anchorContact* y la búsqueda normal continúa (línea 15 a 24). La búsqueda finaliza cuando no se pueden encontrar más rutas en el grafo de contactos.

Aunque es un algoritmo complejo y extenso, CGR se considera una de las estrategias más maduras para el enrutamiento en DTNs espaciales. Por lo tanto, como veremos en capítulos posteriores de esta tesis, se ha hecho especial hincapié en mejoras que pueden integrarse con la actual implementación de CGR.

Capítulo 3

Congestión

3.1. Introducción

En general, la *congestión* se ha definido como el fenómeno producido cuando a la red, o a parte de ella, se le ofrece mas tráfico del que puede cursar. De esta forma, la congestión puede ser provocada por una combinación de restricciones topológicas (anchos de banda, capacidades de almacenamiento, etc) y/o tráfico excesivo. A diferencia de las redes tradicionales basadas en Internet, en DTN no se pueden utilizar respuestas estables extremo a extremo para implementar el control de la congestión. Por lo tanto, de la misma forma que se debieron diseñar esquemas de enrutamiento adaptados a este tipo de redes, se deben diseñar también mecanismos diferentes para evitar la congestión.

En este capítulo repasamos el funcionamiento del control de congestión en Internet y en redes DTN, analizamos distintos tipos de congestión, y proponemos mecanismos novedosos que resultan particularmente aplicables en DTNs planificadas.

3.1.1. Congestión en Internet

Resulta útil destacar en primer lugar la diferencia entre control de flujo y control de congestión. Si bien ambos se implementan mediante una disminución de la tasa de datos del transmisor, existe una diferencia entre estos conceptos. Mientras que en el control de flujo se intenta que el transmisor no envíe más datos que los que *el receptor* puede manejar, en el control de congestión se intenta que el transmisor no envíe más datos que los que *la red* es capaz de manejar. Por lo tanto, la congestión ocurre cuando las demandas de tráfico exceden la capacidad de red disponible (incluyendo todos los enlaces y enrutadores desde la fuente al destino).

Los algoritmos para el control de la congestión en Internet se implementan en la capa de transporte TCP y están publicados en el RFC 5681 de la IETF [90]. La idea básica consiste en detectar la congestión utilizando la pérdida de paquetes como indicador para luego reducir la tasa de transmisión de la fuente. Por cada conexión, TCP mantiene una *ventana de congestión*, que limita el número total de segmentos de datos de los cuales no se ha recibido acuse de recibo (ACK) y que pueden estar en tránsito de extremo a extremo. TCP usa un mecanismo llamado *inicio lento* para aumentar el tamaño de la ventana de congestión de forma exponencial hasta un determinado umbral, luego del cual se ingresa en una etapa llamada *evitación de la congestión* en la que el crecimiento de la ventana pasa a ser lineal. En caso de pérdida de segmentos se reduce la ventana de forma multiplicativa y se vuelve a la etapa de comienzo lento o a la etapa de evitación de la congestión dependiendo de la implementación particular.

3.1.2. Congestión en DTN

Por otra parte, en DTN no se tiene una realimentación estable extremo a extremo debido a la naturaleza esporádica de los contactos. Además, la pérdida de paquetes puede ser grande debido a las condiciones físicas en que funcionan estas redes y, por otro lado, resulta esperable que los buffers de los nodos mantengan grandes volúmenes de datos durante largos intervalos de tiempo. Por lo tanto, estas métricas no resultan indicadores directos de congestión, lo que dificulta aún más tanto su predicción como evitación.

Silva et al. propusieron una taxonomía para clasificar los mecanismos de control de congestión en redes DTN [91]. Por un lado, existen mecanismos *reactivos* salto a salto

tales como la transferencia de custodia donde se aceptan o rechazan envíos mediante un monitoreo de la ocupación del buffer y esquemas que generan paquetes específicos una vez que se detecta la congestión. Estos resultan en general insuficientes en redes altamente disconexas debido a que la realimentación demora demasiado tiempo en llegar a los nodos que pueden tomar las acciones necesarias para hacer frente a la situación. Por otro lado, se han propuesto también mecanismos *proactivos* basados en considerar la capacidad de los contactos planificados. Sin embargo, estos mecanismos solo resuelven el problema parcialmente, ya que solo consideran las capacidades locales a un nodo. En el algoritmo CGR, por ejemplo, mientras se realiza el encolamiento de bundles, se va actualizando y manteniendo la *capacidad residual* de los primeros contactos de cada ruta escogida. En general, se ha asumido que no hay beneficio en considerar la capacidad del resto de los contactos de la ruta ya que la decisión no resulta determinista en los siguientes nodos. A continuación analizamos y ponemos en tela de juicio estas suposiciones.

3.2. Congestión con un Único Tráfico

La Figura 3.1 describe un ejemplo de una topología con 3 contactos entre 3 nodos y donde $N1$ genera un único tráfico de 10 bundles de datos en el estado $k2$ destinado a $N3$. Existen 2 rutas hacia el destino: R1 compuesta por $C_{k2,1,2}$ y $C_{k3,2,3}$, y R2 compuesta por el contacto directo $C_{k4,1,3}$.

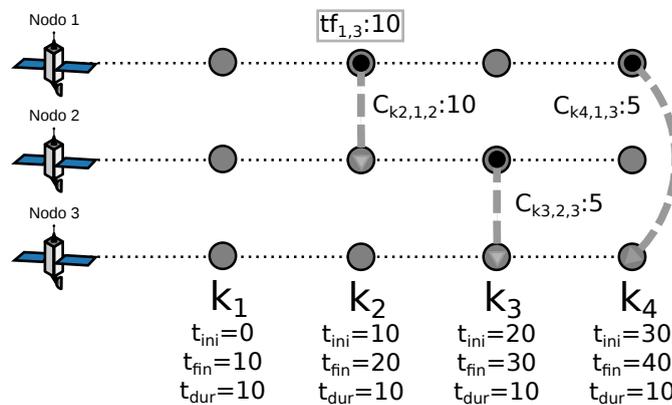


FIGURA 3.1: Plan de contactos de caso de ejemplo con un flujo de tráfico

Debido a que $N3$ tiene un transponder con menor capacidad, los contactos con $N3$ tienen asignado una capacidad de 5 bundles de datos, mientras que el contacto entre $N1$ y $N2$ tiene una capacidad mayor de 10 bundles de datos.

En la Figura 3.2 se observan dos posibles flujos de estos datos. En 3.2 a), $N1$ envía 10 bundles a $N2$, de los cuales solo 5 podrán ser entregados finalmente a $N3$ debido a la limitada capacidad del contacto $C_{k3,2,3}$. Por otro lado, en 3.2 b), se puede aprovechar tanto $R1$ como $R2$ para realizar una entrega del tráfico sin provocar congestión. En este caso solo hemos considerado la congestión provocada por la limitada capacidad de los contactos. Sin embargo, la congestión también puede ocurrir debido a limitaciones en los buffers de los nodos. Por ejemplo, si $N2$ tuviera un buffer que solo pudiera almacenar hasta 5 bundles, se vería un efecto de congestión similar o incluso peor, debido a que se deberían descartar 5 bundles. Como mencionamos anteriormente, el algoritmo CGR solo mantiene el estado de la capacidad residual del primer contacto de cada ruta. Por lo tanto, en este ejemplo se comportaría como en el caso que genera congestión. Con el objetivo de hacer frente a este tipo de situaciones proponemos un esquema llamado *Local Path Aware CGR (LPA-CGR)* que describimos a continuación.

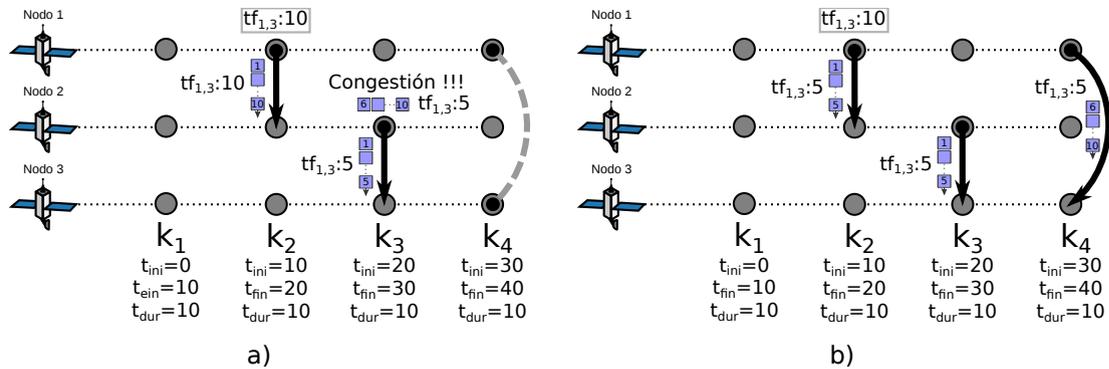


FIGURA 3.2: Flujo de un único tráfico: a) con congestión, b) sin congestión

3.3. Local Path Aware CGR (LPA-CGR)

3.3.1. Concepto

Mientras que CGR solo mantiene un estado actualizado de la capacidad residual de los contactos locales (primer contacto de cada ruta), *Local Path Aware CGR (LPA-CGR)*

fue diseñado para considerar la capacidad residual de la ruta completa. En caso de que los contactos no se superpongan temporalmente, la capacidad residual de la ruta se encontrará limitada por el contacto con menos capacidad residual o el nodo con menos espacio disponible en su buffer a lo largo de la ruta. Como resultado, un nodo vecino solo será considerado factible por LPA-CGR si y solo si la capacidad residual de la ruta asociada puede acomodar el tamaño del bundle a enviar.

Algoritmo 4: LPA-CGR Forward

input : bundle to forward B , contact plan Cp , route list Rl , excluded nodes En
output: bundle B is enqueued in the corresponding queue
global : $BufferCap$

- 1 **if** Cp changed since last Rl calculation **then**
- 2 | $Rl \leftarrow \emptyset$;
- 3 **if** B forbids return to sender **then**
- 4 | $En \leftarrow B$ sender node;
- 5 $Pn \leftarrow \text{identifyProxNodes}(B, Cp, Rl, En)$;
- 6 **if** B is critical **then**
- 7 | enqueue a copy of B to each node in Pn ;
- 8 | updateBuffersResidualCapacity;
- 9 | decreaseContactsResidualCapacity;
- 10 | decreaseRoutesResidualCapacity;
- 11 | **return**
- 12 set $nextHop$ to empty;
- 13 **for** $pn \in Pn$ **do**
- 14 | **if** $nextHop$ is empty **then**
- 15 | | $nextHop = pn$
- 16 | **else if** $pn.arrivalTime < nextHop.arrivalTime$ **then**
- 17 | | $nextHop = pn$
- 18 | **else if** $pn.arrivalTime > nextHop.arrivalTime$ **then**
- 19 | | **continue**
- 20 | **else if** $pn.hops < nextHop.hops$ **then**
- 21 | | $nextHop = pn$
- 22 | **else if** $pn.hops > nextHop.hops$ **then**
- 23 | | **continue**
- 24 | **else if** $pn.id < nextHop.id$ **then**
- 25 | | $nextHop = pn$
- 26 **if** $nextHop$ is not empty **then**
- 27 | enqueue B to $nextHop$;
- 28 | updateBuffersResidualCapacity;
- 29 | decreaseContactsResidualCapacity;
- 30 | decreaseRoutesResidualCapacity;
- 31 | manageOverbook($B, nextHop$)
- 32 **else**
- 33 | enqueue B to limbo
- 34 **return**

3.3.2. Descripción Algorítmica

Para cumplir con el comportamiento requerido, resulta necesario agregar las variables *residualCapacity* tanto en los contactos como en las rutas. Además, se debe crear y actualizar una estructura global al nodo llamada *BufferCap* que representará el estado del buffer de cada nodo de la red en función del tiempo. Esta estructura debe inicializarse con los tamaños de los buffers de cada nodo, los cuales deben ser brindados como información adicional en el PC. El Algoritmo 1 de CGR debe modificarse como se muestra en el Algoritmo 4 mediante el agregado de las líneas 8-10 y 28-30 que realizan las acciones descritas cada vez que la rutina encola un bundle a un vecino.

Algoritmo 5: LPA-CGR Identify Proximate Node List

input : bundle to forward B , contact plan Cp , route list Rl , excluded nodes En ,

output: proximate nodes list Pn

```

1 if  $Rl$  is empty then
2   |  $Rl \leftarrow \text{loadRouteList}(B, Cp)$ ;
3    $Pn \leftarrow \emptyset$ ;
4 for  $route \in Rl$  do
5   | if  $route.toTime \leq currentTime$  then
6   |   | continue (ignore past route)
7   | if  $route.arrivalTime \geq B.deadline$  then
8   |   | continue (route arrives late)
9   | if  $route.residualCapacity < B.bitLenght$  then
10  |   | continue (not enough capacity)
11  | if  $route.nextHop \in En$  then
12  |   | continue (next hop is excluded)
13  | for  $pn \in Pn$  do
14  |   | if  $pn = route.nextHop$  then
15  |     | if  $pn.arrTime > route.arrTime$  then
16  |       | replace  $pn$  with  $route.nextHop$ 
17  |     | else if  $pn.arrTime < route.arrTime$  then
18  |       |   | continue (previous route was better)
19  |     | else if  $pn.hops > route.hops$  then
20  |       |   | replace  $pn$  with  $route.nextHop$ 
21  |     | else if  $pn.hops < route.hops$  then
22  |       |   | continue (previous route was better)
23  |     |   | break
24  |   | if  $route.nextHop \notin Pn$  then
25  |     |  $pn \leftarrow route.nextHop$ ;
26  |     |  $Pn \leftarrow pn$ ;
27 return  $Pn$ 

```

Por otra parte, el Algoritmo 5 es una versión modificada del Algoritmo 2 de CGR donde se han removido las líneas 13 y 14 (que consideraban solo la capacidad del primer

contacto) y se han modificado las líneas 9 y 10 de forma tal de considerar la capacidad residual y actualizada de la ruta completa. Notar que, anteriormente, la capacidad de la ruta era un valor estático que únicamente evitaba que un bundle no fuera mayor en tamaño a ninguno de los contactos de la ruta. En cambio, la capacidad residual es una variable que es modificada por la rutina 4 luego de cada encolamiento del tráfico local.

3.3.3. Consideraciones sobre Envío al Nodo Previo

A pesar de que LPA-CGR proporciona un esquema libre de congestión con respecto al tráfico local, no considera el tráfico que pueden llegar a generar otros nodos de la red. Por lo tanto, pueden existir situaciones en que se genere congestión, donde los nodos intermedios de una ruta requieran tomar nuevas decisiones para entregar los bundles a destino (*re-ruteo* o *re-enrutamiento*). Sin embargo, en muchos casos, esto implica devolver el bundle al nodo del cual proviene, lo cual está prohibido en la especificación de CGR con el fin de evitar bucles de enrutamiento. Como resultado, esta política puede hacer que los nodos fallen al reaccionar y recuperarse de la congestión.

Un caso mostrando este problema se ilustra en la topología con 3 nodos y 4 estados de la Figura 3.3. Aquí, $N1$ genera un tráfico de 50 bundles en $k1$ destinados a $N3$ mientras que $N2$ genera un tráfico de 100 bundles también en $k1$ y destinados a $N3$. Las capacidades $C_{k1,1,2}$, $C_{k2,2,3}$, $C_{k3,2,1}$, $C_{k4,1,3}$ son todas iguales a 100 bundles y representamos, dentro de rectángulos debajo de cada nodo, su *visión local* en términos de capacidad residual de determinados contactos. En este caso, si tanto $N1$ como $N2$ aplican CGR, $N1$ intentará que sus 50 bundles sigan la ruta conformada por $C_{k1,1,2}$ y $C_{k2,2,3}$, mientras que $N2$ intentará que sus 100 bundles utilicen la ruta formada por el contacto directo

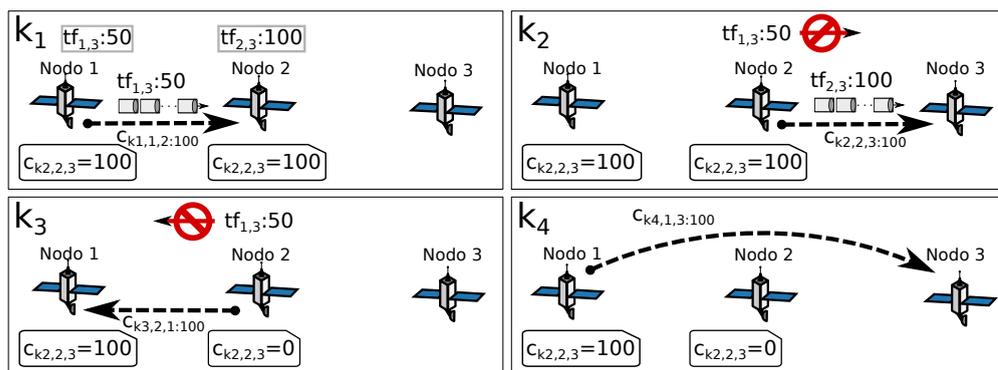


FIGURA 3.3: Reacción a la congestión mediante re-ruteo hacia el nodo previo

$C_{k2,2,3}$. El problema de congestión se detectará en $k2$ debido a que $N2$ agotará la capacidad residual de $C_{k2,2,3}$ con su propio tráfico sin dejar lugar para el envío del tráfico de $N1$. Ahora $N2$ debe re-rutear el tráfico de $N1$ y podría hacerlo a través de los contactos $C_{k3,2,1}$, $C_{k4,1,3}$. Sin embargo, la política de no retorno al nodo previo le impide aprovechar dichos contactos. De esta manera, si no existen rutas adicionales desde $N2$ hacia $N3$ (a través de otros nodos vecinos), $N2$ no tiene mas remedio que eliminar los bundles o dejarlos en un espacio especial del buffer llamado *limbo*. Por lo tanto, para evitar este problema, consideraremos en general que *LPA-CGR* tiene habilitado el retorno a otros nodos y le sumaremos la terminación *FB* o *Forward Back*.

En este caso, la congestión se produjo debido a que $N1$ no tenia manera de saber que $N2$ también generaría tráfico y utilizaría los recursos que $N1$ pretendía utilizar. Es decir, La visión que tienen los nodos con respecto a las capacidades residuales de contactos y rutas es local. Por otra parte, una visión global del tráfico que genera cada nodo implicaría una gran cantidad de mensajes de realimentación entre los nodos, hecho que en una red DTN disconexa resulta imposible o altamente ineficiente. Sin embargo, como veremos en las siguientes secciones, existen determinadas situaciones en las que es posible que cada nodo tenga noción del tráfico global sin la necesidad de dichas realimentaciones.

3.4. Congestión con Múltiples Tráficos

La Figura 3.4 describe un ejemplo similar al de la Figura 3.1 con la diferencia de que en este caso los 3 contactos tienen capacidad para enviar 10 bundles, y donde existe además un tráfico adicional de 10 bundles generado en $k1$ desde $N2$ hacia $N3$.

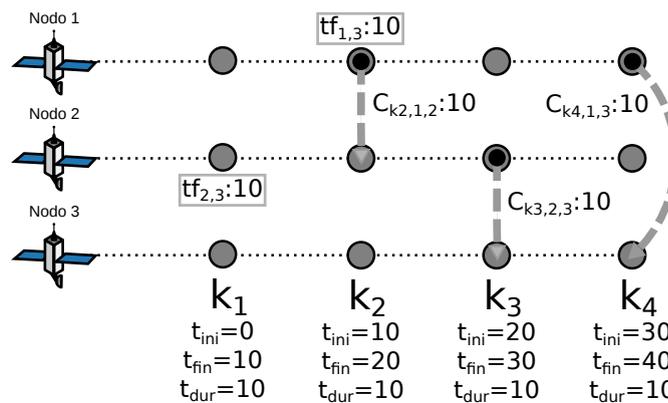


FIGURA 3.4: Plan de contactos de caso de ejemplo con dos flujos de tráfico

En la figura 3.5 se observan dos posibles flujos de estos datos. En 3.2 a), $N2$ envía sus 10 bundles a $N3$ aprovechando el contacto directo $C_{k3,2,3}$, mientras que $N1$ envía sus 10 bundles a $N2$ con la esperanza de seguir la ruta conformada por $C_{k2,1,2}$ y $C_{k3,2,3}$. Sin embargo, debido a que $N2$ reserva la capacidad total de $C_{k2,2,3}$ para su propio tráfico, no es capaz de enviar el tráfico de $N1$ por la ruta que el mismo $N1$ pretende utilizar. Estamos en este caso en una situación de congestión provocada por dos flujos de tráfico. Por otra parte, como se muestra en la Figura 3.2 b), si $N1$ enviara su tráfico utilizando solo el contacto $C_{k4,1,3}$, no existiría congestión y se entregaría la totalidad del tráfico. Resulta a primera vista contraintuitivo el hecho de que si se utilizara CGR con un PC con menor información, esto es, sin el contacto $C_{k2,1,2}$, se evitaría la congestión y se obtendría un rendimiento mayor. Sin embargo como veremos mas adelante, bajo ciertas condiciones resulta posible utilizar esquemas que tengan en cuenta este tipo de situaciones con el objetivo de evitar la congestión.

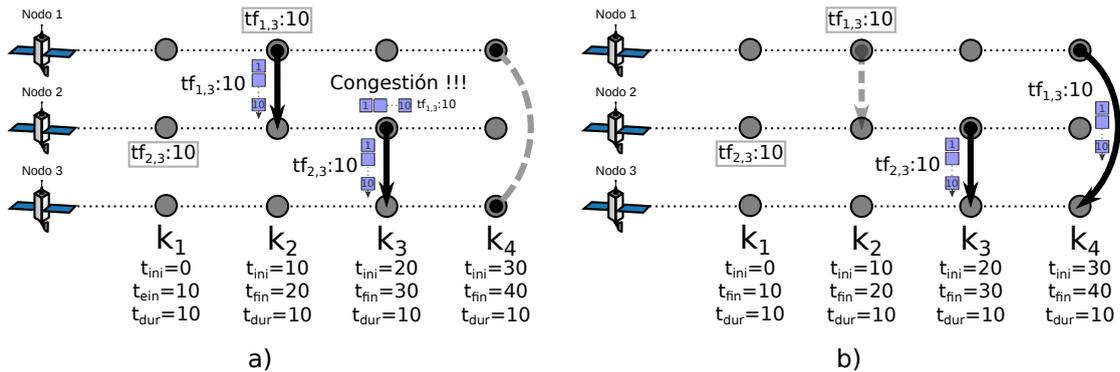


FIGURA 3.5: Flujo de dos tráficos: a) con congestión, b) sin congestión

Antes de desarrollar dichos esquemas, resulta útil contar con un modelo que proporcione el flujo óptimo y libre de congestión en redes DTN. Este modelo nos permitirá, por un lado, comprender las situaciones en que se genera la congestión, y por otro, desarrollar y comparar soluciones que sean aplicadas de manera distribuida.

3.5. Modelo de Flujo Óptimo de Múltiples Tráficos

El problema de enviar múltiples tráficos de forma óptima se conoce en la literatura como *multi-commodity flow* en Inglés [92]. En el contexto de redes DTN, los trabajos [15, 93, 94] presentaron modelos de programación lineal que permiten resolver este problema

en redes con conexiones intermitentes y planificadas. A continuación describimos una versión de estos modelos adaptada a nuestras necesidades particulares.

3.5.1. Coeficientes y Variables de Decisión

Retomamos aquí el modelado de la topología evolutiva como máquina de estados finitos descrita en el Capítulo 2. La topología se discretiza en K estados, donde cada estado $k_q \in K$ representa la red durante un intervalo de tiempo $[t_{q-1}, t_q]$. De esta manera, la red se define mediante estados $k_1, \dots, k_q, \dots, k_f$ que corresponden a intervalos de tiempo $[t_0, t_1], \dots, [t_{q-1}, t_q], \dots, [t_{f-1}, t_f]$. Dentro de cada k_q , la red se representa con un grafo G_{k_q} cuyos vértices $v \in V$ corresponden a nodos y cuyos arcos $e \in E_{k_q}$ corresponden a comunicaciones entre los nodos en k_q . La notación c_e representa la capacidad en una unidad arbitraria tal como bytes ó bundles que pueden ser enviados desde el nodo fuente hacia el nodo destino de un arco e durante el estado k_q correspondiente a dicho arco. Además, representaremos como b_v a la capacidad del buffer del nodo v en las unidades escogidas (bytes ó bundles). Con el objetivo de utilizar una notación compacta llamaremos I^v al conjunto de arcos e entrantes a un nodo v , mientras que llamaremos O^v al conjunto de arcos que salen del mismo.

Adicionalmente D constituye el conjunto de todas las demandas de tráfico $d_{t_q}^{y,z}$ generadas en el tiempo t_q desde el nodo y hacia el nodo z . Asumiremos en general que todas las demandas de tráfico se generan en el comienzo de algún estado. En caso de no ser así,

Coeficientes de Entrada

$t_q \in T$	Instantes de tiempo
$k_q = [t_{q-1}, t_q] \in K$	Estados / Intervalos de tiempo
$v \in V$	Nodos de la red
$e \in E_{k_q}$	Arcos de un grafo en el estado k_q
c_e	Capacidad del arco e en el estado k_q correspondiente
b_v	Capacidad de almacenamiento del nodo v
$d_{t_q}^{y,z} \in D$	Tráfico desde el nodo y al nodo z originado en el tiempo t_q

Variables de Salida

$\{X_e^{y,z}\}$	Tráfico desde y a z enviado en el arco e
$\{B_{t_q,v}^{y,z}\}$	Ocupación del buffer del nodo v en el tiempo t_q por el tráfico y, z

TABLA 3.1: Parámetros del modelo de flujo óptimo de tráfico

siempre es posible agregar un nuevo estado intermedio para contemplar la generación de dicho tráfico. Por otra parte, dentro de las variables de salida, $X_e^{y,z}$ constituye la cantidad del tráfico y, z que es enviada en el arco e , mientras que la variable $B_{v,t_q}^{y,z}$ es la cantidad del tráfico y, z almacenada en el nodo v en el tiempo t_q . Resumimos estos parámetros del modelo en la Tabla 3.1.

3.5.2. Función Objetivo y Restricciones

Una vez definidos los coeficientes y variables de decisión, procedemos a vincularlas mediante la función objetivo definida en 3.1 sujeta a las restricciones 3.2 a 3.7.

$$\text{minimizar: } \sum_{k_q \in K} \sum_{e \in E_{k_q}} \sum_{y \in V} \sum_{z \in V} w(k_q) * X_e^{y,z} \quad (3.1)$$

Sujeto a:

$$B_{t_q,v}^{y,z} = \begin{cases} B_{t_{q-1},v}^{y,z} + \sum_{e \in I^v} X_e^{y,z} - \sum_{e \in O^v} X_e^{y,z} + d_{t_q}^{y,z} & \text{if } y = v \\ B_{t_{q-1},v}^{y,z} + \sum_{e \in I^v} X_e^{y,z} - \sum_{e \in O^v} X_e^{y,z} & \text{if } y \neq v \end{cases} \quad \forall t_q, v, y, z \quad (3.2)$$

$$\sum_{y \in V} \sum_{z \in V} B_{t_q,v}^{y,z} \leq b_v \quad \forall t_q, v \quad (3.3)$$

$$\sum_{y \in V} \sum_{z \in V} X_e^{y,z} \leq c_e \quad \forall k_q, e \quad (3.4)$$

$$B_{t_0,v}^{y,z} = \begin{cases} d_{t_0}^{y,z} & \text{if } y = v \\ 0 & \text{if } y \neq v \end{cases} \quad \forall v, y, z \quad (3.5)$$

$$B_{t_q,v}^{y,z} \geq \begin{cases} d_{t_q}^{y,z} & \text{if } y = v \\ 0 & \text{if } y \neq v \end{cases} \quad \forall t_q > t_0, v, y, z \quad (3.6)$$

$$B_{t_f,v}^{y,z} = \begin{cases} \sum_{t_q \in T} \sum_{y \in V} d_{t_q}^{y,z} & \text{if } z = v \\ 0 & \text{if } z \neq v \end{cases} \quad \forall v, y, z \quad (3.7)$$

Queda conformado de esta manera un modelo de programación lineal cuya entrada consiste en un conjunto de tráficos ($d_{t_q}^{y,z}$) que deben ser enviados desde las fuentes (y) a los destinos (z) a través de una topología variante en el tiempo con ciertas capacidades de almacenamiento (b_v) y transmisión (c_e), y cuya salida consiste en los flujos ($X_e^{y,z}$) óptimos que deben seguir los distintos tráficos.

La función objetivo definida en 3.1 busca minimizar la suma de los productos $w(k_q) * X_e^{y,z}$, donde $w(k_q)$ es una función de ponderación que asigna un peso creciente a cada estado. De esta manera, mientras mayor sea el incremento causado por w , mayor será el costo de utilizar arcos posteriores en el tiempo, es decir, al minimizar el producto se prioriza entregar el tráfico lo antes posible. Por otro lado, mientras menor sea el costo creciente de w , mayor importancia se atribuye a utilizar el menor número posible de arcos.

Con respecto a las restricciones, la ecuación 3.2 define la ocupación del buffer de cada nodo (v) en cada instante de tiempo (t_q) teniendo en cuenta el estado en el tiempo anterior (t_{q-1}), los flujos entrantes ($e \in I^v$) y salientes ($e \in O^v$) en el intervalo $[t_{q-1}, t_q]$, y el tráfico generado en t_q ($d_{t_q}^{y,z}$). Por su parte, la ecuación 3.3 limita la ocupación del buffer ($B_{t_q,v}^{y,z}$) de cada nodo de acuerdo a b_v , mientras que la ecuación 3.4 limita la cantidad de tráfico que puede fluir por cada arco ($X_e^{y,z}$) de acuerdo a c_e .

Además, la restricción 3.5 inicializa los buffers en el instante inicial (t_0) mientras que la restricción 3.6 asegura que los tráficos generados en estados intermedios no sean enviados en estados anteriores. Por último, la ecuación 3.7 asegura que la suma de todos los tráficos y, z generados a lo largo del tiempo residan finalmente en el buffer del nodo destino correspondiente ($B_{t_f,v}^{y,z}$).

3.5.3. Ejemplo

Una buena manera de comprender y visualizar la respuesta que proporciona este modelo consiste en analizar una topología simple tal como la de la Figura 3.4, donde, como vimos, se podía generar congestión cuando múltiples tráficos se envían por determinadas rutas. En este caso el modelo brinda una solución como la representada en la Figura 3.6. Además de los flujos de tráfico, se puede observar que se retorna la cantidad de bundles almacenados en cada buffer tanto al comienzo como al final de cada estado. En este caso, solo se tuvieron en cuenta limitaciones en las capacidades de los contactos y no en los

buffers. Notar que la solución del modelo coincide exactamente con la que analizamos en la Figura 3.5 b).

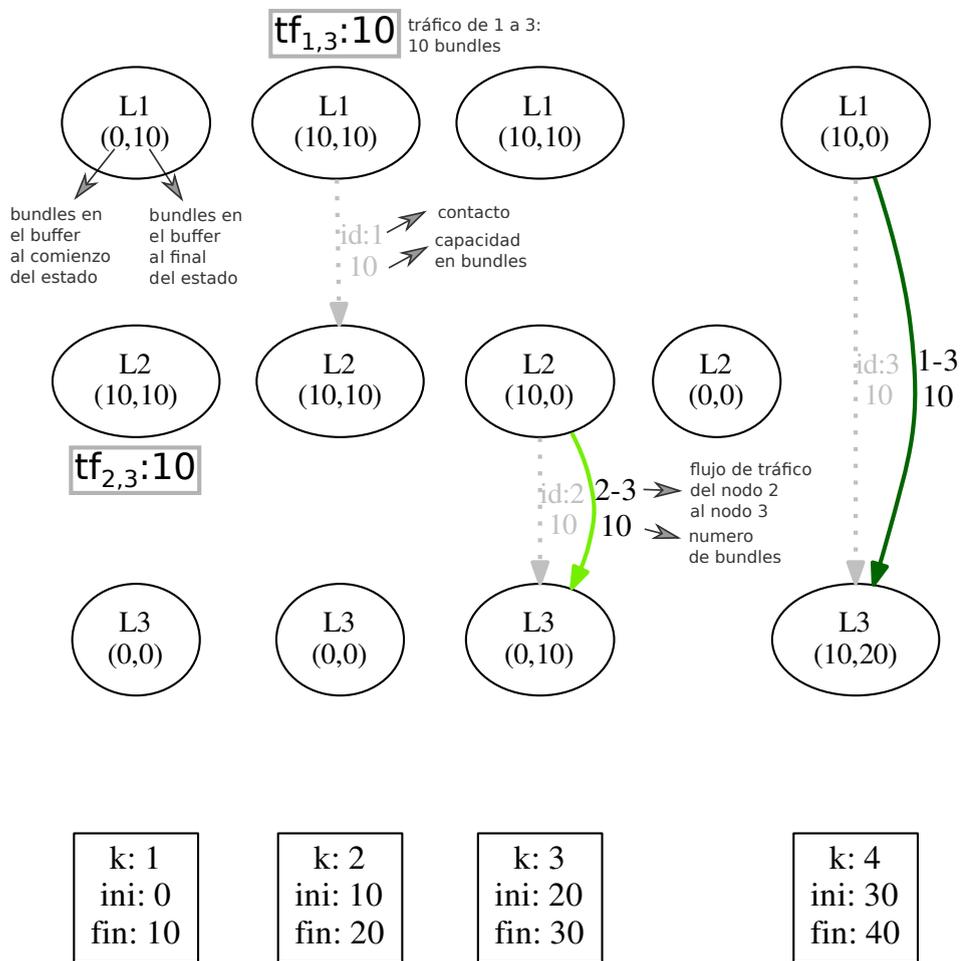


FIGURA 3.6: Solución proporcionada por modelo de flujos óptimos

3.6. Multi Graph CGR (MG-CGR)

3.6.1. Concepto

Como vimos en la sección anterior, si se dispone de una visión global tanto de la topología como del tráfico generado, es posible obtener soluciones óptimas de flujo y libres de congestión para múltiple tráficos. Además, la predictibilidad del tráfico es una característica típica de aplicaciones de redes espaciales tales como las de observación de la Tierra donde, en general, un MOC determina de antemano el uso de instrumentos a bordo y de las cargas útiles. De esta manera, tanto la cantidad de datos como el tiempo de generación pueden ser lo suficientemente predecibles como para motivar un diseño o

filtrado del plan de contactos que permita obtener ciertas ventajas en base a esta valiosa información adicional.

Por otra parte, debido a que podrían existir redes espaciales que no presenten un tráfico predecible, CGR se pensó mas como un esquema general en el que la única información de entrada es el plan de contactos planificado. En este esquema todos los nodos de la red reciben el mismo plan de contactos.

Sin embargo, en esta sección proponemos un enfoque novedoso que permite aprovechar la predictibilidad del tráfico (en los casos en que sea posible) para diseñar un plan de contactos *específico* para cada nodo, buscando en última instancia que los tráficos de toda la red fluyan sin congestión, por mas que el enrutamiento y encolamiento se realice de forma distribuida.

3.6.2. Descripción Algorítmica

Llamamos *Multi Graph CGR (MG-CGR)* a un esquema de planificación y enrutamiento que como se representa en la Figura 3.7 consiste de dos etapas:

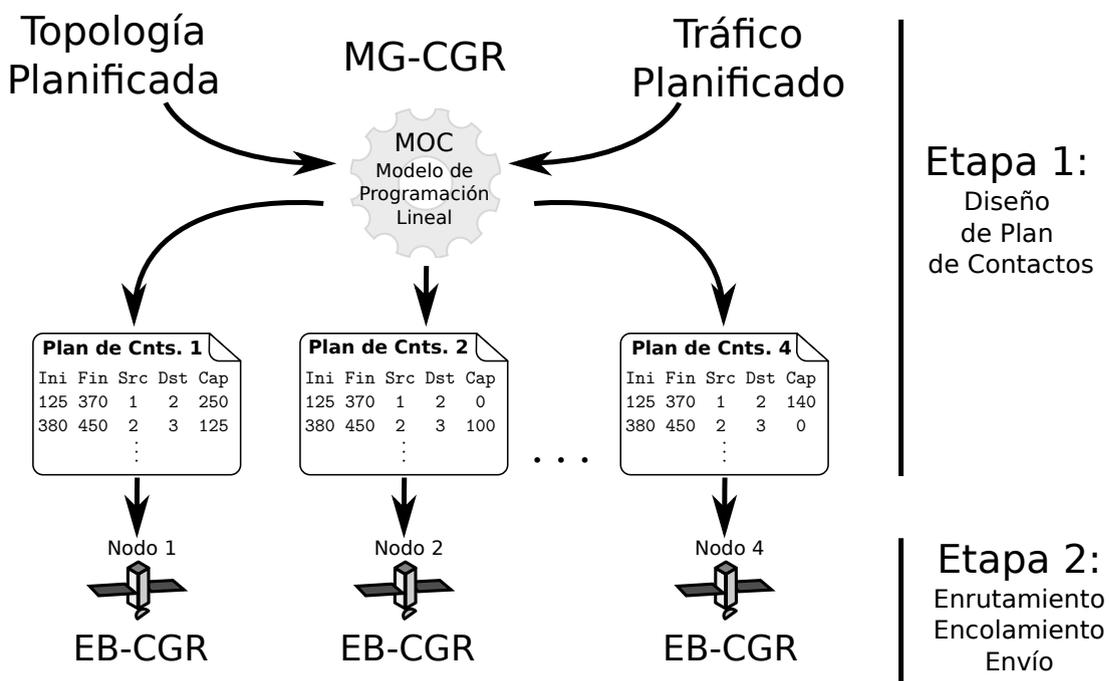


FIGURA 3.7: Representación gráfica de esquema MG-CGR

i) En una primera instancia se recibe como entrada la topología (CP) y el tráfico planificado (D), se ejecuta el modelo de programación lineal descrito en 3.5, y se deriva un plan de contactos específico para cada nodo. El plan de contactos CP_{nx} para un nodo nx se obtiene recorriendo todos los contactos de CP y modificando la capacidad de cada contacto de acuerdo a los valores que las variables $X_e^{y,z}$ tienen para el tráfico generado únicamente por el nodo nx . Si este proceso se llevara a cabo para el caso del ejemplo 3.6 se obtendría:

- un CP_{n1} con una capacidad de 10 bundles para el contacto con $id = 3$, y capacidad 0 para los contactos con $id = 1$ y $id = 2$.
- un CP_{n2} con una capacidad de 10 bundles para el contacto con $id = 2$, y capacidad 0 para los contactos con $id = 1$ y $id = 3$.
- un CP_{n3} con una capacidad de 0 para los contactos con $id = 1$, $id = 2$ y $id = 3$.

Notar que este proceso constituye una asignación de capacidades, donde se reservan porciones de contactos (o contactos completos) para determinados tráficos. La suma de todos los CP_{nx} debe ser menor o igual al CP inicial en términos de capacidad.

ii) En una segunda instancia los nodos utilizan su plan de contactos CP_{nx} para llevar a cabo el enrutamiento, encolamiento y envío de bundles. Para ello, es necesario que cada nodo calcule las rutas *únicamente* para el tráfico generado localmente y que coloque dicha ruta en la cabecera del bundle mediante la extensión EB-CGR descrita en 2.5.1. Por otro lado, los nodos intermedios ny deben respetar la decisión tomada por el nodo que generó un determinado tráfico utilizando contactos que en su plan específico CP_{ny} podrían tener capacidad 0. De esta manera, los contactos con capacidad 0 indican que un nodo puede utilizarlos para enrutar tráfico en tránsito (de otros nodos), mientras que los contactos con capacidad distinta de 0 indican que dicho nodo puede utilizarlos para enrutar tráfico generado localmente. Podría darse además el caso de un contacto que tuviera una capacidad total C , de la cual una porción C_1 sea asignada para el nodo nx , mientras que una porción C_2 sea asignada para un nodo ny . En este caso, ny podría usar solo la capacidad C_2 para su tráfico generado localmente, mientras que debería utilizar el remanente de C para respetar la decisión tomada por nx .

3.6.3. Implementabilidad

Cuando se realizan los procesos descritos, y el flujo que siguen los bundles coincide exactamente con la asignación realizada por el modelo de programación lineal, decimos que la solución es *implementable*. Es decir, los nodos ejecutan EB-CGR de forma distribuida, con sus planes de contactos específicos, y respetan la utilización de capacidades asignada según el modelo. Sin embargo, como veremos a continuación, pueden existir situaciones que lleven a que una solución no sea implementable de forma distribuida y que requieran por lo tanto de ciertas adaptaciones.

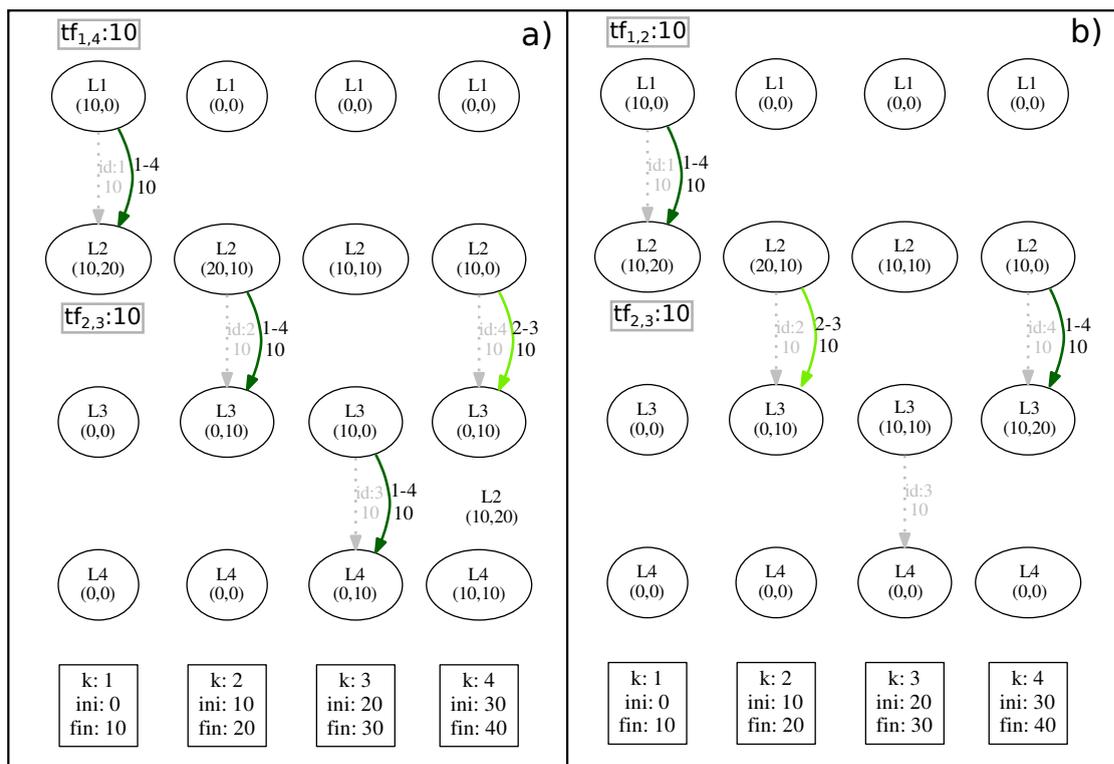


FIGURA 3.8: Problema de encolamiento por nodo

3.6.3.1. Problema de Encolamiento por Nodo

En la Figura 3.8 tenemos un caso de 4 nodos y 4 estados donde tanto $L1$ como $L2$ generan tráficos en $t = 0$ con destinos $L4$ y $L3$ respectivamente. En 3.8 a) se puede observar la solución obtenida con el modelo, donde se obtiene un CP_{L1} con los contactos $id = 1,2,3$ con capacidades de 10 y el contacto $id = 4$ con capacidad 0, y un CP_{L2} con el contacto $id = 4$ con capacidad 10 y los contactos $id = 1,2,3$ con capacidad 0.

Sin embargo, en 3.8 b) podemos ver el flujo de bundles cuando se ejecuta EB-CGR de forma distribuida, donde claramente existen discrepancias con respecto a lo planeado. Para poder explicar este problema resulta necesario resaltar que si bien CGR realiza el cálculo de la mejor ruta teniendo en cuenta los contactos, el encolamiento se realiza seleccionando el *próximo nodo* como el nodo receptor del primer contacto (Algoritmo 1, línea 24). El hecho de tener un encolamiento por nodo y no por contacto, lleva en este caso a que si bien $L2$ calcula la ruta correcta conformada por el contacto $id = 4$ para el tráfico $tf_{2,3}$, el encolamiento de los bundles para $L3$ sea despachado en un contacto anterior. Por lo tanto, una posible solución consiste en realizar un encolamiento por contacto. De forma alternativa, se puede mantener un encolamiento por nodo pero restringiendo a que el envío se haga en un contacto determinado. De esta manera, el flujo de 3.8 b) coincidiría exactamente con el flujo de 3.8 a), y podríamos afirmar que esta solución es implementable.

3.6.3.2. Problema de Múltiples Tráficos por Nodo

Un segundo problema puede surgir cuando se intenta enviar mas de un tráfico por nodo.

En la Figura 3.9 tenemos un caso de 3 nodos y 4 estados donde $L1$ genera los tráficos $tf_{1,2}$ y $tf_{1,3}$ en los tiempos $t = 0$ y $t = 10$ respectivamente. En 3.9 a) se puede observar

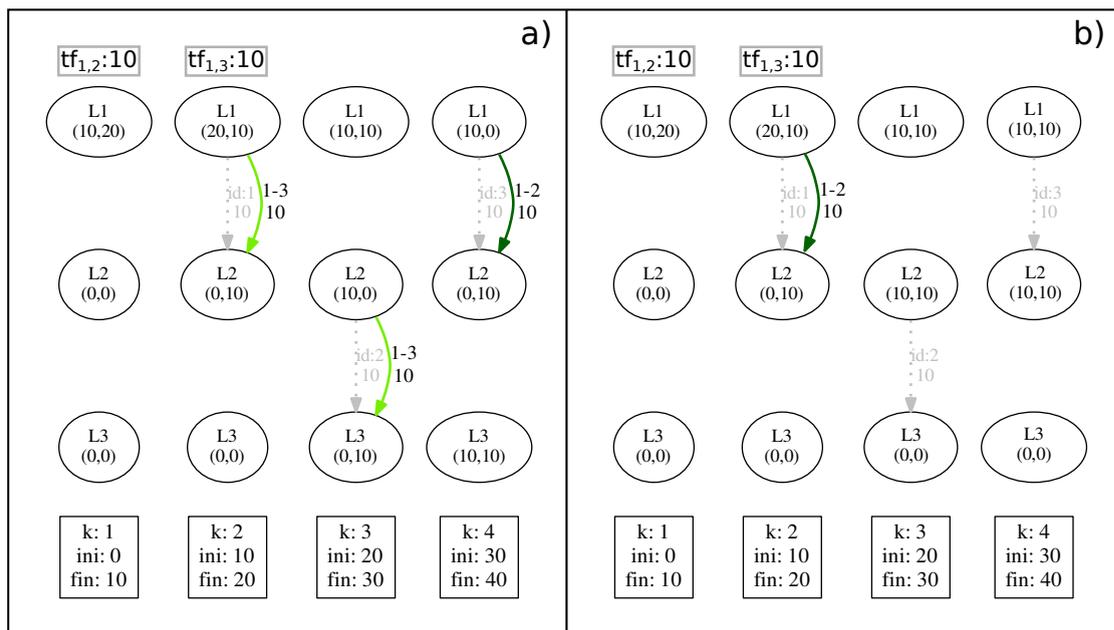


FIGURA 3.9: Problema de múltiples tráficos por nodo

la solución obtenida con el modelo, donde se obtiene un CP_{L1} con los contactos $id = 1, 2, 3$ con sus capacidades completas. Sin embargo, en 3.9 b) observamos el flujo cuando se ejecuta EB-CGR de forma distribuida, donde nuevamente existen diferencias con respecto a lo planeado. En este caso, la diferencia se debe a que el encolamiento de tráfico llevado a cabo por CGR se efectúa de forma cronológica a medida que se generan dichos tráficos. De esta manera, mientras que EB-CGR envía el tráfico $tf_{1,2}$ por la ruta con mejor tiempo de entrega (3.9 b)), la solución del modelo pretendía que se utilice una ruta posterior para dicho tráfico, en aras de que el tráfico $tf_{1,3}$ (generado después) pueda ser entregado a destino.

Una forma de solucionar este problema podría basarse en entregar un plan de contactos por tráfico en lugar de un plan de contactos por nodo. Sin embargo, el costo de enviar esta información no se justifica si se compara con los beneficios que se obtendrían. Si pretendemos soluciones más escalables, debemos mantener la condición de generar un único plan de contactos por nodo. Como se muestra en la Figura 3.10, se tienen dos alternativas para obtener soluciones óptimas que sean además implementables:

- 1) Modificar el modelo LP para que proporcione soluciones implementables, o bien, filtrar las soluciones del modelo que sean implementables.
- 2) Mejorar soluciones implementables con el fin de obtener o acercarse a una solución óptima.

Con respecto a la primera opción, si bien es una alternativa válida de ser explorada, no resulta evidente la manera de agregar restricciones adicionales al modelo que aseguren la implementabilidad de la solución obtenida. Por otra parte, el filtrado de soluciones óptimas requiere de verificaciones adicionales que las ejecute de forma distribuida para

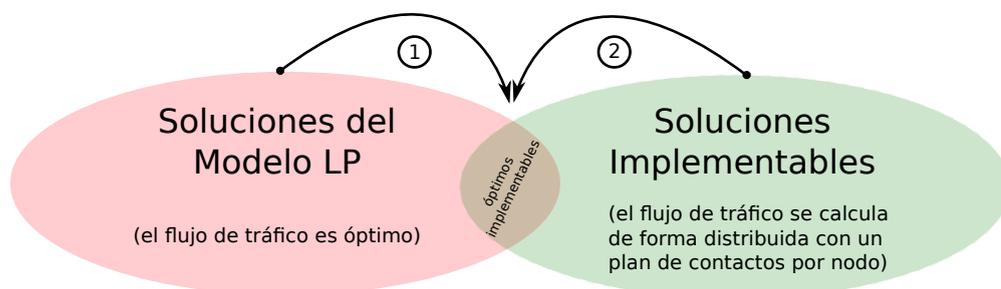


FIGURA 3.10: Alternativas para obtener soluciones óptimas e implementables

constatar si se respetan los flujos de tráfico proporcionados por el modelo. En el caso de que la superposición de soluciones óptimas e implementables sea nula, este procedimiento sería incapaz de brindar la solución buscada, e incluso, incapaz de asegurar si hay o no superposición de los conjuntos de soluciones mencionados.

Con respecto a la segunda opción, el hecho de partir de soluciones implementables permite establecer una cota de tiempo superior en la cual se requiere de una solución. En las siguientes secciones exploramos esta alternativa mediante esquemas que brindan soluciones que siempre resultan implementables y que pueden ser mejoradas para acercarse a soluciones óptimas.

3.7. Global Path Aware CGR (GPA-CGR)

3.7.1. Concepto

Como vimos en la sección anterior, en escenarios donde el tráfico es predecible, es posible utilizar esta información adicional para mejorar el uso de recursos y optimizar los tiempos de entrega. Sin embargo, la utilización del esquema MG-CGR está restringida a casos donde existe como máximo un único tráfico por nodo. Si bien esta situación no es usual, podemos rescatar la idea de asignar porciones de contactos a los nodos mediante planes de contacto específicos. El objetivo es obtener soluciones que sean implementables aún cuando existen múltiples tráficos por nodo.

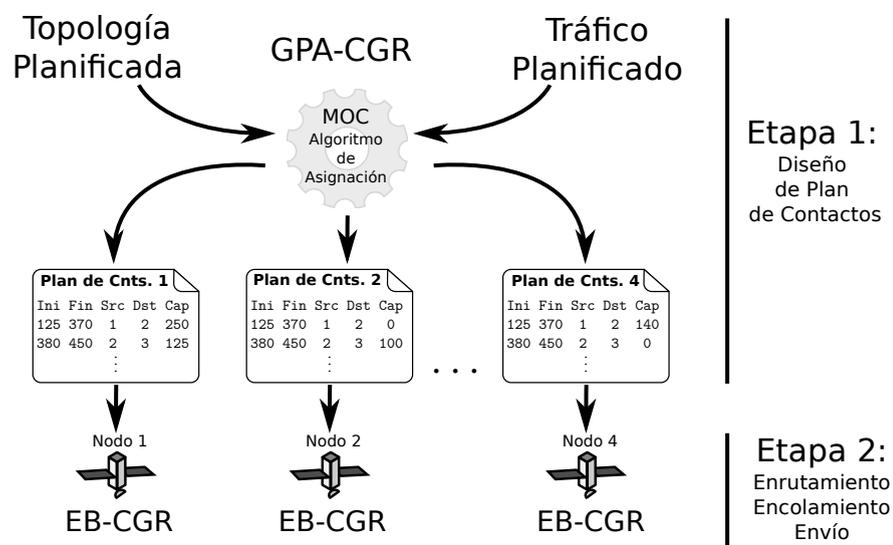


FIGURA 3.11: Representación gráfica del esquema GPA-CGR

3.7.2. Descripción Algorítmica

Llamamos *Global Path Aware CGR (GPA-CGR)* a este nuevo esquema que como se representa en la Figura 3.11 consiste de dos etapas similares a las utilizadas para MG-CGR. La diferencia radica en la primera etapa, donde en lugar de realizar la asignación de contactos mediante un modelo LP, se utiliza el Algoritmo de asignación 6 que explicamos a continuación. Este algoritmo garantiza la implementabilidad al realizar asignaciones de los tráficos a los contactos de forma cronológica con respecto a los tiempos de generación de dichos tráficos.

El procedimiento recibe como entrada la topología *Topology* y el tráfico planificado *Traffic*, y retorna a la salida un arreglo de planes de contactos *Cplans[]* (uno para cada nodo) junto a una estructura auxiliar *RoutedTraffic* que contiene las asignaciones de tráfico a los contactos. En la línea 1 se obtienen los tráficos (*generators*) ordenados por tiempo de generación a partir del tráfico planificado. Un *generator* g_i consiste de un volumen de tráfico en bytes, un tiempo de generación en segundos y los nodos origen y destino. Luego, en la línea 2 se utiliza una rutina similar a la que utiliza CGR para calcular una tabla de rutas global (*RouteTable*) que contiene todas las rutas temporales posibles que cada *generator* puede utilizar para llegar a su nodo destino. En las líneas 3-12 cada *generator* se “enruta” con un proceso que consiste primero en obtener la mejor ruta (*Route*) con capacidad remanente de *RouteTable* en la línea 5, y luego realizar

Algoritmo 6: GPA-CGR

```

input : Topology, Traffic
output: Cplans[], RoutedTraffic
global : RoutedTraffic, BufferCap

1 Generators  $\leftarrow$  GetSortedGenerators(Traffic);
2 RouteTable  $\leftarrow$  ComputeRouteTable(Topology, Generators);
3 for  $g_i \in$  Generators do
4   while  $g_i.$ GetCurrentDataLength()  $>$  0 do
5     Route  $\leftarrow$  getBestRoute( $g_i, RouteTable$ );
6     if Route  $\neq \emptyset$  then
7       RouteGenerator( $g_i, Route$ );
8       UpdateBuffersCapacity();
9       DecreaseContactsCapacity();
10      UpdateRouteTable();
11    else
12      skip  $g_i$ ;
13 for  $n_i \in$  Topology.Nodes do
14    $Cplans.$ At( $n_i$ )  $\leftarrow$  GetContactPlan(RoutedTraffic, n_i);
15 return [Cplans[], RoutedTraffic]

```

una asignación de flujo a los contactos correspondientes de la ruta en la estructura *RoutedTraffic* en la línea 7. Luego, una estructura llamada *BufferCap* se actualiza en la línea 8, la capacidad de los contactos se reduce en la línea 9 y *RouteTable* se actualiza en la línea 10. La estructura *BufferCap* tiene en cuenta la ocupación del buffer de los nodos como una función del tiempo de la misma manera que lo hace el esquema LPA-CGR 3.3. La única diferencia consiste en que aquí esta estructura es global y conoce todos los nodos de red y el tráfico generado por los mismos. Finalmente, en las líneas 13 y 14 se obtiene un plan de contactos específico para cada nodo utilizando la estructura *RoutedTraffic* y considerando solo el tráfico generado por ese nodo. Los demás contactos se mantienen con una capacidad de 0, lo que implica que el contacto existe y que puede ser utilizado para enviar el tráfico de otros nodos pero que no se puede tener en cuenta para los tráficos generados localmente. En consecuencia, la suma de todos los planes de contactos derivados *Cplans*[] da como resultado un plan de contactos con una capacidad inferior o igual a la topología original.

De la misma manera que en MG-CGR, una vez que se distribuyen todos los planes de contactos, cada nodo puede usar EB-CGR para enrutar el tráfico a su destino. Dado que la ruta se guarda en el encabezado de los bundles, el tráfico que fluye a través de nodos intermedios no requiere de un recálculo de ruta ni se necesita conocer el valor de capacidad de topología originalmente reservado a otros nodos. Una simple validación de ruta es suficiente para asegurar que existan los contactos necesarios en el plan de contactos local.

Un aspecto importante a considerar en este esquema lo constituye la precisión en la predictibilidad del tráfico. Dado que los planes de contactos se diseñan para acomodar solamente el tráfico previsto, se cuenta con una limitada capacidad para enviar un tráfico potencialmente mayor. Para evitar esta situación, se pueden incorporar márgenes de error en los cálculos de capacidad o incluso distribuir un segundo plan de contactos de respaldo con las capacidades marginales restantes en el sistema para que todos los nodos las consideren en el caso de eventos no planificados. Sin embargo, este enfoque no dispone de la capacidad de evitación de la congestión del esquema original. Dejamos el análisis y la definición de esta estrategia como investigación futura.

3.8. Optimización Evolutiva de GPA-CGR (EGPA-CGR)

3.8.1. Concepto

Aunque la solución obtenida por GPA-CGR permite un flujo de tráfico libre de congestión, es posible obtener diferentes soluciones que también sean libres de congestión obtenidas mediante otros mecanismos. Por lo tanto, un aspecto clave a considerar es la cuestión de si la solución proporcionada por GPA-CGR se puede optimizar en función de alguna métrica tal como el tiempo de entrega de los tráfico o el uso de recursos de la red.

Una forma de explorar entre las diferentes soluciones podría ser variando/eliminando algunos contactos de la topología y luego aplicando el algoritmo GPA-CGR a la topología modificada. Como veremos, este método puede conducir a soluciones que, además de ser libres de congestión, son mejores soluciones en términos de alguna métrica tal como bundles entregados o el uso de recursos de la red. Esto da como resultado un complejo problema de búsqueda combinatoria que puede abordarse con meta-heurísticas tales como los algoritmos evolutivos [95]. En general, estos algoritmos se basan en la noción de competencia entre individuos (soluciones) de una población, imitando de esta manera la evolución de las especies en la naturaleza [96].

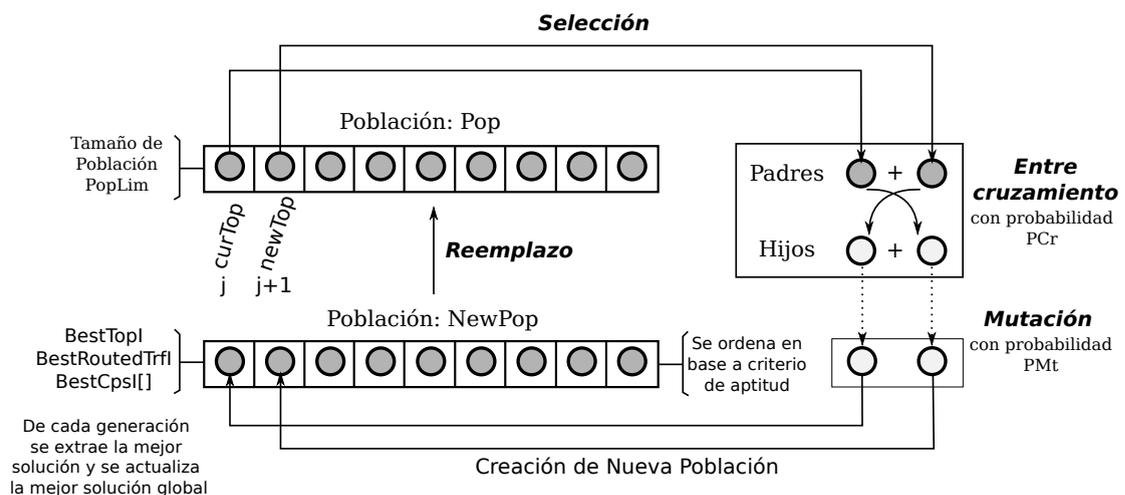


FIGURA 3.12: Representación gráfica del esquema EGPA-CGR

3.8.2. Descripción Algorítmica

A continuación proporcionamos una descripción de *Evolutionary Global Path Aware CGR* (*EGPA-CGR*) mediante el Algoritmo 7, representado además gráficamente en la Figura 3.12. Este algoritmo recibe como entrada la topología (*Topology*), el tráfico planificado (*Traffic*), la probabilidad de entrecruzamiento o *crossing-over* (*PCr*), la probabilidad de mutación (*PMt*), el tamaño límite de la población (*PopLim*) y el número de iteraciones (*Iters*) como criterio de detención. El resultado es el mismo que GPA-CGR: un plan de contactos específico para cada nodo $Cps[]$ y una asignación de tráficos *RoutedTrf* que es útil para determinar cuando una solución es mejor que otra (*fitness*).

En la línea 1, la población inicial (*Pop*) se inicializa con topologías *mutadas* que son generadas al modificar la topología de entrada mediante la eliminación aleatoria de un número de contactos. En la línea 2, la población inicial se ordena de acuerdo a un criterio de aptitud que consiste en la priorización de determinadas métricas sobre otras. Para obtener dichas métricas se aplica el algoritmo GPA-CGR a cada topología, se obtiene la estructura *RoutedTrf* resultante y finalmente se obtienen las métricas *i*) proporción de paquetes entregados, *ii*) tiempos de contacto del sistema normalizado y *iii*) tiempo de entrega del tráfico. Estas métricas se explicarán en la Sección de Evaluación 3.9 y el usuario tiene la posibilidad de ponderar cada una de ellas para modificar el criterio de aptitud y ajustar así el enfoque de la solución objetivo obtenida en el proceso de búsqueda.

El bucle *for* en las líneas 3-36 recorre los individuos de la población (*Pop*) y les aplica operaciones de entrecruzamiento, mutación y selección con el objetivo de obtener una nueva población (*NewPop*) que conduzca a mejores soluciones. Las líneas 4-8 obtienen dos individuos (*CurTop* y *NewTop*) que serán sometidos a operaciones de entrecruzamiento y mutación. El ciclo *while* entre las líneas 7 y 28 tiene el objetivo de elegir el individuo *NewTop* con el que se realizará el cruzamiento. El cruce se realiza en concordancia con la probabilidad *PCr* en la línea 9 y 10 y se obtienen las topologías hijas *TopA* y *TopB*. Las líneas 11 y 12 mutan esas topologías de acuerdo con la probabilidad *PMt* y finalmente se guardan en *NewPop* en las líneas 13 y 14. Las líneas 15 y 16 permiten salir de cualquier iteración si la población crece más de un cierto límite. En la línea 17 se elimina un individuo de *Pop* una vez que logra una operación de cruce con otro individuo. Las líneas 18-21 restauran los iteradores para obtener nuevos individuos

Algoritmo 7: EGPA-CGR

```

input : Topology, Traffic, PCr, PMt, ITERS, PopLim
output: Cps[], RoutedTrf
global : Pop, NewPop
global : BestTop, BestTopI
global : BestRoutedTrf, BestRoutedTrfI
global : BestCps[], BestCpsI[]

1 InitializePopulation(PopLim);
2 SortPopulation(Pop);
3 for  $i \leftarrow 1$  to ITERS do
4    $j \leftarrow Pop.Begin()$ ;
5    $CurTop \leftarrow Pop.At(j)$ ;
6    $j \leftarrow j + 1$ ;
7   while  $j \neq Pop.End()$  do
8      $NewTop \leftarrow Pop.At(j)$ ;
9     if  $Random(0, 1) \leq PCr$  then
10       $TopA, TopB = CrossOver(CurTop, NewTop)$ ;
11       $Mutate(TopA, PMt)$ ;
12       $Mutate(TopB, PMt)$ ;
13       $NewPop.PushBack(TopA)$ ;
14       $NewPop.PushBack(TopB)$ ;
15      if  $NewPop.Size() \geq PopLim$  then
16         $\lfloor$   $break$ ;
17       $Pop.PopFront()$ ;
18       $j \leftarrow Pop.Begin()$ ;
19      if  $j \neq Pop.End()$  then
20         $curTop = Pop.At(j)$ ;
21         $j \leftarrow j + 1$ ;
22      else
23         $j \leftarrow j + 1$ ;
24        if  $j = Pop.End()$  then
25           $Pop.PopFront()$ ;
26           $j \leftarrow Pop.Begin()$ ;
27           $CurTop \leftarrow Pop.At(j)$ ;
28           $j \leftarrow j + 1$ ;
29      SortPopulation(NewPop);
30       $BestTopI = NewPop.Front()$ ;
31       $[BestCpsI[], BestRoutedTrfI] = GPA-CGR(BestTopI, Traffic)$ ;
32      if  $IsBetter(BestRoutedTrfI, BestRoutedTrf)$  then
33         $BestRoutedTrf = BestRoutedTrfI$ ;
34         $BestCps[] = BestCpsI[]$ ;
35       $Pop = NewPop$ ;
36       $NewPop.Clear()$ ;
37 return  $[BestCps[], BestRoutedTrf]$ 

```

de la población cuando el cruce se realizó efectivamente y las líneas 23-28 realizan la misma tarea cuando no se realizó el cruce. En la línea 29 se ordenan los individuos de *NewPop* una vez que ha finalizado una iteración y la línea 30 obtiene la mejor topología individual de la iteración (*BestTopI*). La línea 31 obtiene la asignación de flujo (*BestRoutedTrfI*) y el arreglo de planes de contactos derivados (*BestCpI*) llamando al procedimiento GPA-CGR. Las líneas 32-34 actualizan la mejor solución global. Las

líneas 35-36 asignan *NewPop* a *Pop* y borran *NewPop* para comenzar una nueva iteración del algoritmo genético. Finalmente, la mejor solución global que consiste en una asignación de flujo y un arreglo de planes de contactos se retorna en la línea 37.

3.9. Evaluación de Rendimiento

3.9.1. Caso de Estudio

Con el fin de evaluar las técnicas de evitación de la congestión descritas, proponemos el estudio de un caso realista que consiste de una formación lineal de satélites de órbita baja. Existen varias razones que respaldan la formación de vuelo lineal, utilizada por ejemplo en sistemas tales como la constelación A-Train de la NASA [97]. Entre ellas, si las naves espaciales están lo suficientemente cerca en esta formación, perciben las mismas perturbaciones de gravedad, lo cual permite ahorros significativos de propulsión (generalmente utilizados para el mantenimiento de los satélites en determinadas posiciones relativas). Además, es una formación deseable desde el punto de vista del lanzador ya que no requiere una órbita de transferencia para entregar los satélites a su posición final. Por otra parte, desde una perspectiva de misión de observación de la Tierra, permite obtener imágenes estereoscópicas o de gran angular de observación de la Tierra. Si se suministra con enlaces intersatelitales (ISL) y con un enfoque DTN adecuado de gestión de datos, el sistema propuesto puede compartir transponders de enlace descendente optimizando el uso de recursos limitados tales como la potencia y las asignaciones de frecuencia.

Comienzo de la Topología	Jan-1st, 2016, 0hs 0min 0sec
Final de la Topología	Jan-1st, 2016, 21hs 43min 18sec
Coefficiente Bstar (/ER)	90039
Inclinación (grados)	98°
RAAN (grados)	0°
Eccentricity	9152
Argumento de Perigeo (grados)	0°, 5°, 10°, y 15°
Anomalía media (grados)	0°
Movimiento medio (rev/día)	15,07561758 rev/día

TABLA 3.2: Parámetros orbitales e intervalos de tiempo de caso de estudio de formación lineal

diferentes técnicas de evitación de la congestión. Además, con el objetivo de analizar el impacto de los buffers limitados en el rendimiento de los algoritmos propuestos, se agregan restricciones adicionales. Particularmente, cada satélite tiene un buffer pequeño de 9 MBytes y la estación terrena un buffer de gran tamaño de 100 GBytes para poder recibir todo el tráfico generado.

Probablemente, la métrica más importante para comparar CGR, LPA-CGR, GPA-CGR y EGPA-CGR sea el tráfico total efectivamente entregado a la estación terrena. Sin embargo, también es necesario comprender que tan eficientemente se logró dicha entrega. Por lo tanto, también medimos los tiempos de contacto utilizados y el tiempo de entrega del tráfico. De ahora en adelante, nos referiremos a estas métricas como *tasa de bundles entregados*, *uso de tiempos de contacto normalizado* y *tiempo de entrega* respectivamente. La *tasa de bundles entregados* se calcula como los bundles entregados sobre los bundles generados, mientras que el *uso de tiempos de contacto normalizado* se calcula como el uso de tiempos de contacto para comunicación sobre los bundles entregados. Ésta última constituye una manera de hacer comparaciones más justas entre los esquemas de enrutamiento cuando los bundles no pueden alcanzar sus destinos. Otro aspecto a mencionar es que en este estudio nos enfocamos en esquemas proactivos para evitar la congestión sin tener en cuenta mecanismos reactivos como la transferencia de custodia. Por lo tanto, los bundles son descartados cuando llegan a un nodo con un buffer repleto. Finalmente, para determinar y analizar estas métricas, todos los algoritmos y topologías se implementaron en una herramienta de simulación que describimos brevemente a continuación.

3.9.2. Herramienta de Simulación: Dtnsim

Nuestro grupo de trabajo desarrolló Dtnsim con el objetivo de modelar, estudiar y evaluar el comportamiento de algoritmos de enrutamiento en redes DTN planificadas. Está basada en una simplificación de la arquitectura por capas descrita en el RFC4838 [32], se desarrolló en el framework Omnet++ [98], fué presentada en [99], y se encuentra disponible para ser utilizada como software libre ¹. Además, DtnSim cuenta con la capacidad de generar estadísticas tanto escalares como vectoriales que permiten cuantificar y analizar el comportamiento de la red en una simulación. Se dispone de métricas tanto

¹Repositorio Público de DtnSim: <https://bitbucket.org/lcd-unc-ar/dtnsim>

globales como locales a cada nodo: bundles generados, transmitidos y recibidos, uso de la memoria (SDR), cantidad de veces que se ejecuta el algoritmo de enrutamiento, reruteos, tiempos de entrega, etc. En la Figura 3.14 se observa una captura de la interfaz gráfica de DtnSim donde dos nodos intercambian tráfico mediante el pasaje de mensajes.

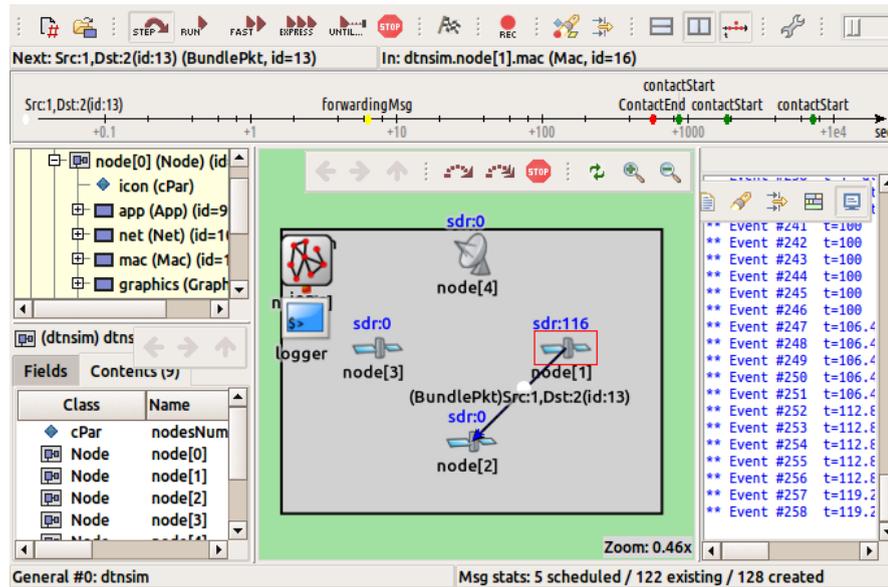


FIGURA 3.14: DtnSim: Simulador de redes DTN

3.9.3. Análisis de Resultados

Los resultados de simulación obtenidos se resumen en la Figura 3.15, donde el eje de abscisas representa una carga de tráfico creciente de $\rho = 0,1$ (59 paquetes o 737,5 KB por nodo) hasta $\rho = 1$ (592 paquetes o 7,4 MB por nodo). En el eje de ordenadas tenemos la *tasa de entrega de bundles*, el *uso de tiempos de contacto normalizado* y el *tiempo de entrega* para los esquemas CGR, LPA-CGR con retorno a nodo previo habilitado, GPA-CGR y EGPA-CGR. En particular, el retorno al nodo previo está deshabilitado para CGR ya que los ciclos de enrutamiento (*loops*) hacen que su métrica *uso de tiempos de contacto normalizado* aumente drásticamente. El esquema EGPA-CGR está configurado para mejorar las métricas en el siguiente orden: primero *tasa de entrega de bundles*, luego *uso de tiempos de contacto normalizado* y finalmente *tiempo de entrega*. La Figura 3.15 c) ilustra además 4 áreas resaltadas que representan los estados donde la formación tiene contactos con el segmento terreno a través de los arcos $c_{k4,1,0}$, $c_{k8,4,0}$, $c_{k12,1,0}$ y $c_{k16,4,0}$ respectivamente.

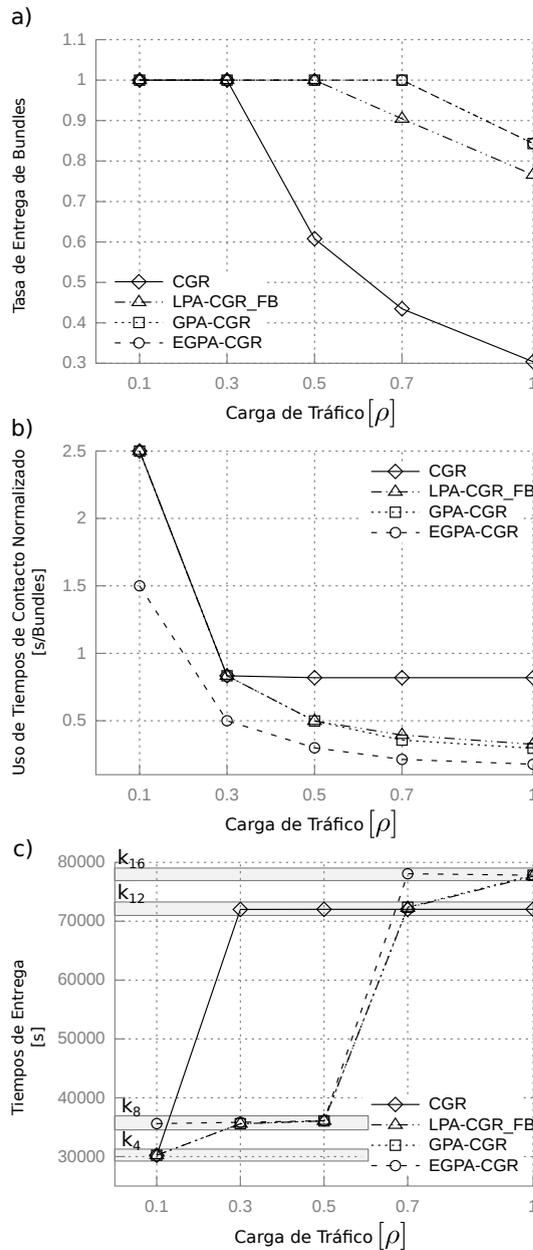


FIGURA 3.15: Resultados de simulación

En general, para una baja carga todos los mecanismos de gestión de la congestión proporcionan una entrega óptima como se muestra en la Figura 3.15 a). Sin embargo, se evidencia un punto de inflexión para CGR a partir de $\rho = 0,3$ (177 bundles por nodo que suman 708 en el sistema). Para explicar adecuadamente este comportamiento, la Figura 3.16 a) ilustra los flujos de tráfico para CGR en el caso particular de $\rho = 1$. Aquí, CGR solo está utilizando los contactos $c_{k4,1,0}$ y $c_{k12,1,0}$ para llegar al nodo de destino 0 con una proporción decreciente de entrega de bundles a medida que aumenta la carga de tráfico. Esto se debe a que cuando cada nodo enruta su tráfico, considera un contacto utilizable

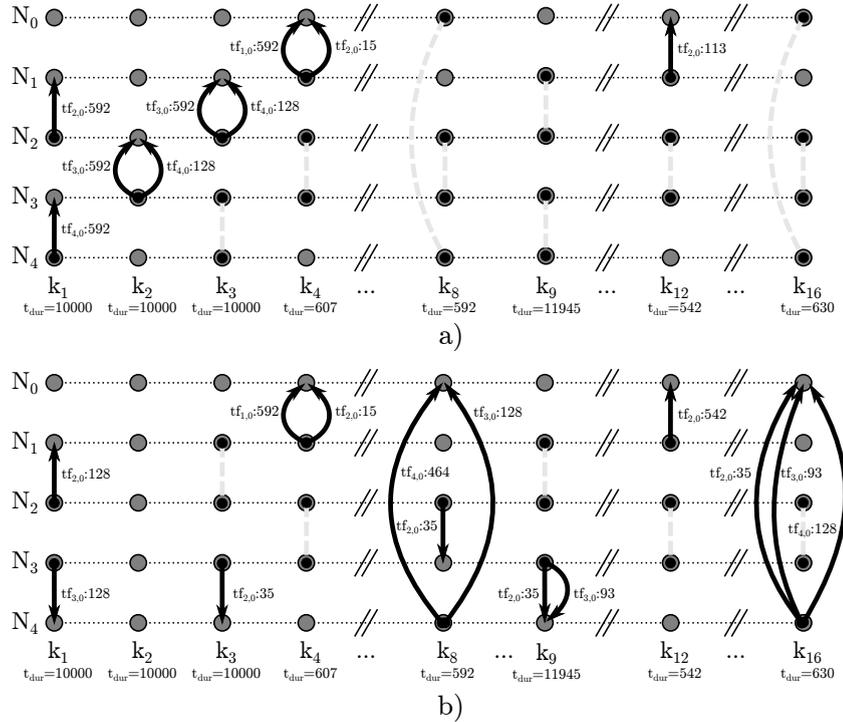


FIGURA 3.16: Flujos de tráfico para a) CGR y b) EGPA-CGR con $\rho = 1$

en su plan de contacto $(c_{k_4,1,0})$, ignorando que su capacidad puede verse ocupada por el tráfico de otros nodos. Solo el Nodo 1 puede notar la sobre-reserva de $c_{k_4,1,0}$ ya que recibe un flujo de tráfico de $tf_{2,0} : 592$ en k_1 y $tf_{3,0} : 592$ más $tf_{4,0} : 128$ en k_3 . Sin embargo, a pesar de que este nodo puede calcular rutas alternativas a través de $c_{k_8,4,0}$, no puede reenviar ningún dato debido a la política explicada anteriormente. Como resultado, el tráfico se bloquea hasta que se use un segundo contacto favorable en k_{12} . Se debe tener en cuenta además que muchos bundles se pierden debido a la limitada capacidad de los buffers de cada nodo, ya que no hay un mecanismo de custodia disponible.

LPA-CGR mejora el rendimiento de la tasa de entrega de bundles de CGR, ya que permite que los nodos intermedios 4, 3 y 2 también predigan y reaccionen ante la congestión por adelantado. En contraste con los flujos de CGR ilustrados en la Figura 3.16 a), los nodos intermedios 3 y 2 prevén capacidad de contactos futuros en las rutas, lo que les permite determinar que la capacidad de $c_{k_4,1,0}$ se encuentra completamente reservada antes que el tráfico llegue al Nodo 1. Como resultado, los bundles se reenvían a través de rutas alternativas que incluyen $c_{k_8,4,0}$, $c_{k_{12},1,0}$ y $c_{k_{16},4,0}$. Sin embargo, dado que LPA-CGR ignora el tráfico generado por otros nodos, se desperdicia una cantidad importante de recursos de comunicación en el rebote de datos hacia adelante y hacia atrás hasta que se agotan las capacidades de cada plan de contacto local de cada nodo. En esta

etapa, algunos bundles pueden terminar atrapados en nodos intermedios haciendo que la tasa de entrega de LPA-CGR sea de un 76,6% del rendimiento máximo. En general, CGR usa menos tiempos de contacto que LPA-CGR, pero el hecho de que este último entregue una cantidad mucho mayor de bundles a destino hace que el uso de tiempos de contacto normalizado sea siempre mayor para LPA-CGR como se ve en la Figura 3.15 b). Por otro lado, esto se produce a expensas de obtener un mayor tiempo de entrega para LPA-CGR cuando la carga de tráfico es $\rho = 1$, como se muestra en la Figura 3.15 c).

Los mejores resultados se obtienen claramente con GPA-CGR y EGPA-CGR ya que estos esquemas son capaces de evitar la congestión al proporcionar un plan de contactos específico para cada nodo. La Figura 3.16 b) evidencia que el hecho de eliminar el contacto $c_{k2,2,3}$ y hacer una reserva de capacidad adecuada permite que el Nodo 3 y el Nodo 4 tomen las decisiones convenientes de enrutar su tráfico a través de los contactos $c_{k8,4,0}$ y $c_{k16,4,0}$, mientras que el Nodo 1 y el Nodo 2 pueden usar libremente los contactos $c_{k4,1,0}$ y $c_{k12,1,0}$. Este tipo de procedimientos también permite obtener mejores resultados que CGR o LPA-CGR en términos de uso de tiempos de contacto normalizado. Se debe tener en cuenta que EGPA-CGR se configuró para mejorar los bundles entregados y el uso de los tiempos de contacto antes que el tiempo de entrega. Por lo tanto, siempre obtendrá mejores o iguales métricas que GPA-CGR en las Figuras 3.15 a) y b), y posiblemente una métrica peor en la Figura 3.15 c). Esto se debe al hecho de que cuando se prioriza el uso de tiempos de contacto normalizado en lugar del tiempo de entrega, se usarán menos contactos con el objetivo de reducir el uso de la red a expensas de entregar el tráfico más tarde. De esta forma, EGPA-CGR proporciona un esquema flexible y configurable para obtener planes de contactos personalizados que optimizan las métricas deseadas.

Otro efecto que se evidencia en los resultados es cómo la restricción de la capacidad del buffer afecta el rendimiento de la red a medida que aumenta la carga de tráfico. Notablemente, cuando $\rho \geq 0,7$ incluso los esquemas GPA-CGR y EGPA-CGR se ven afectados de manera que no pueden entregar todos los bundles generados en el período de la topología. Para hacer un análisis más profundo de la restricción del buffer establecemos la carga de tráfico en $\rho = 1$ y variamos la capacidad del buffer de cada nodo satelital entre 7.4 MBytes y 14.4 MBytes. La métrica de la tasa de entrega de bundles se obtiene utilizando los esquemas de enrutamiento descriptos. Además, el esquema PA-CGR (similar a

LPA-CGR pero que no tiene en cuenta las limitaciones de los buffers) también se incorpora al análisis con el objetivo de comparar la mejora obtenida con LPA-CGR. Como se muestra claramente en la Figura 3.17 GPA-CGR y EGPA-CGR pueden entregar todos los bundles generados cuando los almacenamientos intermedios son lo suficientemente grandes, mientras que la PA-CGR resulta mas afectado que LPA-CGR para el mismo escenario.

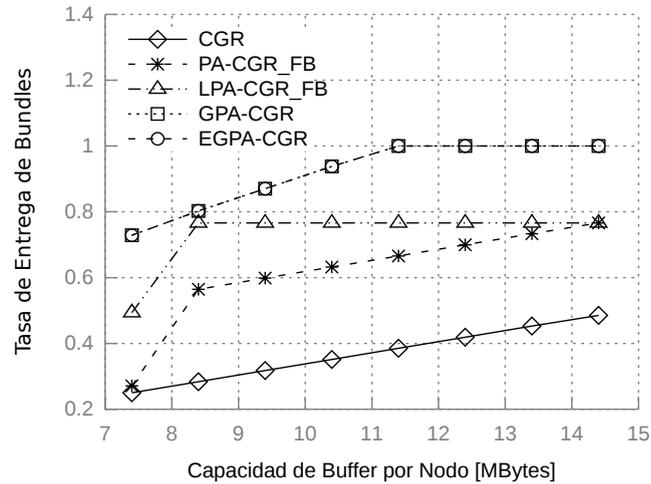


FIGURA 3.17: Resultados de variación de los tamaños de los buffers

3.9.4. Análisis de Algoritmos

Para resumir y comparar las diferentes estrategias descritas discutimos ahora los principales aspectos plasmados en la Tabla 3.3. Tanto CGR como LPA-CGR evitan la congestión considerando sólo el tráfico generado localmente. CGR sólo tiene en cuenta el primer contacto de la ruta, mientras que LPA-CGR amplía ese alcance a todos los contactos y capacidad de los buffers a lo largo de la ruta. Las mejoras en las métricas obtenidas mediante el uso de LPA-CGR se consiguen a expensas de un incremento en la complejidad computacional del algoritmo. Por otro lado, tanto GPA-CGR como EGPA-CGR obtienen mejores métricas porque evitan la congestión a nivel global al asignar un plan de contactos específico para cada nodo, basados en la suposición de que los generadores de tráfico pueden ser planificados de antemano por un nodo central. Otra ventaja de usar los esquemas globales está en el número de ejecuciones de los algoritmos. La parte intensiva de GPA-CGR y EGPA-CGR se aplica sólo una vez en un nodo central y luego el algoritmo asociado EB-CGR se ejecuta una vez por bundle sólo en el nodo fuente del tráfico guardando la ruta calculada en la cabecera del bundle.

Características \ Algoritmos	CGR	LPA-CGR	GPA-CGR/EGPA-CGR
Nivel de Congestión	Local (1° contacto)	Local (toda la ruta)	Global
Información Requerida	Topología	Topología	Topología + Trafico
Plan de Contactos	El mismo para todos los nodos	El mismo para todos los nodos	Uno diferente para cada nodo
Sobre Carga (<i>Overhead</i>)	Ninguna	Ninguna	Ruta calculada
Número de Ejecuciones	Una por bundle en cada nodo	Una por bundle en cada nodo	Una en el nodo central + una por bundle solo en el nodo origen

TABLA 3.3: Tabla de comparación de algoritmos

La Figura 3.18 muestra el número de ejecuciones realizadas con cada algoritmo en función de una carga de tráfico variable para el escenario descrito anteriormente en esta sección. El mejor rendimiento en términos de número de ejecuciones se consigue mediante los esquemas globales. CGR se considera tanto con como sin la opción de retorno hacia el nodo previo habilitada. LPA-CGR_FB se ejecuta menos veces que CGR_FB gracias a la capacidad mejorada de evitar la congestión, mientras que la aparente desventaja de LPA-CGR_FB con respecto a CGR se considera una compensación por la gran mejora obtenida por LPA-CGR_FB en la tasa de bundles entregados.

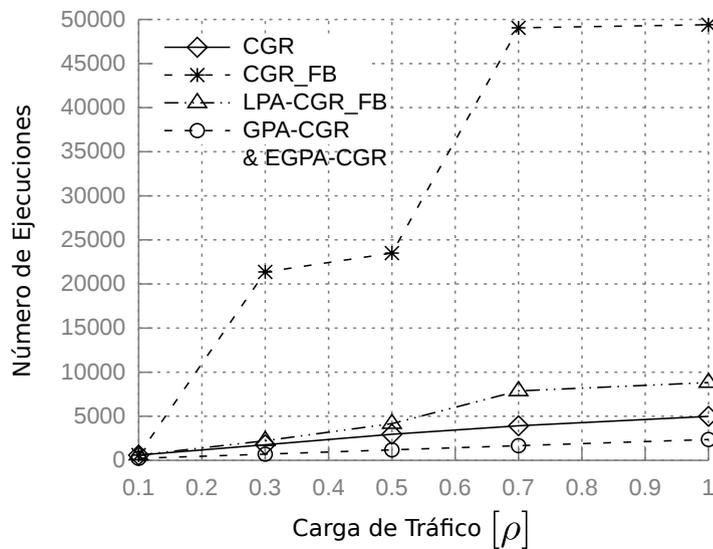


FIGURA 3.18: Ejecuciones de los algoritmos

3.10. Conclusiones Parciales

En este capítulo se introdujo y describió el problema de la congestión en las redes DTN. En particular, demostramos que este fenómeno no ha sido completamente considerado por los algoritmos actuales tales como CGR, a pesar de que puede reducir drásticamente el rendimiento global en la entrega de datos. Además, mostramos que la congestión puede deberse tanto a un tráfico local excesivo a lo largo de una ruta como a tráfico generado por otros nodos, lo que en ambos casos genera una sobre-reserva de los contactos o del almacenamiento limitado de los nodos.

Como resultado, contribuimos con estrategias novedosas para mitigar y evitar la congestión de manera proactiva. Por un lado, se introdujo LPA-CGR que extiende CGR al considerar la capacidad total de las rutas con el fin de mitigar la congestión provocada por tráfico local. En segundo lugar, propusimos GPA-CGR que aprovecha la previsibilidad del tráfico para evitar completamente la congestión reservando recursos de comunicación por adelantado. Además, extendimos este enfoque con un algoritmo evolutivo llamado *EGPA-CGR* que proporciona mayor flexibilidad en la definición de la función objetivo a optimizar.

Finalmente, al evaluar estas soluciones en una formación satelital de referencia, demostramos que LPA-CGR puede superar significativamente a CGR en términos de tasa de entrega de datos y uso de recursos con un incremento mínimo en la complejidad del algoritmo. Además, al asumir un tráfico predecible, se observó una mejora aún más importante para GPA-CGR y EGPA-CGR a expensas de la provisión de un plan de contactos específico por nodo. Finalmente, dejamos como trabajo adicional la investigación de las extensiones de GPA-CGR y EGPA-CGR para abordar las incertidumbres de previsibilidad y los fallos de comunicación del sistema.

Capítulo 4

Incertidumbre

4.1. Introducción

En este capítulo investigaremos los efectos que tienen las fallas en las redes DTN espaciales. Extenderemos trabajos anteriores [7, 100] al considerar la inyección de fallas no sólo aleatorias si no que también dirigidas hacia los puntos críticos de la red. Por un lado se analizarán cotas superiores de rendimiento al hacer que los algoritmos sean conscientes de las fallas futuras. Por otra parte, se realizará un estudio comparativo de esquemas tanto planificados como oportunistas para cuantificar los beneficios obtenidos al utilizar la replicación de paquetes. Se espera que las conclusiones obtenidas sirvan como base para mejorar el diseño de planes de contactos y los algoritmos de enrutamiento para ser más robustos y resilientes ante la presencia de fallas.

4.2. Fallas en Sistemas Espaciales

Existen varias causas potenciales de fallas permanentes en el entorno espacial, tales como colisiones de objetos espaciales, rotura de componentes debido a vibraciones de lanzamiento, desgasificación de material, entre otras. Otras fallas resultan de carácter transitorio y en general son provocadas por rangos de temperatura muy variables (falta de disipación de calor por convección), interferencias inesperadas por radiofrecuencia y efectos de radiación. En su conjunto, estos efectos resultan de gran importancia en misiones espaciales, donde se requieren sistemas que puedan operar de forma confiable por largos períodos de tiempo con poco o ningún mantenimiento.

Dentro de las fallas transitorias, se destacan aquellas provocadas por los efectos de la radiación. En particular, cualquier circuito que contenga elementos de memoria (registros, flip flops, etc) puede en cualquier momento sufrir la modificación de uno o varios bits de información debido a partículas ionizantes (SEU, single event upsets) [73]. La tolerancia a estas fallas transitorias por radiación han recibido gran atención de la industria y academia debido al aumento de la sensibilidad asociado al mayor grado de integración de los circuitos electrónicos [101]. Como consecuencia, los sistemas embebidos modernos pueden ser susceptibles a partículas de baja energía incluyendo aquellas presentes en la atmósfera baja de la Tierra. Tradicionalmente, las misiones espaciales son diseñadas para tolerar estas fallas transitorias detectando el comportamiento erróneo y recuperando el sistema por medio de un reinicio.

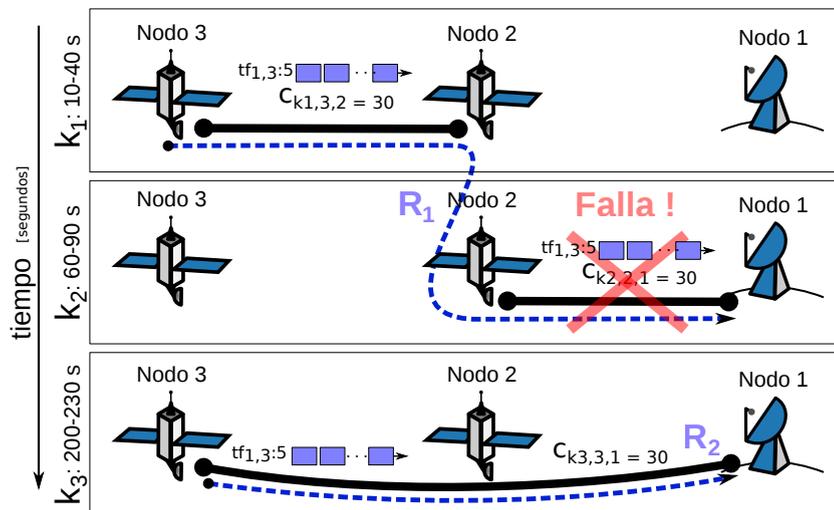


FIGURA 4.1: Enrutamiento en escenario planificado con fallas

En este capítulo estudiamos el impacto que tienen la fallas transitorias sobre redes DTN planificadas. Para modelar estos eventos no planificados, consideramos un modelo de incertidumbre simple en el que se puede suprimir un contacto completo (desde el inicio hasta el final del contacto) del plan de contactos de dos maneras diferentes: aleatoria (caso realista) y dirigida (caso pesimista).

En el primer caso, un contacto se puede eliminar de manera aleatoria e independiente de otros contactos con una probabilidad determinada. Como consecuencia de estas incertidumbres, el plan de contactos asumido por el algoritmo de enrutamiento puede no representar la topología real de la red. Esto significa que el cálculo de la tabla de rutas óptimas podría dejar de ser válido. Tomando como ejemplo la red de la Figura 4.1, podemos observar que el Nodo 3 tiene tráfico destinado al Nodo 1, y calcula las rutas $R1$ y $R2$. Si el Nodo 3 escoge la mejor ruta ($R1$) para llegar al Nodo 1, y el modelo de incertidumbre inyecta una falla en el contacto $C_{k2,2,1}$, el tráfico no será capaz de llegar a su destino.

En el segundo caso, las inyecciones dirigidas de fallas buscarán afectar los contactos más importantes desde el punto de vista de la conexión de los nodos de la red. Específicamente, asumimos que los contactos más importantes van a ser aquellos que posean una mayor medida de centralidad (*edge betweenness centrality* [102]), es decir, aquellos por los cuales pasan la mayor cantidad de rutas más cortas. Por ejemplo, en la Figura 4.2,

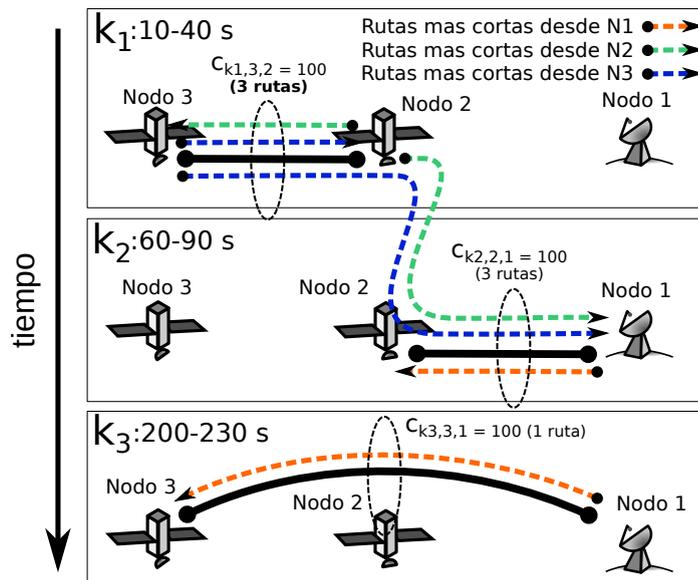


FIGURA 4.2: Cantidad de rutas por contacto

se ilustran las rutas más cortas del caso de ejemplo anterior, donde se observa que los contactos $c_{k1,3,2}$ y $c_{k2,2,1}$ son utilizados por 3 de las rutas más cortas entre los nodos de la red, por lo que serían los primeros contactos considerados para la inyección dirigida de fallas.

A continuación realizaremos dos estudios por medio de simulación que nos permitirán, por un lado, cuantificar la degradación del rendimiento de CGR en escenarios con fallas, y por otro, comparar distintos esquemas de enrutamiento que utilicen la replicación de paquetes como medio para hacer frente a la incertidumbre.

4.3. Estudio de Cotas de Rendimiento

4.3.1. Extensión de DtnSim

Para este estudio en particular, DtnSim ha sido extendido con dos módulos: uno que permite generar topologías aleatorias con distintas densidades de contactos entre los nodos, y otro que permite especificar cuales son los contactos que fallarán. De esta forma, contamos con una herramienta que nos permite analizar el desempeño de distintas redes DTN bajo la influencia de distintos patrones de fallas.

4.3.2. Escenarios Simulados

Con el fin de estudiar el impacto de fallas transitorias se generan modelos de redes simples que si bien no son obtenidas de un patrón orbital real, proporcionan una primera intuición sobre el fenómeno en cuestión. En particular, se realizan simulaciones de topologías de 100 segundos, fragmentadas en 10 estados de 10s y con 4 nodos. Los contactos son generados aleatoriamente con distintos grados de conexión entre los nodos. El grado de conexión se asocia directamente con la densidad de contactos δ que toma valores $\delta = 0,1$, $\delta = 0,5$ y $\delta = 1,0$. De esta forma, una red con $\delta = 1,0$ está totalmente conectada con contactos permanentes desde el comienzo de cada estado hasta el final del mismo, para todos los estados. Cada nodo genera un tráfico de 100 bundles de 100 bytes por bundle hacia cada uno de los nodos de la red. De esta forma, el tráfico generado por el nodo 1 es de 300 bundles y el tráfico total generado por todos los nodos de la red es de 1200 bundles. Para evitar la superposición con otros efectos como la congestión, la

capacidad de los contactos se escoge de forma tal que todo el tráfico generado en la red pueda ser transmitido por cualquier contacto. Finalmente, sobre estos escenarios DTN, se inyectan fallas de forma aleatoria y dirigida a los contactos de mayor relevancia en el flujo del tráfico. Estas inyecciones se presentan en proporción a la cantidad total posible de fallas, es decir, una proporción de fallas de 1,0 se asocia con una situación en que todos los contactos de la topología tienen fallas. Como método de enrutamiento para encaminar el tráfico hacia los respectivos destinos, cada nodo ejecuta el algoritmo CGR. Con el objetivo de comparar el rendimiento de CGR ante la presencia de fallas, se analizan dos esquemas: a) *Faults Aware*, donde cada nodo es consciente de los contactos con fallas y es capaz de ignorarlos a la hora de enrutar y encolar los bundles; y b) *Non Faults Aware*, donde los nodos no son conscientes de las fallas. En este último caso, los nodos tomarán ciertas decisiones de enrutamiento y encolamiento por contactos con fallas. En el momento de establecer esos contactos, los nodos serán capaces de detectar las fallas y deberán reaccionar a las mismas realizando el re-enrutamiento de los bundles encolados en dichos contactos.

4.3.3. Análisis de Resultados

El rendimiento final de la red se observa a partir de las métricas de bundles recibidos correctamente en destino y bundles re-ruteados (aquellos que ante la detección de una falla, deben ser sometidos nuevamente al cálculo de una ruta alternativa). En efecto, cuando no hay fallas, se espera que la cantidad de bundles recibidos sea máxima y la cantidad de bundles re-ruteados sea mínima. Como métricas adicionales se analizan la disponibilidad de rutas y el número de pares de nodos conectados por al menos una ruta, dada una determinada cantidad de contactos con fallas.

En las Figuras 4.3 y 4.4 se observa la variación de la disponibilidad de rutas y el número de pares conectados a medida que aumenta la proporción de contactos con fallas en los dos casos mencionados: I) inyección aleatoria y II) inyección dirigida. Dentro de cada caso se obtuvieron 3 curvas correspondientes a densidades de contactos $\delta = 0,1$, $\delta = 0,5$ y $\delta = 1,0$. Aquí puede apreciarse como disminuye tanto la disponibilidad de rutas como la cantidad de nodos conectados por al menos una ruta a medida que hay una mayor cantidad de contactos con fallas. Además, resulta clara la diferencia entre el caso de inyección aleatoria con respecto al caso de inyección dirigida. En este último, la

conectividad entre los nodos DTN comienza a ser notablemente menor cuando hay más de un 40% de contactos con fallas.

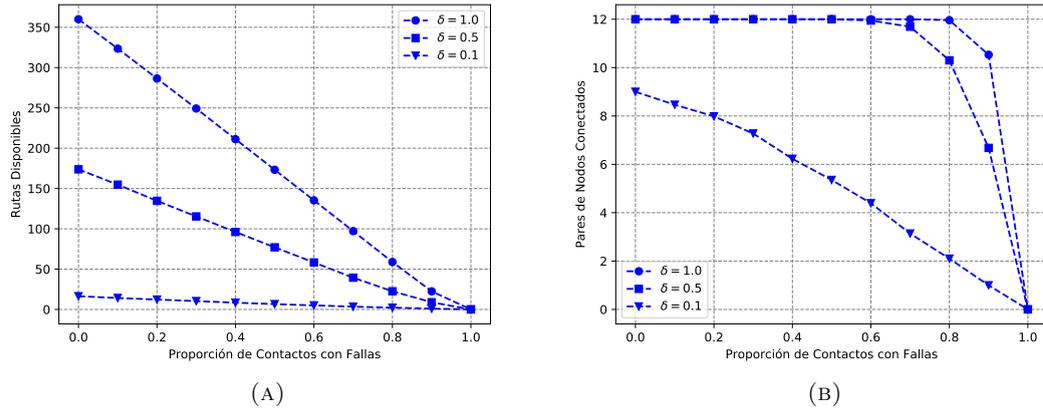


FIGURA 4.3: Disponibilidad de rutas con inyección de fallas de forma aleatoria

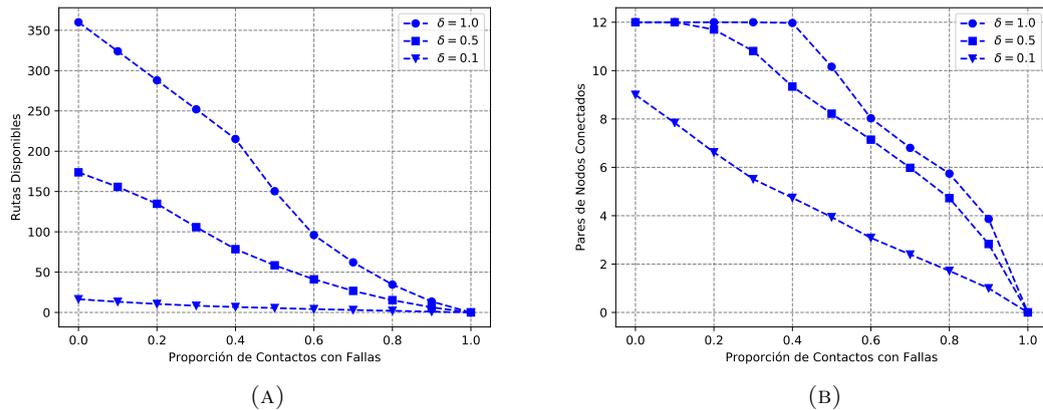


FIGURA 4.4: Disponibilidad de rutas con inyección de fallas de forma dirigida

Efectivamente, la disponibilidad de rutas tiene un impacto directo en el flujo final del tráfico y por ende en el rendimiento final del sistema. El rendimiento de la red se ve claramente afectado de manera acorde a las métricas anteriores como se ilustra en las Figuras 4.5 y 4.6. Aquí se observan los bundles recibidos y re-ruteados teniendo en cuenta los dos modos de inyección de fallas descritos y los dos modos de consciencia respecto a las fallas de CGR.

Si tomamos por ejemplo la Figura 4.5 A), observamos que cuando la proporción de contactos con fallas es 0, se entregan todos los bundles generados (1200) a sus respectivos destinos con $\delta = 0,5$ y $\delta = 1,0$. No pasa lo mismo cuando $\delta = 0,1$, donde, aún sin la presencia de fallas, se entregan 900 bundles. Como se observa en la Figura 4.3 B), esto

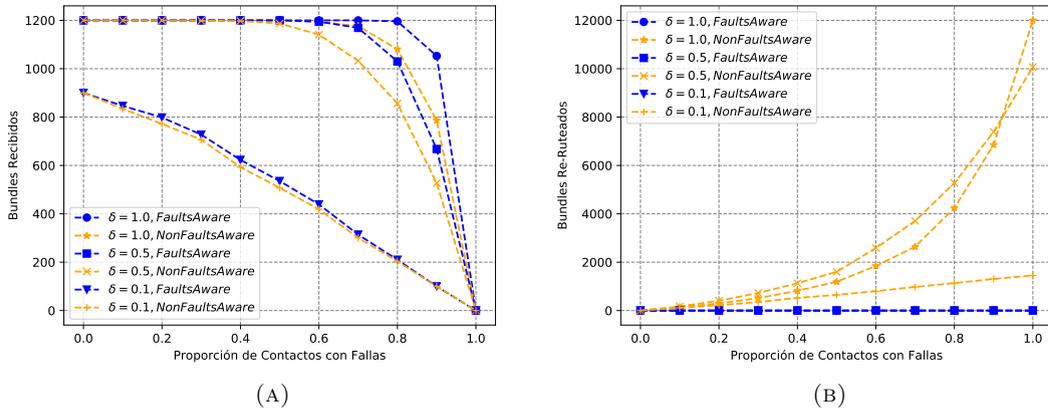


FIGURA 4.5: Rendimiento con inyección de fallas de forma aleatoria

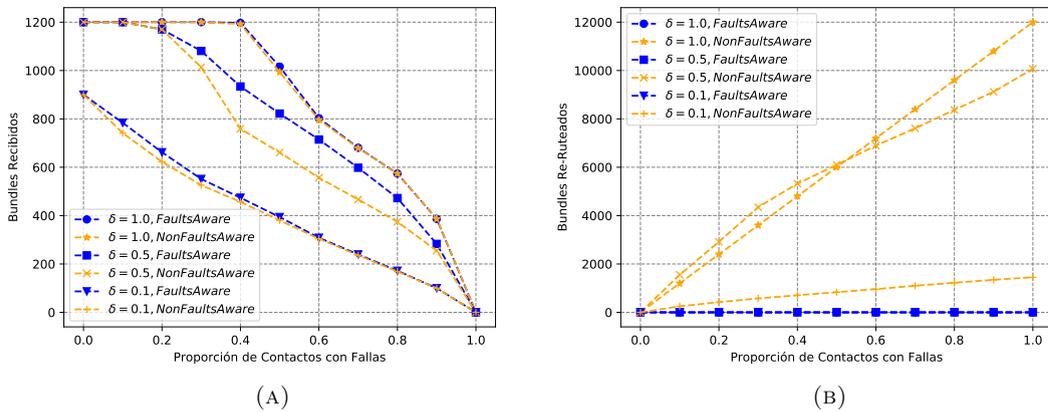


FIGURA 4.6: Rendimiento con inyección de fallas de forma dirigida

se debe a que la cantidad de pares de nodos conectados en ese caso es de 9. Es decir, cuando hay pocos contactos puede darse la situación en que haya pares de nodos entre los cuales no haya ruta posible para enrutar el tráfico generado. Observamos también en este mismo caso que, dada una cantidad de fallas, la cantidad de bundles recibidos siempre es menor cuando el algoritmo CGR no es consciente de las fallas debido a la toma de decisiones de enrutamiento que no pueden concretarse según lo previsto. En efecto, resulta que el modo consciente de fallas aporta el rendimiento óptimo (dadas las condiciones del algoritmo) ante una disminución de recursos de red (contactos) debido a fallas.

Por otro lado, analizando la Figura 4.5 B), vemos en primer lugar que los re-ruteos producidos en el modo consciente a fallas es siempre 0. Esto se debe a que en este modo, el algoritmo CGR nunca va a enrutar tráfico por donde conoce que hay fallas.

En el peor de los casos, conservará el tráfico en su memoria por falta de rutas que puedan llevar el tráfico a destino. Por otra parte, un fenómeno no tan intuitivo se da en el modo no consciente a fallas con respecto al cruce entre las curvas con densidades $\delta = 0,5$ y $\delta = 1,0$. Esto se debe a que cuando la proporción de contactos con fallas es baja, la probabilidad de que CGR elija re-rutear bundles en contactos que también tienen fallas es mas alta cuando $\delta = 0,5$ que cuando $\delta = 1,0$. Visto de otra forma, cuando la densidad es $\delta = 1,0$, es mas probable que un re-ruteo hecho por CGR utilice contactos que no tienen fallas, disminuyendo así el número de re-ruteos. Por otro lado, cuando la proporción de contactos con fallas es mayor, comienza a dominar otro efecto: la cantidad de rutas. Cuando todos los contactos fallan, no hay rutas disponibles y el tráfico entregado es 0. Sin embargo, en el modo no consciente de fallas y cuando $\delta = 1,0$, CGR asume que tiene una mayor cantidad de rutas respecto al caso de $\delta = 0,5$ por lo que intenta re-rutear el tráfico por rutas que en realidad no existen debido a las fallas. Por último, la Figura 4.6 ilustra los mismos efectos antes mencionados aunque con un mayor grado de degradamiento de la red debido a que las fallas son dirigidas hacia aquellos contactos de mayor relevancia para el flujo de datos (caso pesimista).

4.4. Estudio de Replicación

4.4.1. Esquemas de Enrutamiento

Como hemos mostrado a lo largo de esta tesis, la utilización de información planificada es un modelo adecuado para utilizar en redes satelitales. Sin embargo, también se han considerado soluciones oportunísticas con el objetivo de minimizar el esfuerzo de cálculo y evitar la distribución del plan de contactos, como así también los posibles problemas de sincronización. Mientras que los algoritmos de enrutamiento en las redes DTN planificadas determinan con precisión el siguiente mejor vecino (o salto) en la ruta para un único bundle, el enrutamiento en las DTN oportunistas intenta crear copias con el fin de maximizar la probabilidad de entrega. Aquí consideraremos los siguientes tipos de esquemas:

4.4.1.1. Oportunísticos

Cuando no se pueden hacer suposiciones sobre la futura conectividad de la red, se han considerado soluciones triviales basadas en la duplicación extensiva de paquetes. Por ejemplo, los esquemas basados en inundaciones, como el enrutamiento *Epidémico* [36], envían una copia del paquete a cada nodo vecino que entra en contacto. Sólo se excluyen los que ya tienen una copia del paquete. Como resultado, la probabilidad de entregar el paquete a su destino se maximiza a expensas de grandes costos de transmisión.

Por otro lado, esquemas menos agresivos como *Spray and Wait* [37] han mostrado métricas de rendimiento similares a las de las inundaciones, pero con menos sobrecarga de la red. En *Spray and Wait*, el nodo origen (es decir, la fuente de tráfico) distribuye un número limitado de copias a los primeros vecinos contactados y luego espera hasta que una de esas copias llegue al destino. De hecho, el número óptimo de copias depende de las características de la topología de la red.

4.4.1.2. Planificados (tiempo de entrega)

Cuando la topología puede ser determinada de antemano, esquemas como CGR pueden aprovechar el conocimiento previo del plan de contacto para calcular una tabla de rutas. Los nodos pueden luego utilizar la tabla de rutas para encolar y enviar bundles de forma eficiente a través de contactos que siguen rutas óptimas hacia el destino. En general, las soluciones CGR existentes tienen como objetivo minimizar una métrica de tiempo de entrega (*delivery time*), reduciendo así el tiempo que los datos pasan en tránsito en la red. Para un tiempo de entrega igual, se elige la ruta con menor número de saltos. Nos referimos a esta declaración clásica de CGR como *CGR-DelTime*.

4.4.1.3. Planificados (número de saltos)

La optimización estricta del tiempo de entrega puede implicar la utilización de varios contactos y, por lo tanto, de varias transmisiones que terminan reduciendo la eficiencia energética. Además, en el contexto de incertidumbres, mientras mayor sea el número de saltos, mayor será la probabilidad de que un bundle encuentre un nodo o contacto con fallas en su trayectoria. De hecho, cuando todos los contactos tienen la misma

probabilidad de fallar, un trayecto con el número mínimo de saltos es el más confiable, aunque no necesariamente el más rápido. Con el fin de mejorar la entrega de paquetes en escenarios predecibles bajo planes de contacto con incertidumbre, proponemos una versión modificada de CGR llamada *CGR-Hops* en la que la métrica de número de saltos tiene una mayor relevancia que el tiempo de entrega. Cuando sea posible, *CGR-Hops* se comportará como un enrutamiento directo donde la fuente de tráfico entrega el bundle directamente al destino.

4.4.1.4. Planificados (2 copias)

Se puede utilizar la replicación controlada de bundles para garantizar la entrega de datos. De hecho, la formulación original de CGR hace varias copias de los llamados bundles críticos y los envía a todos los posibles vecinos o a un conjunto de ellos. En este trabajo, presentamos una formulación de CGR llamada *CGR-2Copies* que utiliza replicación para superar la incertidumbre en la planificación. En *CGR-2Copies*, el nodo fuente de tráfico hace dos copias del bundle generado, pero se eligen rutas diferentes para cada una de ellas en función de métricas distintas. Por un lado, la primera copia se envía por la ruta con el mejor tiempo de entrega, igual que en el CGR tradicional. Por otro lado, también se evalúa una segunda ruta con el número mínimo de saltos. Si la segunda ruta tiene un primer vecino diferente, la segunda copia del bundle se envía a través de esta ruta alternativa. Como resultado, *CGR-2Copies* enviará un bundle a través de la ruta más rápida (pero menos confiable) y otra a través de la más robusta (y más lenta). Sin embargo, dado que las propiedades, los parámetros y el estado de los nodos satelitales se conocen de antemano, la incertidumbre también puede estimarse y aprovecharse mediante el algoritmo de enrutamiento. Como resultado, evaluamos una segunda variante en la que el algoritmo utiliza los parámetros de incertidumbre para ahorrar recursos de transmisión y almacenamiento. En *CGR-2Copies-0.5*, el bundle siempre se envía a través de la ruta más confiable (número mínimo de saltos), pero sólo a través de la ruta más rápida (menor tiempo de entrega) cuando su probabilidad de llegar a destino a través de esa ruta es superior a 0.5.

4.4.2. Métricas de Rendimiento

El rendimiento de cada esquema de enrutamiento se evaluará teniendo en cuenta las siguientes métricas:

- *Tasa de Entrega*: número de bundles entregados con éxito sobre el número de bundles generados, excluidas las copias.
- *Tiempo de Entrega Medio por Bundle*: retraso medio en segundos por bundle entregado con éxito.
- *Eficiencia Energética*: número de bundles entregados con éxito sobre el número total de transmisiones.
- *Número Medio de Bundles en SDR*: número de bundles en el buffer de todos los nodos promediados en el tiempo (ocupación media del buffer).
- *Número Medio de Saltos por Bundle*: número medio de transmisiones o contactos utilizados por bundle entregado con éxito

4.4.3. Escenarios Simulados

Utilizando nuevamente la herramienta de simulación DtnSim, evaluamos dos tipos de escenarios: uno con redes aleatorias para ganar intuición sobre los efectos y las relaciones de compromiso involucradas, y otro con una constelación realista de satélites. Detallamos ambos escenarios a continuación.

4.4.3.1. Redes Aleatorias

Generamos 10 topologías aleatorias de 100 segundos fragmentadas en 10 estados de 10 segundos con 8 nodos. La conectividad entre nodos se basa en el parámetro de densidad de contactos δ que puede tomar valores entre 0,0 y 1,0. Una red con $\delta = 1,0$ está completamente (y continuamente) conectada con contactos presentes en todos los estados, mientras que $\delta = 0$ implica que no existen contactos en absoluto. Para los casos que se estudian en este análisis, fijamos $\delta = 0,2$. Aunque no se incluyen los resultados con otros valores de δ , el único efecto de aumentarlo es que se necesitan valores más altos de

incertidumbre para observar las mismas reducciones de rendimiento. Además, cada nodo genera 10 bundles para cada uno de los otros nodos de la red en el momento $t = 0$. Por un lado, proponemos un caso de estudio I) con *baja congestión* en el que todo el tráfico generado puede ser transferido utilizando un único contacto (bundles de 1785 Bytes). Por otro lado, analizamos un caso de estudio II) con *alta congestión* donde los nodos generan 20 veces más tráfico (bundles de 35700 Bytes). La velocidad de transmisión de datos configurada para todos los contactos es de 800 Kbps. Además, aunque no imponemos restricciones de buffer en este escenario, medimos la utilización del almacenamiento local (*Spacecraft Data Recorder, SDR*) en cada satélite.

4.4.3.2. Constelación Walker

Un tercer caso de estudio más realista III) se basa en un patrón de constelación *Walker* en órbita terrestre baja que fué también estudiado en [7]. En particular, establecemos 25 objetivos terrestres aislados en todo el mundo (IDs 7 a 31) que necesitan intercambiar datos con una única estación terrestre ubicada en Argentina (ID 1) utilizando 16 satélites como mulas de datos (IDs 32 a 47). Los satélites están equipados con enlaces ISL, lo que implica que los contactos también son posibles en órbita cuando el alcance lo permite (véase [7] para mayores detalles). La topología resultante se propaga durante 3 días y se ilustra en la Figura 4.7. Cada objetivo terrestre genera 10 bundles destinados a la estación terrena y la estación terrena genera 10 bundles destinados a cada objetivo terrestre. Los tamaños de los bundles se establecen en 78 Bytes y las velocidades de datos se configuraron en 100 Kbps para todos los contactos.

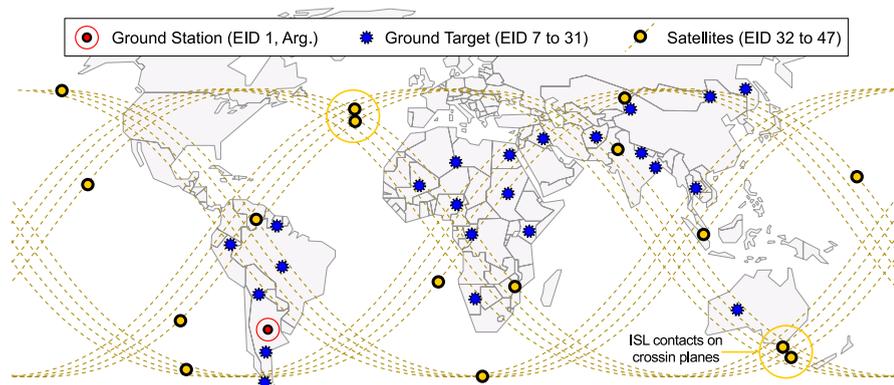


FIGURA 4.7: Constelación Walker

En particular, DtnSim fue configurado para evaluar el *CGR-DelTime* original, *CGR-Hops*, las dos versiones de CGR basadas en replicación (*CGR-2Copies* y *CGR-2Copies-0.5*), *Spray-and-wait-2Copies*, *Spray-and-wait-4Copies* y el enrutamiento *Epidémico*. En todos los casos, la proporción de fallas varía entre 0,0 y 0,8. Una proporción de 0,0 es la situación en la que todos los contactos se producen exactamente como estaban planificados, mientras que una proporción de 1,0 sería la situación en la que todos los contactos fallan y no es posible ninguna transmisión de datos entre los nodos.

4.4.4. Análisis de Resultados

4.4.4.1. Caso de Estudio I, Redes Aleatorias, Baja Congestión

La Figura 4.8 muestra que cuando la proporción de fallas es 0,0 todos los esquemas de enrutamiento obtienen casi la misma *tasa de entrega* de bundles, pero cuando la proporción de fallas aumenta, la *tasa de entrega* varía. *CGR-DelTime* supera a *CGR-Hops* porque la menor cantidad de saltos favorece la selección de rutas más lentas (posteriores), lo que reduce las oportunidades de re-enrutar los bundles si se encuentran fallas en contactos posteriores. Como era de esperar, en presencia de una alta proporción de fallas, los esquemas oportunistas tienden a superar a *CGR-DelTime* y *CGR-Hops* (soluciones sin copia). Por ejemplo, a partir de una proporción de fallas de 0,45, *Spray-and-wait-2Copies* se convierte en una mejor alternativa que *CGR-DelTime*. De hecho, cuanto mayor sea el número de copias, más significativa será la tolerancia a las fallas en escenarios sin congestión grave (es decir, en general, hay suficientes recursos de comunicación para utilizar). Curiosamente, considerando sólo dos copias, *CGR-2Copies* muestra una buena relación de entrega en el rango de fallas entre 0,1 y 0,7. En particular, *CGR-2Copies* entrega más bundles que *Spray-and-wait-2Copies* debido al conocimiento adicional de la topología futura, lo cual aumenta las posibilidades de llegar al destino. Como se esperaba, *CGR-2Copies-0.5* se comporta como *CGR-2Copies* hasta una proporción de fallas de 0,3, y luego sigue con las métricas más conservadoras de *CGR-Hops*.

El *tiempo de entrega medio por bundle* y el *número medio de saltos por bundle* muestran que *CGR-DelTime* logra menores retrasos con más transmisiones por bundle, *CGR-Hops* utiliza menos transmisiones pero con mayores retrasos, y los esquemas oportunistas tienen valores intermedios entre ambas soluciones de enrutamiento.

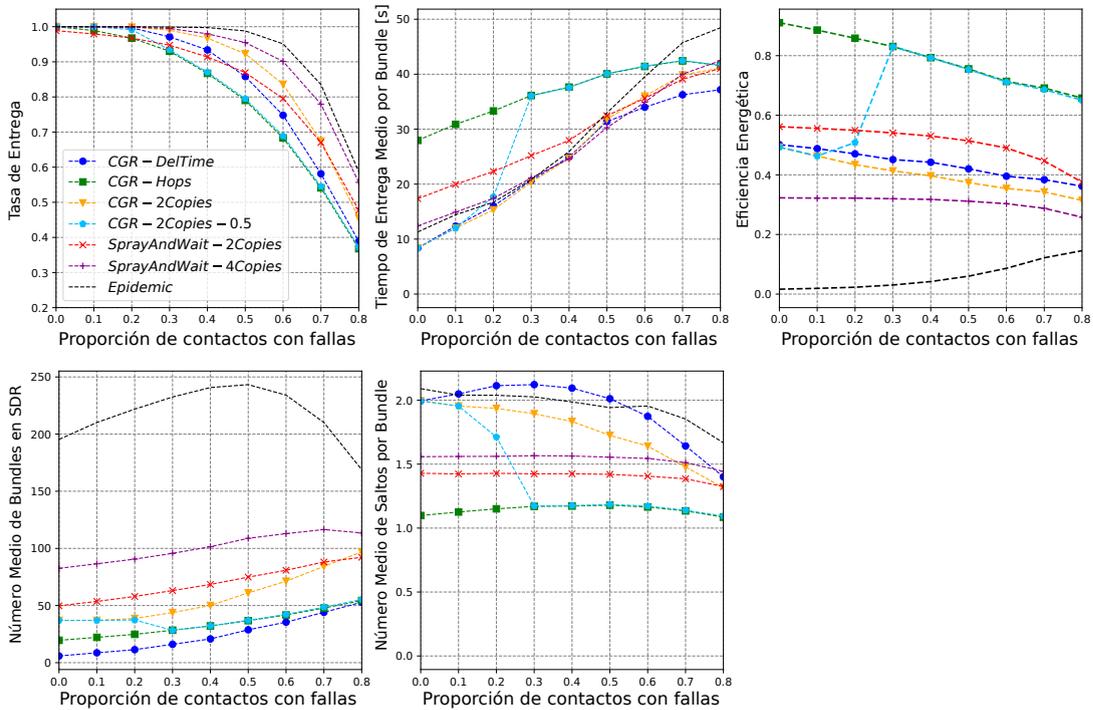


FIGURA 4.8: Resultados de caso de estudio I) redes aleatorias con baja congestión

Sin congestión, *la tasa de entrega* puede aumentarse eligiendo un esquema oportunista con un número adecuado de copias. Sin embargo, este incremento se produce a expensas de la utilización de más recursos en términos de número de bundles transmitidos y ocupación media del buffer. Estos efectos pueden observarse en la *eficiencia energética* y en el *número medio de bundles en SDR*, donde los esquemas *Spray-and-wait-nCopies* y *Epidémico* utilizan más recursos que las variantes *CGR* sin copia. También debe notarse que hay un incremento en la ocupación del buffer por parte de *CGR-2Copias* cuando las proporciones de fallas son altas. Esto se debe a que mientras que los esquemas oportunistas sólo crean copias al principio de los nuevos contactos, los esquemas *CGR-2Copias* las crean y encolan en el almacenamiento hasta que se inicia el contacto esperado. Sin embargo, si todos los contactos están deshabilitados debido a las fallas, las copias se mantendrán en el almacenamiento, aumentando de esta manera la ocupación media del buffer.

4.4.4.2. Caso de Estudio II, Redes Aleatorias, Alta Congestión

De la Figura 4.9 se puede observar que a diferencia del caso de baja congestión, *CGR-Hops* obtiene una mejor *tasa de entrega* que *CGR-DelTime*. Para explicar este comportamiento notamos que cuando los bundles utilizan rutas que llegan antes a destino, estas requieren en general de un mayor número de contactos (y por lo tanto de transmisiones). Pero cuanto mayor es el número de transmisiones, más diferentes son las visiones topológicas locales de cada nodo en términos de capacidad y, por lo tanto, más probable es que se produzca o empeore una congestión. Por otro lado, *CGR-2Copies* supera a *Spray-and-wait-2Copies* para valores de fallas inferiores a 0,4. Las mayores proporciones de fallas y el consiguiente problema de congestión tornan ineficiente la ventaja de utilizar información sobre la topología planificada. Sin embargo, la *tasa de entrega* del enrutamiento *Epidémico* se reduce drásticamente, ya que la creación de una gran cantidad de copias contribuye significativamente a empeorar la congestión. En cuanto al *tiempo de entrega medio por bundle*, las soluciones oportunistas evidencian las mejores métricas.

Con respecto a las métricas de *eficiencia energética* y *número medio de bundles en SDR*, *CGR-Hops* (y *CGR-2Copies-0.5* después de una proporción de fallas de 0,3) muestra

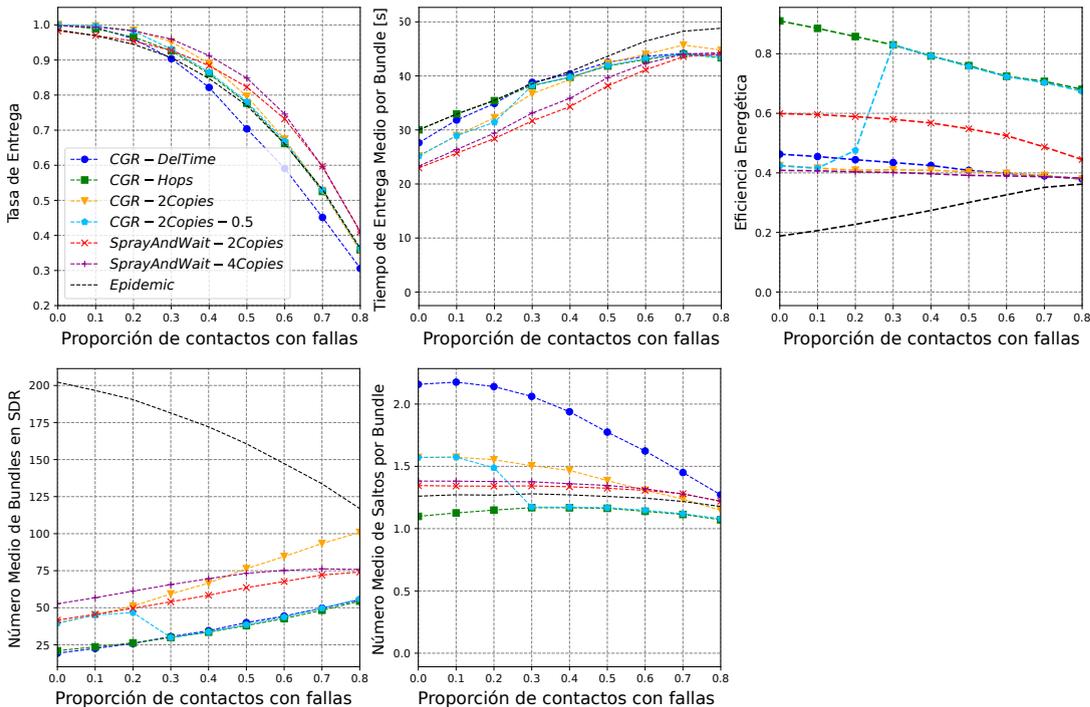


FIGURA 4.9: Resultados de caso de estudio II) redes aleatorias con alta congestión

los mejores valores seguido luego por *Spray-and-wait-2Copies*. Debido a la congestión, *CGR-DelTime* tiene que volver a calcular y reenviar su único bundle varias veces, lo que resulta en una baja métrica de eficiencia energética.

4.4.4.3. Caso de Estudio III, Constelación Walker

La Figura 4.10 muestra los resultados de un escenario más realista basado en la constelación Walker. Aquí, las curvas de todas las métricas son similares al caso de las redes aleatorias con baja congestión, aunque es notable el hecho de que en términos de *tasa de entrega*, *CGR-DelTime* es mejor que el esquema con copias *Spray-and-wait-2Copies*. Además, los esquemas *CGR-2Copies* muestran mejores métricas de *tasa de entrega* y *tiempo de entrega medio por bundle* a expensas de una menor eficiencia energética. Una vez más, el enrutamiento *Epidémico* muestra la mejor *tasa de entrega* a cambio de una eficiencia energética muy baja.

De hecho, el enrutamiento *Epidémico* es mucho más exigente en términos de cantidad de transmisiones y ocupación del buffer en este caso realista. Por otro lado *Spray-and-wait-2Copies* muestra métricas de eficiencia comparables con *CGR-Hops*, ya que entrega

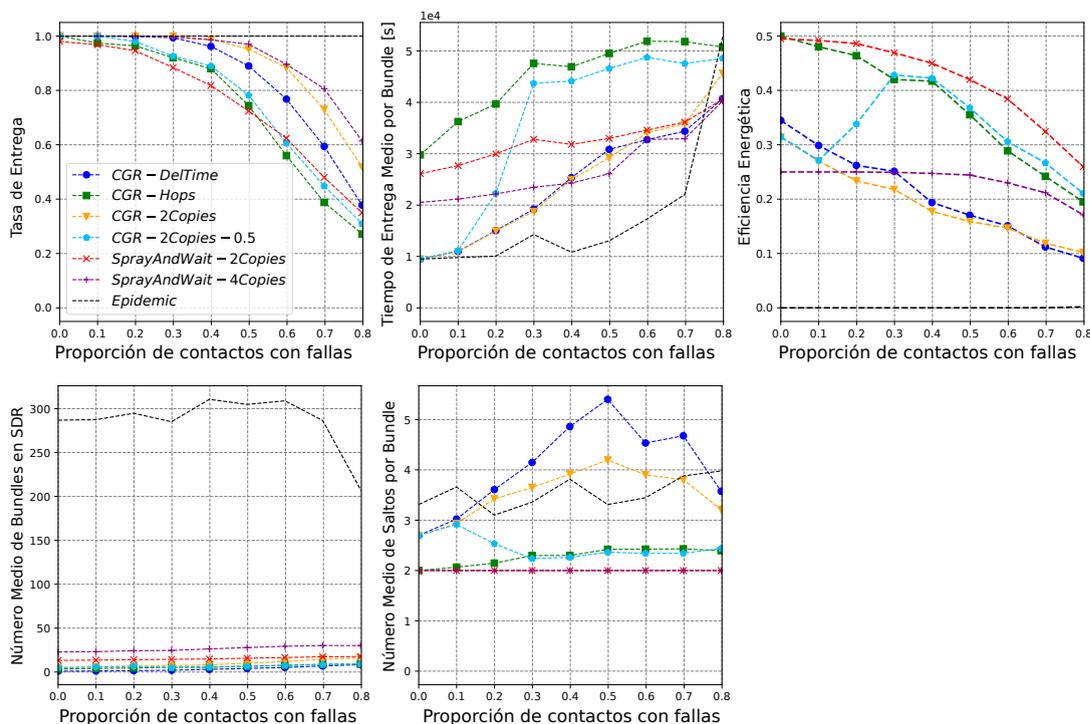


FIGURA 4.10: Resultados de caso de estudio III) constelación Walker

bundles con un promedio de dos saltos. La eficiencia energética de *CGR-DelTime* se degrada gravemente cuando las proporciones de fallas son altas.

A partir de los resultados descritos en este caso de estudio realista, hemos clasificado los diferentes esquemas de enrutamiento por orden de preferencia en función de la proporción de fallas y de los recursos disponibles en la Tabla 4.1. En el caso de recursos abundantes, se prioriza la *tasa de entrega* frente al *tiempo de entrega medio por bundle*, y en el caso de recursos limitados, se da prioridad a la *eficiencia energética* con respecto al *número medio de bundles en SDR*.

Proporción de Fallas	Recursos Abundantes	Recursos Limitados
Sin Fallas (0,0)	1) <i>CGR-DelTime</i>	1) <i>CGR-Hops</i>
Baja (0.0 to 0.2)	1) <i>CGR-2Copies</i> 2) <i>CGR-DelTime</i>	1) <i>Spray-and-wait-2Copies</i> 2) <i>CGR-Hops</i>
Media (0.2 to 0.4)	1) <i>Epidemico</i> 2) <i>CGR-2Copies</i>	1) <i>Spray-and-wait-2Copies</i> 2) <i>CGR-Hops</i>
Alta (0.4 to 0.6)	1) <i>Epidemico</i> 2) <i>Spray-and-wait-4Copies</i>	1) <i>Spray-and-wait-2Copies</i> 2) <i>CGR-2Copies-0.5</i>

TABLA 4.1: Preferencia de esquemas de enrutamiento en constelación Walker

4.5. Discusión

4.5.0.4. Modelo de Fallas

En este trabajo, hemos asumido un modelo de fallas simple basado en la supresión de contactos de la topología real. Aunque esto puede imitar las fallas transitorias (por ejemplo, el restablecimiento de la computadora de abordo) en los nodos DTN, otras incertidumbres también pueden traducirse en una reducción o ampliación del tiempo de contacto. Además, la velocidad de transmisión de datos esperada puede estar sujeta a variaciones debidas a imprecisiones del apuntamiento de la antena o a la orientación del satélite. Posteriores análisis de los esquemas de enrutamiento bajo incertidumbre considerarán modelos más realistas.

4.5.0.5. Re-cálculo de Ruta

Un asunto no trivial es cuándo declarar que un contacto planificado no ha ocurrido como se esperaba. En el análisis presentado, el recálculo de ruta y el reenvío de bundles se activan cuando se alcanza el tiempo final de un contacto no establecido de forma efectiva. Sin embargo, se deben explorar y analizar otras estrategias de reacción. Por ejemplo, la reacción ante las fallas puede producirse tan pronto como el contacto esperado no se produzca (tiempo de inicio), o después de un umbral de tiempo preestablecido, entre otros. Este análisis sobre la estrategia de reacción óptima se deja como trabajo futuro.

4.5.0.6. CGR Probabilístico

La estrategia CGR-2Copies presentada es una primera y limitada aproximación hacia el enrutamiento bajo planes de contacto con incertidumbre. Aunque la proporción de fallas esperada puede utilizarse como parámetro de entrada, la incertidumbre no siempre es homogénea entre todos los nodos, ni constante a lo largo del tiempo. Las versiones futuras deben considerar y explotar los niveles de incertidumbre con mayor granularidad y las métricas adecuadas para poder decidir mejor cuántas copias crear y qué rutas utilizar para enviarlas. Los resultados preliminares obtenidos en este trabajo motivan el desarrollo de soluciones CGR más especializadas que podrían mejorar aún más las métricas de rendimiento y que también se integran con los esquemas de enrutamiento planificados y oportunistas existentes.

4.6. Conclusiones Parciales

En este capítulo hemos realizado un análisis sobre el efecto que tienen las fallas transitorias en el rendimiento de redes satelitales con enrutamiento basado en plan de contactos. Por un lado se analizaron cotas superiores de rendimiento de CGR. Los resultados muestran que la disminución del rendimiento tiene una relación directa con la forma en que se inyectan las fallas, la cantidad de las mismas, y sobre todo, con el grado de consciencia que el algoritmo CGR tiene respecto a dichas fallas. Por otro lado, se evaluaron distintas versiones de esquemas planificados (con y sin copias) y de esquemas oportunisticos en constelaciones satelitales tanto aleatorias como realistas. Los resultados muestran que no existe actualmente una única solución de enrutamiento que sea eficiente en todos los escenarios de fallas y de recursos. Existen distintas relaciones de compromiso entre los tiempos de entrega de los datos y la eficiencia energética con la cual se logra dicha entrega. En los casos en que la proporción de fallas es alta resultan convenientes los esquemas oportunisticos mientras que en casos en que la proporción de fallas es baja resulta conveniente obtener tanta ventaja como sea posible de la topología planificada y hacer frente a la incertidumbre mediante replicación controlada. Estos resultados motivan fuertemente lo que prevemos como trabajo futuro: un CGR probabilístico con características mejoradas de modelado de incertidumbre, replicación y recálculo de rutas.

Capítulo 5

Escalabilidad e Integración

5.1. Introducción

Gran parte del trabajo previo en DTN se ha centrado en optimizar el rendimiento de la entrega de datos y en proponer algoritmos de enrutamiento jerárquico para integrar diferentes tipos de DTNs [103–107]. Sin embargo, el esfuerzo de computo del enrutamiento y su impacto en la escalabilidad de la red se ha considerado un objetivo secundario hasta el momento. Solamente el trabajo presentado en [108] ha propuesto métodos teniendo en cuenta la escalabilidad como un objetivo primordial en DTNs planificadas. Sin embargo, este trabajo no ha considerado constelaciones de satélites realistas en su análisis, ni el estado actual de las soluciones de enrutamiento planificadas tales como CGR.

En este capítulo, se proponen dos mecanismos para mejorar la escalabilidad del procedimiento de enrutamiento a expensas de una disminución aceptable del rendimiento. Además, se brinda información sobre cómo estos procesos tienen un impacto en la relación de compromiso entre la escalabilidad y el rendimiento obtenido en redes satelitales realistas.

5.2. Enrutamiento y Limitaciones de Escalabilidad

El procedimiento de flujo de datos completo generalmente se divide en cuatro etapas: planificación (realizada por un centro de control de misión en Tierra o MOC), enrutamiento (típicamente distribuido pero también puede ser centralizado en tierra), encolamiento y envío (ejecutado localmente por cada nodo DTN). Como se describió en el Capítulo 2, en la etapa de planificación, los planes de contacto se determinan en función de la estimación de episodios futuros de comunicación. Esta tarea implica tener en cuenta tanto la disposición física y orientación de los nodos a lo largo del tiempo, como la configuración de su sistema de comunicación (antena, modulación, potencia de transmisión, etc.). Como resultado, los propagadores orbitales y los modelos de comunicación se combinan para determinar el plan de contacto final que puede modificarse adicionalmente para reducir el consumo de energía o eliminar los contactos que deriven en conflictos [13, 15, 70, 77, 109]. Ya sea que se mantenga en un nodo de planificación centralizado o distribuido a todos los nodos DTN, el plan de contactos es utilizado por algoritmos como CGR para derivar rutas eficientes, que una vez calculadas pueden almacenarse en una tabla de rutas. Posteriormente, cuando se necesita enviar un bundle de datos generado localmente o en tránsito, el proceso de encolamiento se ejecuta recorriendo la tabla de rutas y seleccionando la mejor ruta válida hacia el destino. Como resultado, el bundle se encola hacia el nodo vecino correspondiente al primer contacto de la ruta seleccionada. Finalmente, cuando ocurre un contacto, se lleva a cabo el proceso de envío despachando todos los bundles previamente encolados para el nodo receptor de ese contacto.

Cada ruta está conformada por una secuencia de contactos y, por lo tanto, es válida para un cierto intervalo de tiempo. Cuando se deben calcular las rutas, CGR crea un grafo de contactos en base al plan de contactos de entrada. Esta representación estática facilita la ejecución de algoritmos de red en topologías que evolucionan en el tiempo. Como se discutió en [47] y se detalla en [84], las búsquedas adaptadas de Dijkstra pueden usarse para determinar rutas óptimas sobre estos grafos de contactos. Como se mostró en [110], CGR es un mecanismo de enrutamiento eficaz para dirigir el flujo de bundles en una red DTN caracterizada por contactos deterministas entre un número pequeño o medio de nodos. Sin embargo, la capacidad de CGR para calcular rutas a través de un plan de contactos que comprende millones de contactos resulta inviable. Este problema fue abordado parcialmente en [24] usando varias técnicas para restringir

el esfuerzo de cálculo de rutas y el tamaño de la tabla de rutas. Aunque este enfoque permite ampliar el alcance de CGR a redes más grandes, el problema de la escalabilidad persiste cuando la cantidad de nodos y contactos crece por encima de un cierto límite. Esto motiva el desarrollo de esquemas adicionales que permitan enfrentar el problema de escalabilidad en redes arbitrariamente grandes.

5.3. Modelo de sistema

En esta sección proponemos dos mecanismos cuya idea central consiste en reducir el dominio de enrutamiento definiendo regiones de cálculo y, por lo tanto, limitando el conocimiento de topología proporcionado a cada nodo en la red. Como mostraremos, la reducción de conocimiento se puede hacer a diferentes niveles, y dependiendo del caso, los nodos pueden requerir mecanismos de enrutamiento adicionales para llegar a ciertos destinos.

5.3.1. Global Full Knowledge (GFK)

Nos referiremos como *Global Full Knowledge (GFK)* a un caso de referencia en el que todos los nodos tienen completo conocimiento topológico de todos los demás nodos de la red. En este caso, el plan de contactos puede diseñarse para optimizar alguna métrica de rendimiento o evitar conflictos de recursos pero sin tener en cuenta una división en regiones. Además, utilizaremos CGR como algoritmo de enrutamiento, ya que es la solución más avanzada para las DTNs espaciales.

5.3.2. Definición de Regiones

Para reducir el esfuerzo de cálculo de enrutamiento del caso *GFK*, la red puede dividirse en varios dominios llamados regiones como se ilustra en la Figura 5.1, donde una red espacial de ejemplo se particionó arbitrariamente en seis regiones (A, B, C, D, E, F) de nodos. Definimos una región como un conjunto arbitrario de nodos que podría (aunque no es necesario) estar relacionado con un espacio geográfico real. Se asume que todos los nodos que pertenecen a la misma región tendrán al menos una ruta entre ellos. Sin embargo, debe tenerse en cuenta que, en este tipo de redes, dicha ruta podría expresarse

en una dimensión temporal que modele un flujo de datos de almacenamiento, transporte y envío. Un diagrama más abstracto se ilustra en la Figura 5.2 que muestra la asignación de regiones correspondiente a la red de la Figura 5.1. Se requiere que tanto las regiones como los nodos tengan un identificador único (ID) a lo largo de toda la red, y los nodos pueden ser miembros en una o más regiones al mismo tiempo y con capacidades de comunicación diferentes. Como se ilustra en la Figura 5.2, algunos nodos solo pueden recibir (Rx), transmitir (Tx), o recibir y transmitir bundles en una región determinada. Además, cuando un nodo N tiene la capacidad de recibir bundles como miembro de la región A y transmitir esos bundles como miembros de la región B, definimos que un *vent* $N(A \rightarrow B)$ entre las regiones A y B reside en el nodo N. Dado que los vents son direccionales, $N(A \rightarrow B)$ y $N(B \rightarrow A)$ serían dos vents diferentes residiendo en el mismo nodo. Además, nos referimos a un nodo como *passageway* cuando contiene un conjunto de uno o más vents. De hecho, un passageway puede considerarse como un tipo de *gateway* o *pasarela* capaz de enviar bundles entre diferentes regiones.

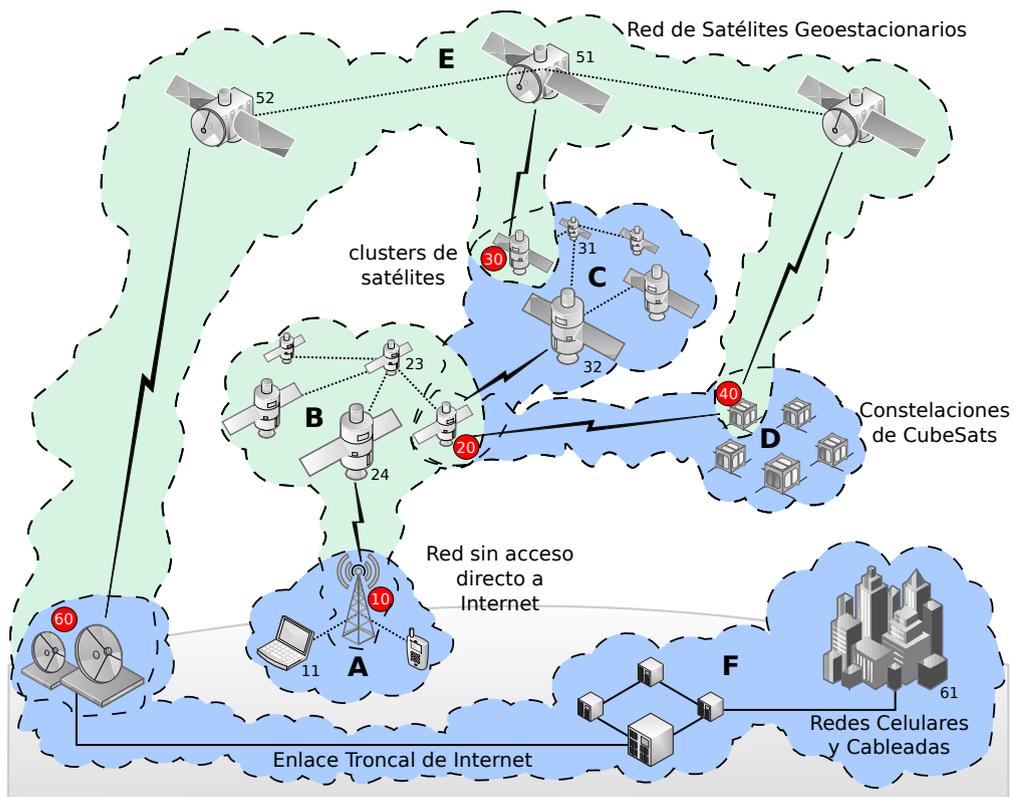


FIGURA 5.1: Red espacial dividida en 6 regiones (A, B, C, D, E, F)

Como se muestra en las siguientes secciones, esta definición de regiones se puede usar para restringir el conocimiento que un nodo determinado recibirá de dos maneras diferentes.

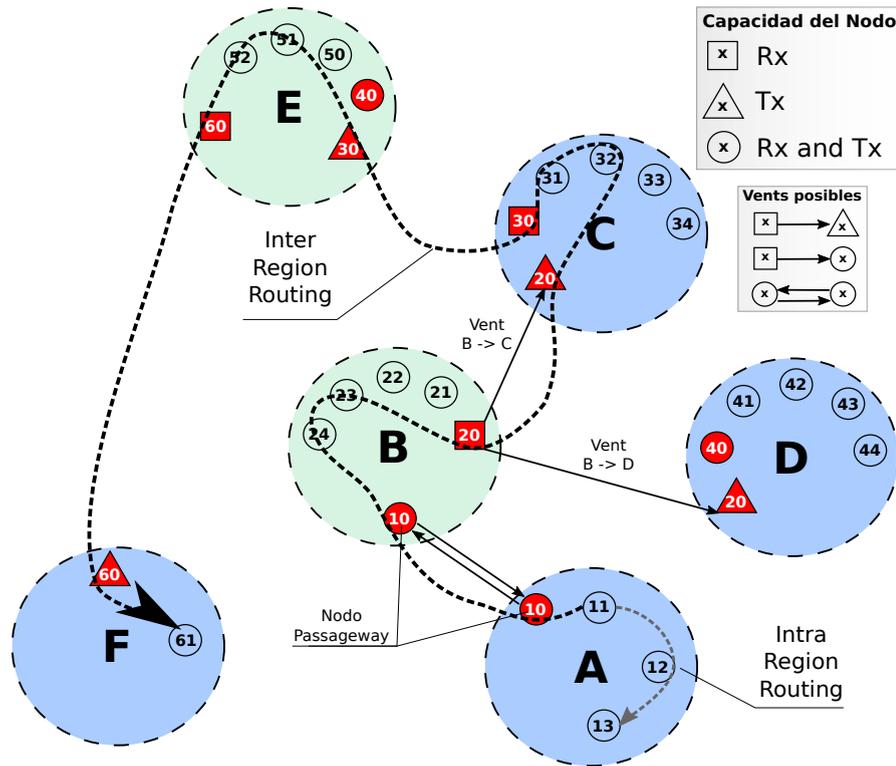


FIGURA 5.2: Regiones y nodos de la red de la Figura 1

5.3.3. Global Regionalized Knowledge (GRK)

Nos referiremos como *regionalización* al proceso de dividir la red en regiones interconectadas por nodos passageway. Luego, la definición de regiones se puede usar para eliminar ciertos contactos del plan de contactos *GFK* o, lo que es lo mismo, permitir solo ciertos contactos en el plan de contactos resultante. Por lo tanto, se mantendrá un contacto $N1 \rightarrow N2$ si $N1$ puede transmitir en la misma región en la que $N2$ puede recibir bundles, o si $N1$ tiene un vent $N1 (R_i \rightarrow R_j)$, y $N2$ puede recibir bundles en la región R_j . Tomando como ejemplo las Figuras 5.1 y 5.2, los contactos $11 \rightarrow 10$ y $10 \rightarrow 23$ se mantendrían en el plan de contactos final, mientras que el contacto $11 \rightarrow 23$ no (a pesar de ser físicamente posible).

Una vez que el conocimiento de la topología se ha reducido siguiendo este proceso, todos los nodos de la red recibirán el mismo plan de contactos que comprenderá la información de todas las regiones. Es decir, los nodos tendrán el conocimiento necesario para llegar a todos los demás nodos de la red mediante la ejecución del algoritmo CGR. Sin embargo, la cantidad de rutas disminuirá después de aplicar esta reducción del plan de contactos. Llamamos *Global Regionalized Knowledge (GRK)* a este mecanismo porque una vez que

el conocimiento se regionaliza, todos los nodos tienen acceso al mismo conocimiento global.

5.3.4. Local Regionalized Knowledge (LRK)

5.3.4.1. Planes de Contactos Regionales

La definición de regiones también se puede usar de una manera radicalmente diferente. Una vez que las regiones se definen como se indica en el caso anterior, y el plan de contactos se reduce en consecuencia, dicho plan se puede dividir en un plan de contactos asociado a cada región. Posteriormente, cada nodo recibirá un plan de contactos por cada región de la cual sea miembro. De esta forma, los nodos que pertenecen a una región recibirán un plan de contactos, mientras que los nodos passageway recibirán dos o más planes de contactos dependiendo de cuántas regiones comuniquen a través de sus vents. Un contacto $N1 \rightarrow N2$ estará en un plan de contactos asociado a una región R_i - C_p si tanto $N1$ como $N2$ pueden transmitir y recibir bundles respectivamente en R_i de acuerdo con la definición de regiones.

5.3.4.2. Enrutamiento Jerárquico

Como en este caso los nodos no recibirán conocimiento topológico sobre los nodos en otras regiones, se necesita un mecanismo adicional para enrutar los datos cuando el destino no es alcanzable con el conocimiento disponible. Por lo tanto, en este caso proponemos que todos los nodos de la red obtengan acceso a la definición de regiones a través de una estructura llamada *region database o base de datos de regiones*, y deberán ejecutar un enrutamiento jerárquico compuesto por un algoritmo de *Enrutamiento Inter-Regional (Inter-RR)* en conjunto con un algoritmo de *Enrutamiento Intra-Regional (Intra-RR)* como CGR.

En general, Intra-RR se puede usar directamente cuando los nodos origen y destino del tráfico son miembros de la misma región. De lo contrario, también se requiere ejecutar Inter-RR. Si tomamos nuevamente el ejemplo de las Figuras 5.1 y 5.2, y el nodo 11 tiene un bundle destinado al nodo 13, Intra-RR es suficiente para enrutar ese bundle hacia su destino. Por otro lado, si el nodo 11 tiene un bundle destinado al nodo 61 en la región F, se requieren los dos procedimientos mencionados:

- Primero, Inter-RR para obtener una *ruta transregional* (compuesta de una secuencia de regiones o vents) a la región destino, de donde se puede derivar a su vez un nodo passageway dentro de la región local. Para que eso sea posible, Inter-RR necesita acceder a la base de datos de regiones para obtener el conjunto de regiones de las cuales 11 es miembro y donde es capaz de transmitir bundles (región A); el conjunto de regiones de las cuales 61 es miembro y donde es capaz de recibir bundles (región F); los nodos passageway de toda la red con sus vents respectivos (nodos 10, 20, 30, 40 y 60 con vents $A \rightarrow B$, $B \rightarrow A$, $B \rightarrow C$, $B \rightarrow D$, $C \rightarrow E$, $D \rightarrow E$ y $E \rightarrow F$).
- Segundo, Intra-RR para obtener una ruta (generalmente compuesta de nodos o contactos) al nodo passageway obtenido del paso anterior (nodo 10).

En el caso en que no se utilice un enrutamiento en el que el nodo fuente guarde la ruta escogida en la cabecera del bundle, estos procedimientos deben repetirse en cada nodo hasta que el nodo passageway 60 reciba dicho bundle. Como ambos nodos 60 y 61 pertenecen a la misma región, 60 solo necesita usar su protocolo Intra-RR para finalmente entregar el bundle al destino.

5.3.4.3. Esquema Inter-RR

Un posible esquema Inter-RR se ilustra en la Figura 5.3, donde se representa un grafo estático conceptual correspondiente al ejemplo descripto. Este grafo es la base del algoritmo Inter-RR propuesto y se crea añadiendo algunos vértices ficticios y auxiliares correspondientes a las regiones de origen (SRC) y destino (DST), y luego un vértice por cada vent que exista en la red. La idea consiste en obtener el camino más corto desde cualquiera de las posibles regiones de origen hacia cualquiera de las posibles regiones de destino conectando estos vents a través de arcos ponderados. Se supone que existe un arco desde un vent a otro cuando la región de recepción del segundo vent es igual a la región de transmisión del primer vent. En ese caso, el segundo vent se considera un vértice vecino del primer vent. Por ejemplo, el vent ($A \rightarrow B$) tiene los vents ($B \rightarrow C$) y ($B \rightarrow D$) como vértices vecinos. En este ejemplo, supondremos que los costos de los arcos para conectar vents son iguales a 1 para todas las regiones atravesadas, con excepción de la región D, que como abarca *CubeSats* de recursos limitados, presentará un costo

mayor. Finalmente, los vértices ficticios están conectados con arcos con costo 0 con los vértices auxiliares con el objetivo de poder obtener la solución mediante la ejecución de una única búsqueda Dijkstra sobre el grafo obtenido en lugar de tener que realizar múltiples búsquedas entre los pares origen-destino. Cabe destacar además, que si el nodo 11 fuera también un miembro transmisor en las regiones hipotéticas G y H, y si el nodo 61 fuera también un miembro receptor en las regiones hipotéticas X y Z, habría arcos adicionales con costo 0 como se ilustra en el grafo de la Figura 5.3.

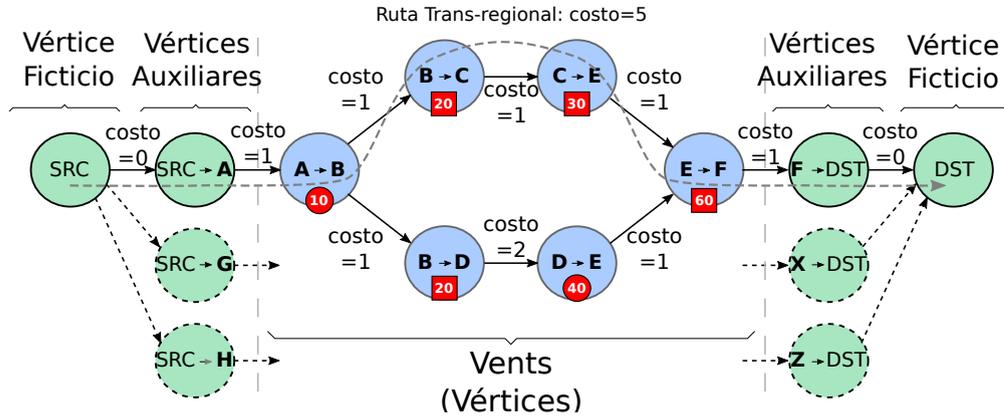


FIGURA 5.3: Grafo estático conformado por el algoritmo de enrutamiento Inter-RR propuesto.

5.3.4.4. Algoritmo Inter-RR

El procedimiento Inter-RR se describe formalmente en el Algoritmo 8. Dado un bundle B y una base de datos de regiones Rdb , un nodo construye el grafo estático mencionado y ejecuta una búsqueda Dijkstra para obtener una ruta transregional a destino ($RegionsRoute$) y finalmente un nodo passageway dentro de la región local (Pn). Las líneas 1-3 obtienen los conjuntos necesarios para crear los vértices del grafo. Las líneas 5-10 comprueban si hay al menos un vent conectando a la región de destino. Si este no es el caso, Inter-RR no podrá encontrar una ruta transregional y, por lo tanto, se devolverá un Pn vacío. En caso contrario, el conjunto de $Vertices$ definido en la línea 11 se llena con las estructuras $Vertex$ definidas en las líneas 12-18. Dado que $RegionsRoute$ será una secuencia de estructuras $Vertex$, cada vértice incluye los atributos para identificar las regiones de la ruta ($fromRegID$, $toRegID$), el passageway correspondiente $pwayID$ y algunos atributos auxiliares para realizar la búsqueda Dijkstra y conformar la ruta ($distFromOrigin$, $visited$, $predecessor$). Las líneas 19 y 20 construyen los vértices correspondientes a la fuente ficticia (NVS) y al destino ficticio (NVD), 21-22 los

correspondientes a las regiones donde el nodo actual $OwnID$ puede transmitir bundles, 23-24 los correspondientes a las regiones donde el nodo destino $B.dstID$ puede recibir bundles, y 25-26 un vértice por cada vent existente en la red. Luego, las líneas 27-33 inicializan los atributos de los vértices antes de realizar la búsqueda Dijkstra en la línea 34. Finalmente, el passageway Pn se obtiene en la línea 35 y se retorna en la línea 36.

Algoritmo 8: Inter-RR

input : bundle to forward B , region database Rdb ,
current node ID $OwnID$

output: passageway node inside local region Pn

```

1  $Vents \leftarrow \text{getVents}(Rdb)$ ;
2  $SrcRegions \leftarrow \text{getTxRegions}(OwnID, Rdb)$ ;
3  $DstRegions \leftarrow \text{getRxRegions}(B.dstID, Rdb)$ ;
4  $Pn \leftarrow \emptyset$ ;
5  $FoundVent \leftarrow false$ ;
6 for  $v \in Vents$  do
7   | if  $v.toRegID \in DstRegions$  then
8   |   |  $FoundVent \leftarrow true$ ;
9 if  $FoundVent == false$  then
10 | return  $Pn$ ;
11 Set  $Vertices$  contains all created  $Vertex$ 
12 Struct  $Vertex$  contains
13 | fromRegID;
14 | toRegID;
15 | pwayID;
16 | distFromOrigin;
17 | visited;
18 | Predecessor;
19  $Vertex$   $NVS("SRC", "SRC")$ ;
20  $Vertex$   $NVD("DST", "DST")$ ;
21 for  $s \in SrcRegions$  do
22 |  $Vertex$   $VS("SRC", s)$ ;
23 for  $d \in DstRegions$  do
24 |  $Vertex$   $VD(d, "DST")$ ;
25 for  $v \in Vents$  do
26 |  $Vertex$   $Vrt(v.fromRegID, v.toRegID, v.pwayID)$ ;
27 for  $v \in Vertices$  do
28 | if  $v == any VS$  then
29 |   |  $v.distFromOrigin \leftarrow 0$ ;
30 | else
31 |   |  $v.distFromOrigin \leftarrow Inf$ ;
32 |   |  $v.visited \leftarrow false$ ;
33 |   |  $v.predecessor \leftarrow \emptyset$ ;
34  $regionsRoute \leftarrow \text{dijkstra}(NVS, NVD, Vertices)$ ;
35  $Pn \leftarrow regionsRoute.firstVertex.pwayID$ ;
36 return  $Pn$ ;
```

5.3.4.5. Utilización de Inter-RR

Es importante tener en cuenta que este esquema se puede utilizar con dos propósitos diferentes:

Por un lado, puede usarse como un medio para mejorar la escalabilidad para redes grandes. Es decir, el mismo procedimiento Intra-RR se puede utilizar a lo largo de toda la red y un nodo disminuirá su esfuerzo de cálculo de enrutamiento debido a la reducción de la información topológica asociada con otras regiones donde el nodo no es miembro.

Por otro lado, como una red DTN puede abarcar subredes compuestas por una gran cantidad de nodos heterogéneos y enlaces con diversas características y funcionalidades, este esquema se puede usar como medio para integrarlas en un marco común donde todos los nodos ejecuten Inter-RR, y donde los protocolos Intra-RR ya existentes, estudiados y optimizados dentro de las subredes, no necesiten ser modificados.

5.4. Evaluación

5.4.1. Entorno de Simulación

Para estudiar y evaluar la relación de compromiso entre escalabilidad y rendimiento en redes DTN planificadas, hemos adaptado el simulador Dtnsim que describimos brevemente en la Sección 3.9.2.

El escenario analizado se corresponde con la ilustración de las Figuras 5.1 y 5.2, con los parámetros orbitales provistos en la tabla 5.1, y con una captura de la red ilustrada en la Figura 5.4. Evaluaremos los mecanismos *GFK*, *GRK* y *LRK* descriptos al considerar los siguientes tres casos de definición de regiones:

- (I) 1 región (A): todos los nodos pertenecen a una región única.
- (II) 4 regiones (A, B, E, F): las regiones B, C y D están unificadas en una sola región B.
- (III) 6 regiones (A, B, C, D, E, F): este caso se corresponde exactamente con la ilustración de las Figuras 5.1 y 5.2.

Además, con el objetivo de brindar resultados representativos de topologías satelitales reales, hemos utilizado el software Satellite Tool Kit (STK) para generar los planes de contactos de entrada para cada nodo para un horizonte de tiempo de 24 horas. Este período de tiempo permite que cada satélite complete varias órbitas alrededor de la Tierra.

Con respecto al tráfico, variamos la cantidad de bundles generados desde el nodo con ID 11 a la estación terrena con ID 61, y analizamos las métricas de *esfuerzo de cálculo de enrutamiento* y *tiempo de entrega medio por bundle* en los diferentes casos. El *esfuerzo de enrutamiento* representa la suma del tiempo de procesamiento dedicado en la simulación por todos los nodos de la red a la tarea de calcular y usar rutas para enviar los bundles. Las simulaciones se han ejecutado en una computadora portátil Lenovo Thinkpad E430 con un procesador Intel Core i3-2330M, 8 GB de RAM, en un sistema operativo Ubuntu 16.10, de 64 bits. El *tiempo de entrega promedio por bundle* se obtiene promediando, para todos los bundles generados, el tiempo entre que dicho bundle es generado y el momento en que se entrega al destino.

LEOs						
Region y IDs de Nodos	Eje <i>SemiMayor</i> [Km]	Excen- tricidad	Incli- nación [grados]	Arg. de Perigeo [grados]	RAAN [grados]	Anomalía Verdadera [grados]
<i>B: 20-24</i>	6500	0	90	0,20,40,60,80	0	0
<i>C: 30-34</i>	6500	0	0	0,20,40,60,80	0	5
<i>D: 40-44</i>	6500	0	60	0,20,40,60,80	30	-20

Estaciones Terrenas		
Nombre de Nodo y ID	Latitud	Longitud
Polo Sur: 10	-90°	0°
Objetivo: 11	-89,9°	0°
White Sand: 60	32,5007°	-106,609°
Canberra: 61	-35,4008°	148,973°

GEOs	
Nombre de Nodo y ID	Longitud
TDRS_3: 50	-87,83°
TDRS_7: 51	76,136°
TDRS_9: 52	-12,719°

TABLA 5.1: Parámetros satelitales

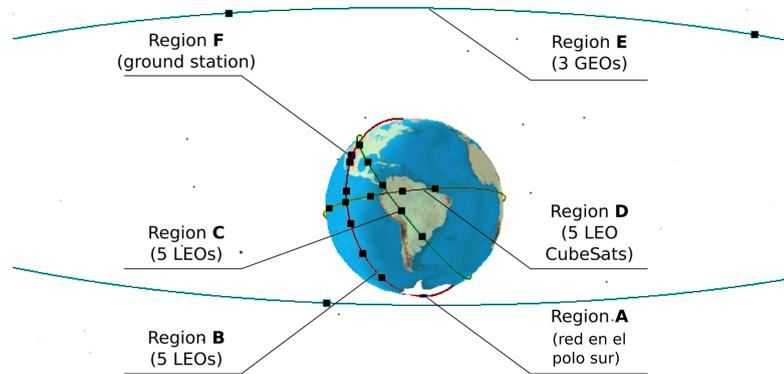


FIGURA 5.4: Captura de la red espacial de las Figuras 1 y 2

5.4.2. Análisis de Resultados

Los resultados para el *esfuerzo de cálculo de enrutamiento* y el *tiempo de entrega medio por bundle* se pueden observar en la Figura 5.5. Cuando se usa solo una región, *GFK*, *GRK* y *LRK* proporcionan los mismos resultados, los cuales usaremos como el caso de referencia donde no se aplican modificaciones a la estrategia de enrutamiento actualmente utilizada. Por otro lado, los esquemas *GRK* y *LRK* se analizan cuando la red se divide en más regiones.

Lo primero que se puede notar es que el *esfuerzo de cálculo del enrutamiento* se reduce a medida que se definen más regiones. Esto sucede porque el hecho de restringir los dominios de enrutamiento y permitir que solo ciertos nodos comuniquen esos dominios provoca una reducción en el tamaño del plan de contactos, y por lo tanto, en la cantidad de rutas que deben ser descubiertas y calculadas. Además, está claro que *LRK* obtiene una mejora con respecto a *GRK* para el mismo número de regiones. La mejora adicional de *LRK* se basa en el hecho de que en este caso los nodos solo necesitan calcular rutas dentro de las regiones locales, mientras que en *GRK* los nodos requieren calcular las rutas hacia cualquier destino en la red. Aunque es cierto que el esquema *LRK* necesita ejecutar el esquema Inter-RR adicional, este algoritmo es mucho más simple que CGR porque no considera las comunicaciones variables en el tiempo entre las regiones. Por lo tanto, la combinación de Inter-RR con CGR sobre planes de contactos regionales termina siendo menos demandante en términos computacionales que ejecutar CGR sobre un plan de contactos que, a pesar de haber sido reducido, abarca toda la red con contactos variables en el tiempo.

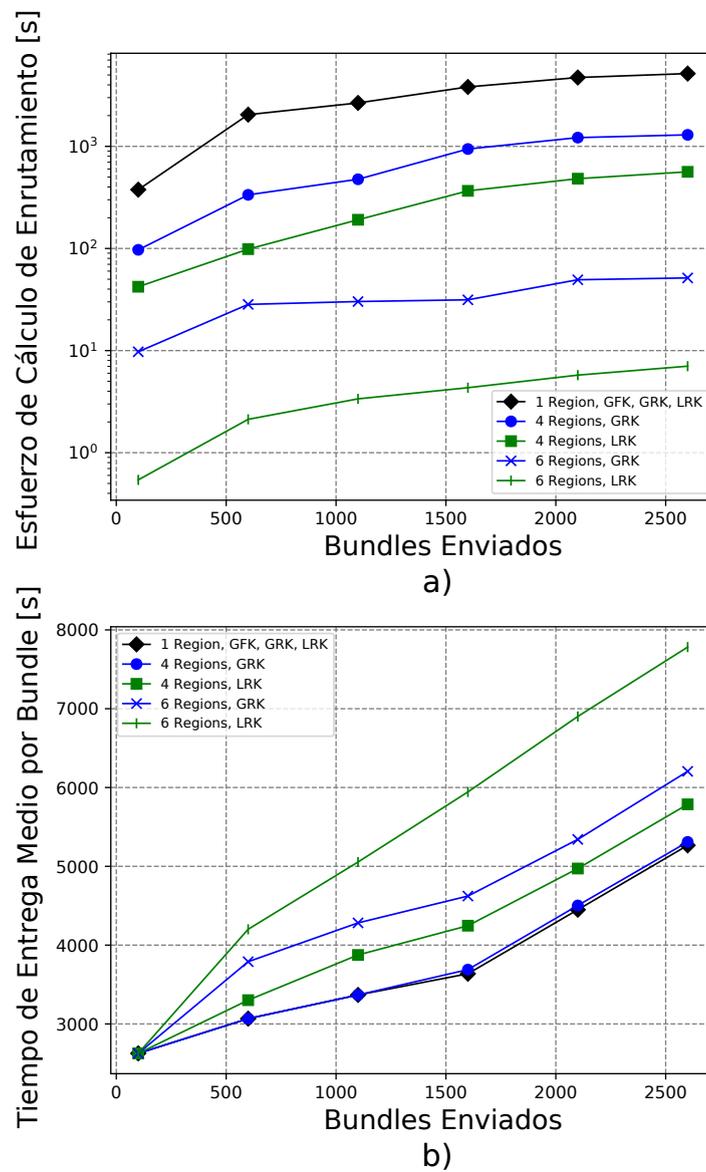


FIGURA 5.5: Resultados de simulación. a) Escalabilidad. b) Rendimiento

Con respecto al rendimiento, de la Figura 5.5 b) se desprende que la penalización resulta mayor a medida que la red se divide en más regiones. Esto se puede explicar por el hecho de que a medida que se definen más regiones, más rutas que se encuentran presentes en el caso *GFK* se han eliminado en los casos *GRK* y *LRK*. Además, este comportamiento se ve exacerbado en *LRK* debido al hecho de considerar enlaces estáticos entre regiones en lugar de contactos variables en el tiempo, como en el caso *GRK*.

Como se muestra en esta sección, existe una notable relación de compromiso entre la escalabilidad y el rendimiento cuando se reduce la información topológica provista a los

nodos. Sin embargo, la ganancia en escalabilidad es considerable y existe cierta flexibilidad para actuar sobre esta relación. Como se discutió, esto se puede lograr dividiendo correctamente la red en regiones, y también eligiendo un cierto número y ubicación de nodos passageway. Desde el punto de vista de un operador de red, el desafío consiste en definir las regiones de tal manera que la calidad y cantidad de rutas resultantes proporcionen un flujo de datos global aceptable. Además de la división en regiones, hemos propuesto dos formas diferentes de utilizar esas regiones para restringir el conocimiento de la topología. Esto agrega una nueva dimensión de flexibilidad cuando se busca un objetivo de rendimiento o escalabilidad.

5.5. Conclusiones Parciales

En este capítulo, hemos propuesto dos mecanismos para lograr la escalabilidad de las redes DTN planificadas en el espacio. Hemos analizado la relación de compromiso entre rendimiento y escalabilidad al simular diferentes casos de uso de una red espacial realista. Como conclusión general los resultados muestran que hay dos dimensiones para actuar sobre esta relación. Por un lado, es importante diseñar y seleccionar un número apropiado de regiones y nodos passageway, que luego se pueden usar para restringir el conocimiento topológico proporcionado a cada nodo. Por otro lado, cada mecanismo propuesto realiza la reducción de conocimiento topológico en un nivel diferente. A medida que se realiza una mayor reducción de conocimiento, mayor es la ganancia en términos de escalabilidad, pero también mayor es la penalización impuesta en términos de rendimiento de la red. Por lo tanto, el desafío de un operador de red consiste en utilizar estas herramientas como un medio para lograr el equilibrio requerido.

Capítulo 6

Conclusiones

En el inicio de esta tesis se comenzó por describir las motivaciones y limitaciones de los sistemas satelitales distribuidos, tanto en el contexto internacional como en Argentina a través de las misiones de observación de la Tierra propuestas por *CONAE*. Particularmente, se analizaron las causas que restringen el uso de protocolos de comunicación tradicionales, tales como los utilizados actualmente en Internet, y se introdujo la arquitectura DTN como un cambio paradigmático en la forma de comunicar constelaciones satelitales. Se describió el estado del arte en DTN y algunos de los desafíos actuales en la implementación operativa de esta arquitectura. Concretamente, esta investigación doctoral se enfocó en los problemas de congestión [2–4], incertidumbre [5–8] y escalabilidad [9, 10]. La hipótesis general planteada inicialmente se basó en que resulta posible mejorar el rendimiento de las comunicaciones mediante técnicas que actúen sobre la planificación de los contactos y/o el algoritmo de enrutamiento. De esta manera, luego de brindar el marco conceptual en el Capítulo 1, y de establecer el modelado en el Capítulo 2, se procedió a validar la hipótesis poniendo el foco en cada uno de los problemas mencionados. En cada caso, los aportes se orientaron a proponer mecanismos y/o modificaciones que actúan sobre el diseño del plan de contactos, sobre el algoritmo de enrutamiento, o sobre ambos.

Con respecto a la congestión, en el Capítulo 3 se contribuyó con estrategias novedosas para evitar la congestión de manera proactiva. Por un lado, se introdujo *LPA-CGR* que extiende *CGR* al considerar la capacidad total de las rutas, lo que le permite mitigar la congestión provocada por tráfico local. En segundo lugar, se propuso *GPA-CGR* que

aprovecha la previsibilidad del tráfico para evitar completamente la congestión global mediante la reserva de recursos de comunicación por adelantado. Además, se extendió este enfoque con un algoritmo evolutivo llamado *EGPA-CGR* que proporciona mayor flexibilidad en la definición de la función objetivo a optimizar.

En lo referido a la incertidumbre, en el Capítulo 4 se realizó un análisis sobre el efecto que tienen las fallas transitorias sobre el enrutamiento basado en plan de contactos. Por un lado, se analizaron cotas superiores de rendimiento, y por otro lado, se propusieron y evaluaron distintas variantes de esquemas tanto planificados como oportunistas, que utilizan la replicación de paquetes para hacer frente a las fallas. En los casos en que la proporción de fallas es alta, se mostró que resultan convenientes los esquemas oportunistas, mientras que en casos en que la proporción de fallas es relativamente baja, resulta conveniente aprovechar tanto como sea posible la información de la topología planificada y hacer frente a la incertidumbre mediante replicación controlada.

Con respecto a la escalabilidad, en el Capítulo 5 se plasmó un trabajo llevado a cabo mediante colaboración internacional con investigadores del *Jet Propulsion Laboratory* de *NASA* y *Caltech*. Concretamente, se propusieron dos esquemas de planificación y enrutamiento llamados *GRK* y *LRK* orientados a mejorar la escalabilidad mediante la disminución del esfuerzo de cálculo de rutas. Se demostró además que existe una estrecha relación de compromiso entre la escalabilidad obtenida y la penalización en el rendimiento, y se propusieron técnicas que podrían ser utilizadas por un operador de red para lograr el equilibrio requerido.

De esta manera, el presente trabajo contribuyó con mecanismos originales que aportan soluciones para desplegar y comunicar constelaciones satelitales. Esto las habilita a compartir recursos y trabajar de forma colaborativa para cumplir misiones distribuidas de forma eficiente. Además del ámbito internacional, la investigación descrita resulta particularmente aplicable a las nuevas misiones propuestas por la agencia espacial Argentina (CONAE), y forma parte de un complejo entramado que comenzará a mostrar sus frutos en los próximos años.

Bibliografía

- [1] J. Fraire. Diseño de plan de contactos para redes satelitales tolerantes a interrupciones. Master thesis, Universidad Nacional de Cordoba, Cordoba, Argentina, Noviembre 2015.
- [2] Pablo G. Madoery, Juan A. Fraire, and Jorge M. Finochietto. Congestion management techniques for disruptiontolerant satellite networks. *International Journal of Satellite Communications and Networking*, 36(2):165–178, May 2017. doi: 10.1002/sat.1210. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/sat.1210>.
- [3] P. G. Madoery, J. A. Fraire, and J. M. Finochietto. Analysis of communication strategies for earth observation satellite constellations. *IEEE Latin America Transactions*, 14(6):2777–2782, June 2016. ISSN 1548-0992. doi: 10.1109/TLA.2016.7555254.
- [4] J. A. Fraire, P. Madoery, J. M. Finochietto, and E. J. Birrane. Congestion modeling and management techniques for predictable disruption tolerant networks. In *2015 IEEE 40th Conference on Local Computer Networks (LCN)*, pages 544–551, Oct 2015. doi: 10.1109/LCN.2015.7366369.
- [5] P. Madoery, F. Raverta, J. Fraire, and J. Finochietto. Routing in space delay tolerant networks under uncertain contact plans. In *2018 IEEE International Conference on Communications (ICC)*, Kansas, MO, USA, May 2018. IEEE.
- [6] P. G. Madoery, F. D. Raverta, J. A. Fraire, and J. M. Finochietto. On the performance analysis of disruption tolerant satellite networks under uncertainties. In *2017 XVII Workshop on Information Processing and Control (RPIC)*, pages 1–6, Sept 2017. doi: 10.23919/RPIC.2017.8214346.

- [7] J. A. Fraire, P. Madoery, S. Burleigh, M. Feldmann, J. Finochietto, A. Charif, N. Zergainoh, and R. Velazco. Assessing contact graph routing performance and reliability in distributed satellite constellations. *Journal of Computer Networks and Communications*, 2017:1–18, 2017. doi: 10.1155/2017/2830542. URL <https://doi.org/10.1155/2017/2830542>.
- [8] F. Raverta, R. Demasi, P. Madoery, J. Fraire, J. Finochietto, and P. D’Argenio. A markov decision process for routing in space dtns with uncertain contact plans. In *5th IEEE International Workshop on Space-Terrestrial Internetworking (STINT)*, Huntsville, AL, USA, December 2018. IEEE.
- [9] P. Madoery, J. Fraire, F. Raverta, J. Finochietto, and S Burleigh. Managing routing scalability in space dtns. In *5th IEEE International Workshop on Space-Terrestrial Internetworking (STINT)*, Huntsville, AL, USA, December 2018. IEEE.
- [10] Pablo Madoery, Carlos Zerbini, Juan Fraire, and Jorge Finochietto. Enhancing contact graph routing forwarding. performance for segmented satellites architectures. In *2017 1st IAA Latin American Symposium on Small Satellites: Advanced Technologies and Distributed Systems*, march 2017.
- [11] J. Fraire, P. Madoery, and J. Finochietto. On the design of fair contact plans for delay tolerant networks. In *2013 IEEE International Conference on Wireless for Space and Extreme Environments (WiSEE)*, 2013.
- [12] J. Fraire, P. Madoery, and J. Finochietto. Leveraging routing performance and congestion avoidance in predictable delay tolerant networks. In *2014 IEEE International Conference on Wireless for Space and Extreme Environments (WiSEE)*, Noordwick, Netherlands, October 2014. IEEE.
- [13] J. A. Fraire, P. G. Madoery, and J. M. Finochietto. On the design and analysis of fair contact plans in predictable delay-tolerant networks. *IEEE Sensors Journal*, 14(11):3874–3882, Nov 2014. ISSN 1530-437X. doi: 10.1109/JSEN.2014.2348917.
- [14] J. A. Fraire, P. G. Madoery, J. M. Finochietto, and G. Leguizamón. Preliminary results of an evolutionary approach towards contact plan design for satellite dtns. In *2015 IEEE International Conference on Wireless for Space and Extreme Environments (WiSEE)*, pages 1–7, Dec 2015. doi: 10.1109/WiSEE.2015.7393097.

- [15] Juan A. Fraire, Pablo G. Madoery, and Jorge M. Finochietto. Traffic-aware contact plan design for disruption-tolerant space sensor networks. *Ad Hoc Networks*, 47:41 – 52, 2016. ISSN 1570-8705. doi: <https://doi.org/10.1016/j.adhoc.2016.04.007>. URL <http://www.sciencedirect.com/science/article/pii/S1570870516301032>.
- [16] Juan A. Fraire, Pablo G. Madoery, Jorge M. Finochietto, Pablo A. Ferreyra, and Raoul Velazco. Internetworking approaches towards along-track segmented satellite architectures. In *2016 IEEE International Conference on Wireless for Space and Extreme Environments (WiSEE)*. IEEE, sep 2016. doi: 10.1109/wisee.2016.7877316. URL <https://doi.org/10.1109/wisee.2016.7877316>.
- [17] J. Barrientos, A. Ferral, L. Cara, J. Fraire, R. Velazco, P. Madoery, and P. Ferreyra. A segmented architecture approach to provide a continuous, long-term, adaptive and cost-effective glaciers monitoring system based on dtn communications and cubesat platforms. In *2017 1st IAA Latin American Symposium on Small Satellites: Advanced Technologies and Distributed Systems*, march 2017.
- [18] F. Raverta P. Madoery J. Finochietto, J. Fraire and R. Cherini. Enhancing successful data delivery in small satellites. In *2017 1st IAA Latin American Symposium on Small Satellites: Advanced Technologies and Distributed Systems*, march 2017.
- [19] Juan A. Fraire, Pablo G. Madoery, Jorge M. Finochietto, and Guillermo Leguizamón. An evolutionary approach towards contact plan design for disruption-tolerant satellite networks. *Applied Soft Computing*, 52:446 – 456, 2017. ISSN 1568-4946. doi: <https://doi.org/10.1016/j.asoc.2016.10.023>. URL <http://www.sciencedirect.com/science/article/pii/S1568494616305361>.
- [20] Juan A. Fraire, Pablo Madoery, and Jorge M. Finochietto. *Contact Plan Design for Predictable Disruptiontolerant Space Sensor Networks*, chapter 7, pages 123–150. Wiley-Blackwell, 2017. ISBN 9781119126492. doi: 10.1002/9781119126492.ch7. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119126492.ch7>.
- [21] Juan A. Fraire, Pablo Madoery, Fernando Raverta, Jorge M. Finochietto, and Raoul Velazco. DtnSim: Bridging the gap between simulation and implementation of space-terrestrial DTNs. In *2017 6th International Conference on Space Mission*

- Challenges for Information Technology (SMC-IT)*. IEEE, sep 2017. doi: 10.1109/smc-it.2017.27. URL <https://doi.org/10.1109/smc-it.2017.27>.
- [22] Pablo G. Madoery, Juan A. Fraire, Fernando D. Raverta, and Jorge M. Finochietto. On the trade-off analysis of centralized and distributed routing in delay tolerant satellite networks. In *Proceedings of the IV School on Systems and Networks, SSN 2018, Valdivia, Chile, October 29-31, 2018.*, pages 89–93, 2018. URL http://ceur-ws.org/Vol-2178/SSN2018_paper_32.pdf.
- [23] Fernando D. Raverta, Pablo G. Madoery, Juan A. Fraire, and Jorge M. Finochietto. Modeling routing in scheduled delay tolerant networks under uncertainties. In *Proceedings of the IV School on Systems and Networks, SSN 2018, Valdivia, Chile, October 29-31, 2018.*, pages 94–97, 2018. URL http://ceur-ws.org/Vol-2178/SSN2018_paper_33.pdf.
- [24] Juan A. Fraire, Pablo G. Madoery, Amir Charif, and Jorge M. Finochietto. On route table computation strategies in delay-tolerant satellite networks. *Ad Hoc Networks*, 2018. ISSN 1570-8705. doi: <https://doi.org/10.1016/j.adhoc.2018.07.002>. URL <http://www.sciencedirect.com/science/article/pii/S1570870518304359>.
- [25] J.A. Fraire, J.M. Finochietto, and S.C. Burleigh. *Delay-Tolerant Satellite Networks*. Artech House space technology and applications library. Artech House, 2017. ISBN 9781630813444. URL <https://books.google.com.ar/books?id=XK5ptAEACAAJ>.
- [26] Scott Burleigh. Tortoise and hare: Ways of thinking about mission communications. In *2011 Workshop on Terrestrial and Space DTN, Space Internetworking Center, Xanthi, Thrace*, 2011. URL https://trs.jpl.nasa.gov/bitstream/handle/2014/43490/11-1849_A1b.pdf?sequence=1.
- [27] Alex McMahon and Stephen Farrell. Delay- and disruption-tolerant networking. *IEEE Internet Computing*, 13(6):82–87, November 2009. ISSN 1089-7801. doi: 10.1109/MIC.2009.127. URL <https://doi.org/10.1109/MIC.2009.127>.
- [28] K. Fall and S. Farrell. Dtn: An architectural retrospective. *IEEE J.Sel. A. Commun.*, 26(5):828–836, June 2008. ISSN 0733-8716. doi: 10.1109/JSAC.2008.080609. URL <https://doi.org/10.1109/JSAC.2008.080609>.

- [29] IPN. DTN Tutorial. http://ipnsig.org/wp-content/uploads/2012/07/DTN_Tutorial_v2.04.pdf, 2012.
- [30] V. Cerf and R. Kahn. A protocol for packet network intercommunication. *IEEE Transactions on Communications*, 22(5):637–648, May 1974. ISSN 0090-6778. doi: 10.1109/TCOM.1974.1092259.
- [31] Kevin Fall. A delay-tolerant network architecture for challenged internets. In *Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '03, pages 27–34, New York, NY, USA, 2003. ACM. ISBN 1-58113-735-4. doi: 10.1145/863955.863960. URL <http://doi.acm.org/10.1145/863955.863960>.
- [32] V. Cerf et al. RFC-4838: Delay-tolerant networking architecture. Network Working Group, IETF, April 2007.
- [33] K. Scott and S. Burleigh. RFC-5050: Bundle protocol specification. Network Working Group, IETF, November 2007.
- [34] M. Demmer, J. Ott, and S. Perreault. RFC-7242: Delay-tolerant networking tcp convergence-layer protocol. Request for Comments RFC 7242, Internet Research Task Force (IRTF), Jun 2014.
- [35] H. Kruse, S. Jero, and S. Ostermann. RFC-7122: datagram convergence layers for the delay and disruption tolerant networking (dtn) bundle protocol and licklider transmission protocol (ltp). Request for Comments RFC 7122, Internet Research Task Force (IRTF), Mar 2014.
- [36] Yahui Wu, Su Deng, Hongbin Huang, and Yiqi Deng. Performance analysis of epidemic routing in delay tolerant networks with overlapping communities and selfish nodes. *International Journal of Computers Communications and Control*, 8(5), 2013.
- [37] T. Spyropoulos, K. Psounis, and S. Cauligi. Spray and wait: an efficient routing scheme for intermittently connected mobile networks. ACM SIGCOMM, 2005.
- [38] A. Lindgren, A. Doria, E. Davies, and Grasic. RFC-6693: Probabilistic routing protocol for intermittently connected networks.

- [39] Anders Lindgren, Avri Doria, and Olov Schelén. Probabilistic routing in intermittently connected networks. *SIGMOBILE Mob. Comput. Commun. Rev.*, 7(3): 19–20, July 2003. ISSN 1559-1662. doi: 10.1145/961268.961272.
- [40] J. Burgess, B. Gallagher, D. Jensen, and B. Levine. Maxprop: Routing for vehicle-based disruption-tolerant networks. IEEE INFOCOM, April 2006.
- [41] D. A. Vallado. *Fundamentals of Astrodynamics and Applications - 4th Edition*. Microcosm, Hawthorne, CA, 2007.
- [42] S. Merugu, M. Ammar, and Zegura E. Routing in space and time in networks with predictable mobility. Technical report, Georgia Institute of Technology, 2006. Tech Report GIT-CC 04-7.
- [43] S. Burleigh. Contact graph routing, IETF-Draft. Jul 2010.
- [44] J. Wyatt, S. Burleigh, J. Ross, L. Torgerson, and Wissler S. Disruption Tolerant Networking Flight Validation Experiment on NASAs EPOXI Mission. In *IEEE SPACOMM*, 2009.
- [45] N. Bezirgiannidis, C. Caini, D.D. Padalino Montenero, M. Ruggieri, and V. Tsoulos. Contact graph routing enhancements for delay tolerant space communications. In *Advanced Satellite Multimedia Systems Conference and the 13th Signal Processing for Space Communications Workshop (ASMS/SPSC), 2014 7th*, pages 17–23, Sept 2014. doi: 10.1109/ASMS-SPSC.2014.6934518.
- [46] E. Birrane, S. Burleigh, and N. Kasch. Analysis of the contact graph routing algorithm: Bounding interplanetary paths. *Acta Astronautica*, 75:108–119, July 2012.
- [47] J. Segui, E. Jennings, and S. Burleigh. Enhancing contact graph routing for delay tolerant space networking. In *2011 IEEE Global Telecommunications Conference (GLOBECOM 2011)*, pages 1–6, Houston, TX, USA, December 2011. IEEE.
- [48] DTN2: a DTN reference implementation. <http://www.dtnrg.org/wiki/Dtn2Documentation>.
- [49] Sebastian Schildt, Johannes Morgenroth, Wolf-Bastian Pöttner, and Lars Wolf. Ibrdtn: A lightweight, modular and highly portable bundle protocol implementation. In *Electronic Communications of the EASST*, 2011.

- [50] IBR-DTN: A modular and lightweight implementation of the bundle protocol. <http://trac.ibr.cs.tu-bs.de/project-cm-2012-ibrdtm>.
- [51] *Micro Planetary Communication Network* (μ pcn). <https://upcn.eu/>.
- [52] Postellation: a Lean and Deployable DTN Implementation. <http://postellation.viagenie.ca/>.
- [53] A. Sudarsono and T. Nakanishi. An implementation of secure data exchange in wireless delay tolerant network using attribute-based encryption. In *Computing and Networking (CANDAR), 2014 Second International Symposium on*, pages 536–542, Dec 2014.
- [54] S. Burleigh. Interplanetary overlay network: An implementation of the dtn bundle protocol. In *4th IEEE Consumer Comms. and Networking Conf., CCNC 2007*, pages 222–226, Las Vegas, NV, 2007.
- [55] Wolf-Bastian Pottner, Johannes Morgenroth, Sebastian Schildt, and Lars C Wolf. An empirical performance comparison of DTN bundle protocol implementations. In *ACM MobiCom 2011 Workshop on Challenged Networks (CHANTS'11)*, Las Vegas, Nevada, USA, 9 2011.
- [56] Interplanetary Overlay Network (ION). <http://sourceforge.net/projects/ion-dtn/>.
- [57] S. Burleigh. RFC-6260: Compressed bundle header encoding specification. Network Working Group, IETF, Nov 2011.
- [58] E. Birrane. Internet-Draft: Streamlined bundle security protocol specification. Network Working Group, IETF (<https://datatracker.ietf.org/doc/draft-birrane-dtn-sbsp/>), May 2014.
- [59] E. Birrane and V. Ramachandran. Internet-Draft: Delay tolerant network management protocol. Network Working Group, IETF (<https://tools.ietf.org/html/draft-irtf-dtnrg-dtnmp-01>), Dec 2014.
- [60] M. Demmer and J. Ott. Delay-tolerant networking tcp convergence-layer protocol. Network Working Group, IETF (<https://tools.ietf.org/html/rfc7242>), Jun 2014.

- [61] H. Kruse, S. Jero, and S. Ostermann. Datagram convergence layers for the delay- and disruption-tolerant networking (dtn) bundle protocol and licklider transmission protocol (ltp). Network Working Group, IETF (<https://tools.ietf.org/html/rfc7122>), Mar 2014.
- [62] author. Inquiry into the heart of a comet. *Science and Children*, 48(6):46–49, Feb 2011.
- [63] W. Ivancic, W. Eddy, D. Stewart, L. Wood, J. Northam, and C. Jackson. Experience with delay-tolerant networking from orbit. *Int. Journal of Satellite Comms. and Networking*, 28(5-6):335–351, September 2010.
- [64] L. Wood, W. Eddy, C. Smith, W. Ivancic, and C. Jackson. Internet-Draft: Saratoga: A scalable data transfer protocol. Network Working Group, IETF (<https://tools.ietf.org/html/draft-wood-tsvwg-saratoga-16>), Oct 2014.
- [65] L. Wood, W.M. Eddy, W. Ivancic, J. McKim, and C. Jackson. Saratoga: a delay-tolerant networking convergence layer with efficient link utilization. In *Satellite and Space Communications, 2007. IWSSC '07. International Workshop on*, pages 168–172, Sept 2007.
- [66] L. Wood, D. Shell, W. Ivancic, B. Conner, E. Miller, D. Stewart, and D. Hodgson. Cleo and vmoc: enabling warfighters to task space payloads. In *Military Communications Conference, 2005. MILCOM 2005. IEEE*, pages 3052–3058 Vol. 5, Oct 2005.
- [67] L. Wood and P. Holliday. Internet-Draft: Using http for delivery in delay/disruption-tolerant networks. Network Working Group, IETF (<https://tools.ietf.org/html/draft-wood-dtnrg-http-dtn-delivery-09>), Jun 2014.
- [68] C. Caini and R. Firrincieli. Application of contact graph routing to LEO satellite DTN communications. In *IEEE International Conference on Communications (ICC)*, pages 3301–3305, June 2012.
- [69] C. Caini, H. Cruickshank, S. Farrell, and M. Marchese. Delay and disruption tolerant networking: An alternative solution for future satellite networking applications. In *Proceedings of the IEEE*, volume 99, pages 1980–1997, November 2011.

- [70] M. Huang, S. Chen, Y. Zhu, and Y. Wang. Cost-efficient topology design problem in time-evolving delay-tolerant networks. In *IEEE GLOBECOM*, pages 1–5, Dec 2010.
- [71] M. Huang, S. Chen, F. Li, and Y. Wang. Topology design in time-evolving delay-tolerant networks with unreliable links. In *IEEE GLOBECOM*, pages 5296–5301, 2012.
- [72] CONAE. Plan Espacial Nacional. <http://www.cetam.fadu.uba.ar/wp-content/uploads/2015/01/Actualizacion-Plan-Espacial-2010-2015.pdf>.
- [73] R. Velazco, P. Fouillat, and R. Reis. *Radiation Effects on Embedded Systems*. Springer, 2007.
- [74] O. Brown and P. Eremenko. The value proposition for fractionated space architectures. In *AIAA-2006-7506, AIAA Space 2006*, San Jose, CA, 2006.
- [75] M. N. Sweeting. Modern small satellites-changing the economics of space. *Proceedings of the IEEE*, 106(3):343–361, March 2018. ISSN 0018-9219. doi: 10.1109/JPROC.2018.2806218.
- [76] T. Berners-Lee. RFC-3986: Uniform resource identifier (uri). Network Working Group, IETF, January 2005.
- [77] J. Fraire and J. Finochietto. Design challenges in contact plans for disruption-tolerant satellite networks. *IEEE Communications Magazine*, May 2015.
- [78] ITU-R. Recommendation itu-r s.1325-3. International Telecommunication Union, 2003.
- [79] H. Mendoza and G. Corral-Briones. Interferencia en sistemas distribuidos de satélites de orbita media y baja. In *XV Reunión de Trabajo Procesamiento de la Información y Control (RPIC)*, pages 1110–1115, San Carlos de Bariloche, Argentina, Sep 2013.
- [80] J. Fraire and J.M. Finochietto. Routing-aware fair contact plan design for predictable delay tolerant networks. *Ad Hoc Networks*, 25:303 – 313, 2015. ISSN 1570-8705. doi: <https://doi.org/10.1016/j.adhoc.2014.07.006>. URL <http://www.>

- sciencedirect.com/science/article/pii/S1570870514001371. New Research Challenges in Mobile, Opportunistic and Delay-Tolerant Networks Energy-Aware Data Centers: Architecture, Infrastructure, and Communication.
- [81] G. Araniti, N. Bezirgiannidis, E. Birrane, I. Bisio, S. Burleigh, C. Caini, M. Feldmann, M. Marchese, J. Segui, and K. Suzuki. Contact graph routing in dtn space networks: overview, enhancements and performance. *IEEE Comms. Magazine*, 53(3):38–46, March 2015. ISSN 0163-6804. doi: 10.1109/MCOM.2015.7060480.
- [82] E. W. Dijkstra. A note on two problems in connection with graphs. *Numer. Math.*, 1(1):269–271, Dec 1959. ISSN 0029-599X. doi: 10.1007/BF01386390.
- [83] S. Burleigh. Interplanetary overlay network: An implementation of the dtn bundle protocol. In *2007 4th IEEE Consumer Communications and Networking Conference*, pages 222–226, Jan 2007. doi: 10.1109/CCNC.2007.51.
- [84] Edward Birrane, Scott Burleigh, and Niels Kasch. Analysis of the contact graph routing algorithm: Bounding interplanetary paths. *Acta Astronautica*, 75:108 – 119, 2012. ISSN 0094-5765. doi: <http://dx.doi.org/10.1016/j.actaastro.2012.02.004>.
- [85] J. A. Fraire, P. Madoery, J. M. Finochietto, and E. J. Birrane. Congestion modeling and management techniques for predictable disruption tolerant networks. In *2015 IEEE 40th Conference on Local Computer Networks (LCN)*, pages 544–551, Oct 2015. doi: 10.1109/LCN.2015.7366369.
- [86] E.J. Birrane. Congestion modeling in graph-routed delay tolerant networks with predictive capacity consumption. In *Global Communications Conference (GLOBECOM), 2013 IEEE*, pages 3016–3022, Dec 2013. doi: 10.1109/GLOCOM.2013.6831534.
- [87] Hongcheng Yan, Qingjun Zhang, and Yong Sun. Local information-based congestion control scheme for space delay/disruption tolerant networks. *Wireless Networks*, 21(6):2087–2099, 2015. ISSN 1572-8196. doi: 10.1007/s11276-015-0911-6. URL <http://dx.doi.org/10.1007/s11276-015-0911-6>.
- [88] S. Burleigh, C. Caini, J. J. Messina, and M. Rodolfi. Toward a unified routing framework for delay-tolerant networking. In *2016 IEEE International Conference*

- on Wireless for Space and Extreme Environments (WiSEE)*, pages 82–86, Sept 2016. doi: 10.1109/WiSEE.2016.7877309.
- [89] CCSDS. *Schedule-Aware Bundle Routing (SABR), White Book*. Consultative Committee for Space Data Systems (CCSDS), ccsds 232.0-b-2 edition, Feb 2016. Proposed Recommendation for Space Data System Standards.
- [90] M. Allman, V. Paxson, ICSI, E. Blanton, and Purdue University. RFC-5681: Tcp congestion control. Request for Comments RFC 5681, Internet Engineering Task Force (IETF), Sep 2009.
- [91] Aloizio P. Silva, Scott Burleigh, Celso M. Hirata, and Katia Obraczka. A survey on congestion control for delay and disruption tolerant networks. *Ad Hoc Networks*, 25, Part B:480 – 494, 2015. ISSN 1570-8705. doi: <http://dx.doi.org/10.1016/j.adhoc.2014.07.032>. URL <http://www.sciencedirect.com/science/article/pii/S1570870514001668>. New Research Challenges in Mobile, Opportunistic and Delay-Tolerant Networks Energy-Aware Data Centers: Architecture, Infrastructure, and Communication.
- [92] T. C. Hu. Multi-commodity network flows. *Oper. Res.*, 11(3):344–360, June 1963. ISSN 0030-364X. doi: 10.1287/opre.11.3.344. URL <http://dx.doi.org/10.1287/opre.11.3.344>.
- [93] Sushant Jain, Kevin Fall, and Rabin Patra. Routing in a delay tolerant network. In *Proceedings of the 2004 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, SIGCOMM '04*, pages 145–158, New York, NY, USA, 2004. ACM. ISBN 1-58113-862-8. doi: 10.1145/1015467.1015484. URL <http://doi.acm.org/10.1145/1015467.1015484>.
- [94] Juan Alonso and Kevin Fall. A linear programming formulation of flows over time with piecewise constant capacity and transit times. Technical report, 2003.
- [95] El-Ghazali Talbi. *Metaheuristics : from design to implementation*. J. Wiley & Sons, Hoboken (N.J.), 2009. ISBN 978-0-470-27858-1. URL <http://opac.inria.fr/record=b1128380>.
- [96] John H. Holland. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. MIT Press, Cambridge, MA, USA, 1992. ISBN 0262082136.

- [97] NASA: A-Train, 10.26.10 (april 2012). <http://atrain.nasa.gov/>.
- [98] András Varga and Rudolf Hornig. An overview of the omnet++ simulation environment. In *SIMUTOOLS 2008*, pages 60:1–60:10, Belgium, 2008. ISBN 978-963-9799-20-2.
- [99] J. A. Fraire, P. G Madoery, F. Raverta, J. M. Finochietto, and R. Velazco. Dtnsim: Bridging the gap between simulation and implementation of space-terrestrial dtms. In *Space Mission Challenges for Information Technology (SMC-IT), 2017 IEEE Int. Conference on*, Sept 2017.
- [100] J. A. Fraire and P. A. Ferreyra. Assessing dtn architecture reliability for distributed satellite constellations: Preliminary results from a case study. In *Biennial Congress of Argentina (ARGENCON), 2014 IEEE*, pages 564–569, June 2014.
- [101] Guohua Zhang, Jing Wang, and Yonghe Liu. Congestion management in delay tolerant networks. In *Proceedings of the 4th Annual International Conference on Wireless Internet, WICON '08*, pages 65:1–65:9, ICST, Brussels, Belgium, Belgium, 2008. ISBN 978-963-9799-36-3.
- [102] Linton C. Freeman. A set of measures of centrality based on betweenness. *Sociometry*, pages 35–41, 1977. URL <http://moreno.ss.uci.edu/23.pdf>.
- [103] Mirco Musolesi and Cecilia Mascolo. A framework for multi-region delay tolerant networking. In *Proceedings of the 2008 ACM Workshop on Wireless Networks and Systems for Developing Regions, WiNS-DR '08*, pages 37–42, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-190-3. doi: 10.1145/1410064.1410072. URL <http://doi.acm.org/10.1145/1410064.1410072>.
- [104] W. Qi, W. Hou, L. Guo, Q. Song, and A. Jamalipour. A unified routing framework for integrated space/air information networks. *IEEE Access*, 4:7084–7103, 2016. doi: 10.1109/ACCESS.2016.2618905.
- [105] K. A. Harras and K. C. Almeroth. Inter-regional messenger scheduling in delay tolerant mobile networks. In *2006 International Symposium on a World of Wireless, Mobile and Multimedia Networks(WoWMoM'06)*, pages 10 pp.–102, 2006. doi: 10.1109/WOWMOM.2006.53.

-
- [106] T. Bhotmange and M. Penurkar. Region based routing in delay tolerant networks. In *2015 International Conference on Computing Communication Control and Automation*, pages 315–321, Feb 2015. doi: 10.1109/ICCUBEA.2015.66.
- [107] Hao Wen, Jia Liu, Chuang Lin, Fengyuan Ren, Pan Li, and Yuguang Fang. Rena: region-based routing in intermittently connected mobile network. In *MSWiM*, 2009.
- [108] Cong Liu and Jie Wu. Scalable routing in delay tolerant networks. In *Proceedings of the 8th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc '07, pages 51–60, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-684-4. doi: 10.1145/1288107.1288115. URL <http://doi.acm.org/10.1145/1288107.1288115>.
- [109] P. Muri et al. Topology design and performance analysis for networked earth observing small satellites. In *MILCOM Proceedings*, pages 1940–1945, Baltimore, Maryland, Nov 2011.
- [110] G. Wang, S. C. Burleigh, R. Wang, L. Shi, and Y. Qian. Scoping contact graph-routing scalability: Investigating the system’s usability in space-vehicle communication networks. *IEEE Vehicular Technology Magazine*, 11(4):46–52, Dec 2016. ISSN 1556-6072. doi: 10.1109/MVT.2016.2594796.