



**HAL**  
open science

## Contributions to the performance study of interplanetary networks

Mahendiran Prathaban

► **To cite this version:**

Mahendiran Prathaban. Contributions to the performance study of interplanetary networks. Other [cs.OH]. Institut National des Télécommunications, 2010. English. NNT : 2010TELE0003 . tel-00538532

**HAL Id: tel-00538532**

**<https://theses.hal.science/tel-00538532v1>**

Submitted on 22 Nov 2010

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Ecole Doctorale EDITE

Thèse présentée pour l'obtention du diplôme de  
DOCTEUR DE L'INSTITUT NATIONAL DES TELECOMMUNICATIONS

*Doctorat délivré conjointement par  
L'Institut National des Télécommunications et l'Université Pierre et Marie Curie - Paris 6*

Spécialité :  
INFORMATIQUE

Par  
Mahendiran PRATHABAN

# Contributions to the Performance Study of Interplanetary Networks

Soutenue le 05 Février 2010 devant le jury composé de :

Guy Pujolle	Professeur à l'Université Pierre et Marie Curie	Président
Pascal Lorenz	Professeur à l'Université de Haute Alsace	Rapporteur
Radu Dan Rugescu	Professeur à l'Université Politehnica de Bucharest	Rapporteur
Zhili Sun	Professeur à l'Université de Surrey	Examineur
Monique Becker	Professeur à Telecom SudParis	Examineur
Joséphine Kohlenberg	Enseignant Chercheur à Telecom SudParis	Encadrant
Gérard Hébuterne	Professeur à Telecom SudParis	Directeur de thèse

Thèse n° 2010TELE0003

## Abstract

Contributions to The Performance Study of Interplanetary Networks

by

Mahendiran Prathaban

Doctor of Philosophy in Computer Science

Télécom SudParis

Professor Gérard HEBUTERNE

Acquiring knowledge about space has fascinated humans since the antiquity. Since a few decades, there is a significant increase in space mission programs like Mars and Lunar Exploration Missions, and Spatial exploration projects become more and more complex, involving several different communicating nodes: satellites, robots (Rovers, Landers) or "aerorobots" (i.e. atmospheric sensors). Most of these nodes can communicate directly with earth. Furthermore, some inter-robot (sensor network) or robot-satellite communications are also required, which means that interplanetary networks, by nature, no longer consist of plain point-to-point interplanetary links but consists in complex networks formed by heterogeneous nodes. Such networks are particularly challenging, considering that connectivity between nodes is usually transient due to orbital movements and planet rotations. For this reason, space missions exhibit communication needs shifting away from the pure telecommunication standpoint and therefore a full-fledged network stack should be designed to fulfill these requirements.

To this extend, efforts have been made by the space standardization bodies such as the Consultative Committee for Space Data Systems (CCSDS) and the Delay Tolerant Networking working group within the Internet Research Task Force (IRTF) to develop an effective protocol architecture for space missions. CCSDS developed complete protocol stack, alternative to the TCP/IP Suite due to its performance limitations over interplanetary network in consequence of large propagation delays as well as consistent information losses. DTNRG has devised an overlay network architecture named Delay Tolerant Architecture (DTN), which are able to tolerate link disruptions and long delay.

In this dissertation, we have analyzed various newly proposed protocol stacks (CCSDS, DTN) for their performance over Interplanetary Network and derived the parameters which affect the performance of the system, like buffer storage, power limi-

tation, also derived the QoS requirement of the specific space applications and proposed a new resource aware routing algorithm, which take in to consideration network and resource limitation to account for routing and forwarding decision with help of an effective buffer management policy.

The space networking protocols works on store and forward paradigm, like CCSDS File delivery protocols. From the analysis we have found that the throughput performance of the protocol mostly depends on the pre-set timeout value of its timer due to varying propagation delay. To improve the performance of the protocol, we have proposed a dynamic timer algorithm which sets the timeout values according to network condition for all the timers in the protocol, in a way that reduces unnecessary retransmission of data PDUs, thereby increasing the throughput performance and also bandwidth usage.

Since there are various interoperable protocols proposed by space agencies, in our work we have developed QoS framework that, when attached to the given application, can select the required underlying protocol stack according to application QoS requirements, which are best suited for the given network conditions to improve overall performance. Both positive and less than optimal results are produced from an implementation of this framework in ns-2.30. Positive results are observed, when the environment is dynamic with long distance, and for the application without real-time requirements. Less optimal results are observed when the framework introduces overhead without parallel performance improvement. Moreover CCSDS File delivery application with both dynamic timer and QoS framework over resource aware routing provide optimal performance with unavoidable overheads.

## Résumé

L'acquisition de connaissances sur l'espace fascine les humains depuis l'antiquité. Depuis quelques décennies, il y a augmentation significative des programmes spatiaux concernant Mars, des missions d'exploration lunaire et des projets d'exploration spatiale de plus en plus complexes, impliquant plusieurs nœuds de communication: satellites, robots (Rovers, Landers) ou "aerorobots" (ie capteurs atmosphériques). La plupart de ces nœuds peuvent communiquer directement avec la terre. En outre, certaines communications inter-robot (réseaux de capteurs) ou robot-satellite sont également requises, ce qui signifie que les réseaux interplanétaires, par nature, ne correspondent plus à des réseaux simples point-à-point, mais les liens interplanétaires sont des réseaux complexes formés de nœuds hétérogènes. De tels réseaux sont particulièrement difficiles, étant donné que la connectivité entre les nœuds est habituellement transitoire en raison des mouvements orbitaux et des rotations de la planète. Pour cette raison, les missions de l'espace ont des besoins de communication qui s'éloignent aujourd'hui de la télécommunication habituelle et nécessitent donc un réseau à part entière avec toute la pile des protocoles qui devra être conçue pour satisfaire ces exigences.

Dans ce cadre, des efforts ont été faits par les organismes de normalisation pour l'espace comme le Comité consultatif pour les systèmes de données spatiales (CCSDS) et le Delay Tolerant Networking groupe de travail au sein de l'Internet Research Task Force (IRTF) pour élaborer une architecture de protocoles efficace pour des missions spatiales. Le CCSDS a développé une pile complète de protocoles, alternative à la pile TCP / IP pour pallier aux limitations de performances sur le réseau interplanétaire causées par les grands délais de propagation et les pertes de cohérence des informations. Le DTNRG a mis au point une architecture de réseau overlay nommé Delay Tolerant Architecture (DTN), qui est capable de tolérer des perturbations du lien et de longs délais.

Dans cette thèse, nous avons analysé quelques unes de ces propositions : (les piles de protocole CCSDS, et DTN) pour leurs performances sur le réseau interplanétaire et obtenu les paramètres qui affectent les performances du système, comme la mémoire tampon, les limitations de puissance. Nous avons également obtenu la QoS exigée pour les applications spatiales spécifiques et proposé un nouvel algorithme de routage prenant en compte les ressources disponibles, qui prend en considération les limitations des ressources du réseau pour le routage et transmet les décisions permettant une politique efficace de gestion des tampons.

Les protocoles de réseaux spatiaux fonctionnent sur un 'store and forward

paradigme', comme les protocoles de remise de fichiers CCSDS. De l'analyse nous avons déduit que les performances du débit dépendent surtout de la valeur préétablie du timeout de son timer en raison des variations du délai de propagation. Pour améliorer les performances du protocole, nous avons proposé un algorithme dynamique du timer qui fixe la valeur du time-out en fonction de l'état du réseau pour tous les timers du protocole, afin de réduire les retransmissions inutiles de données d'UFC, accroissant ainsi les performances de débit et d'utilisation de la bande passante.

Comme il existe différentes piles de protocoles d'interopérabilité proposées par l'agence spatiale, dans nos travaux, nous avons développé un framework de QoS qui est joint à chaque application, et peut sélectionner la pile de protocoles sous-jacents adéquate, selon les exigences de qualité de service des applications, qui sont les mieux adaptés pour améliorer les performances globales de ce réseau. Des résultats positifs et des résultats moins bons que les résultats optimaux sont obtenus à partir d'une simulation de ce framework en ns-2.30. Des résultats positifs sont observés, lorsque l'environnement est dynamique avec une longue distance et pour une application sans exigences temps réel. De moins bons résultats sont observés lorsque le framework introduit de l'overhead, sans amélioration de la performance parallèle. En outre les applications de transmission de fichier CCSDS avec à la fois un timer dynamique et un framework de la QoS sur le routage prenant en compte les ressources offrent des performances optimales compte tenu des overheads inévitables.

*"Dedicated to my beloved parents & family"*

# Contents

<b>List of Figures</b>	<b>vi</b>
<b>List of Tables</b>	<b>ix</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Context	2
1.1.1 Why Not Terrestrial Protocols in IPN	3
1.2 Research Issues	4
1.3 Contribution	5
1.4 Outline	8
<b>2 State of the art</b>	<b>9</b>
2.1 Network Architectures	9
2.1.1 Mars-Earth backbone network	10
2.1.2 Mars vehicle proximity networks	10
2.1.3 Local area networks (LAN's) on-board the Mars vehicles	11
2.2 Communication Architecture	11
2.2.1 Mars Near-Term Communication Architecture	11
2.2.2 Mars Mid-Term Communication Architecture	11
2.2.3 Mars Far-Term Communication Architecture	13
2.3 Communication Challenges in Space Environment	14
2.3.1 High Signal Propagation Delays	14
2.3.2 High Data Corruption Rates	14
2.3.3 Meager, Asymmetric Bandwidth	14
2.3.4 Blackouts or Intermittent Connectivity	15
2.4 Characteristic of TCP over Mars-Earth backbone network	15
2.4.1 TCP over Error-Prone Links	16
2.4.2 TCP over Asymmetric Channels	16
2.4.3 TCP over Limited Link Capacity	17
2.4.4 TCP over Intermittent Connections	17
2.4.5 Discussion	18
2.5 Protocol Standards in Deep space Missions	18
2.5.1 The Consultative Committee for Space Data Systems (CCSDS)	19
2.5.2 Space Communications Protocol Standards (SCPS)	22
2.5.3 Towards Delay Tolerant Networking Architecture	23
2.6 Delay and Disruption-Tolerant Networking (DTN)	24
2.6.1 DTN Architecture	25



2.6.2	Bundle Protocol (BP)	26
2.6.3	Licklider Transmission Protocol (LTP)	28
2.7	Discussion	30
<b>3</b>	<b>Performance of DTN protocols in Interplanetary Network</b>	<b>31</b>
3.1	Deep Space Network Architecture	32
3.1.1	IPN Backbone Network	32
3.1.2	Planetary Network	33
3.1.3	Reference Scenario	33
3.2	DTN routing protocols performance	35
3.2.1	DTN Routing	35
3.3	DTN bundle protocols performance Comparisons	39
3.3.1	Packet Delivery Ratio	40
3.3.2	Impact of Buffer Constraint	41
3.3.3	Impact of Packet Size	43
3.3.4	Performance over Intermitted Link	43
3.4	Buffer Management Policy for DTN	44
3.4.1	Control strategy for Real Time traffic	45
3.4.2	Control strategy for Image data traffic	46
3.4.3	Performance of the buffer management policy	49
3.5	Discussion	57
<b>4</b>	<b>Resource Aware Routing for Interplanetary Mars Surface Network</b>	<b>58</b>
4.1	Motivation	58
4.2	Why to Study Mars Rovers	59
4.3	Mars Exploration Rovers - Antenna	60
4.3.1	Resource Limitations	60
4.3.2	Communication Limitations	62
4.4	Rover Communication Links - characters	62
4.4.1	Network Architecture	63
4.4.2	Direct-To-Earth Link	64
4.4.3	Rover-Orbiter Relay Link	65
4.4.4	Lander-Rover Link	66
4.4.5	Power Estimation	67
4.5	Cost Based Antenna Selection Algorithm	69
4.5.1	Basic Assumptions	69
4.5.2	Network Model	70
4.5.3	Cost Function	71
4.5.4	Selection Rules	72
4.5.5	Selection Procedure	73
4.6	Implementation and Analysis	76
4.7	Resource Aware Routing Protocol	80
4.7.1	Buffer Control Strategy	81
4.7.2	Information Sharing and Antenna Selection	82
4.7.3	Routing	83
4.8	Functionality Analysis	86
4.8.1	Next Hop selection % for different scenarios	87
4.9	Performance Analysis	87

4.9.1	Simulation Topology . . . . .	88
4.9.2	Performance of the protocol for best effort traffic . . . . .	89
4.9.3	Performance of the protocol for throughput sensitive traffic . . . . .	91
4.9.4	Performance of the protocol for Delay sensitive traffic . . . . .	92
4.9.5	Comparison with PROPHET . . . . .	94
4.10	Discussion . . . . .	94
<b>5</b>	<b>Dynamic NAK Timer Algorithm for CCSDS File Delivery Protocol</b>	<b>96</b>
5.1	Need for Dynamic timeout value . . . . .	96
5.2	CCSDS File Delivery Protocol (CFDP) . . . . .	97
5.2.1	Design Concept . . . . .	97
5.2.2	Protocol Operation . . . . .	98
5.2.3	Deferred NAK Mode . . . . .	102
5.3	Related Works . . . . .	103
5.4	Dynamic Timer Algorithm . . . . .	104
5.4.1	Preliminaries . . . . .	104
5.4.2	EOF and NAK Timer Algorithm . . . . .	105
5.5	Implementation and Validation . . . . .	109
5.5.1	Validation . . . . .	109
5.5.2	Analytical Model and its assumptions . . . . .	110
5.5.3	Counter Threshold . . . . .	112
5.6	Performance Analysis . . . . .	114
5.6.1	Simulation Topology . . . . .	114
5.6.2	Metric . . . . .	115
5.6.3	Performance on symmetric channel . . . . .	115
5.6.4	Performance on asymmetric channel . . . . .	115
5.6.5	Number of Retransmission Required . . . . .	117
5.6.6	Throughput performance in Intermitted connectivity . . . . .	118
5.6.7	Number of File . . . . .	119
5.6.8	Impact of Buffer Size . . . . .	120
5.6.9	Would Immediate NAK Mode Make a Difference? . . . . .	120
5.7	Discussion . . . . .	121
<b>6</b>	<b>QoS based Protocol Selection in Interplanetary Network</b>	<b>122</b>
6.1	Motivation . . . . .	122
6.2	QoS mapping in Interplanetary Network . . . . .	123
6.3	QoS framework to select protocol stack for communication . . . . .	124
6.3.1	Application Layer QoS Metrics . . . . .	125
6.3.2	Measurement Module . . . . .	126
6.3.3	Admission Control . . . . .	126
6.3.4	Mapping Table . . . . .	127
6.3.5	Protocol Choice . . . . .	128
6.3.6	Autonomic Behavior . . . . .	131
6.4	Implementation and Analysis . . . . .	132
6.4.1	Demonstration . . . . .	132
6.5	Performance Analysis . . . . .	134
6.5.1	Telemetry Application Scenario . . . . .	134
6.5.2	Scientific Data Application Scenario . . . . .	138

6.5.3	Multimedia Scenario . . . . .	140
6.5.4	Processing delay . . . . .	141
6.6	Discussion . . . . .	142
<b>7</b>	<b>Conclusion</b> . . . . .	<b>143</b>
7.1	Contributions . . . . .	143
7.2	Perspectives . . . . .	146
	<b>Résumé - Contributions aux Études de Performances des Réseaux In-</b> <b>terplanétaires</b> . . . . .	<b>147</b>
	<b>Bibliography</b> . . . . .	<b>196</b>

# List of Figures

1.1	Interplanetary Internet - Future Mars Mission [Image Courtesy: NASA [19] ] . . . . .	2
2.1	Mars Near-Term Communication Architecture [Image Courtesy: NASA [73]] . . . . .	12
2.2	Mars Mid-Term Communication Architecture [Image Courtesy: NASA [73]]	12
2.3	Mars Far-Term Communication Architecture [Image Courtesy: NASA [73]]	13
2.4	Space Link Protocols. [Image Courtesy: CCSDS [24]] . . . . .	20
2.5	CFDP Operation [Image Courtesy: CCSDS [25]] . . . . .	20
2.6	SCPS End-to-End Services. [Image Courtesy: SCPS [83]] . . . . .	22
2.7	An Example of Data Flow in a Bundling-based Interplanetary Internet. [Image Courtesy: JPL/ NASA [79] . . . . .	26
2.8	Current Bundling Architecture. [Image Courtesy: JPL/ NASA [82]] . . . . .	28
3.1	Interplanetary Architecture [Image Courtesy: NASA [19] ] . . . . .	32
3.2	Planetary Network Architecture [Image Courtesy: NASA [19] ] . . . . .	33
3.3	The Reference Scenario . . . . .	34
3.4	Message Exchange in the Epidemic Routing protocol [Image Courtesy: [95] ] . . . . .	37
3.5	Conceptual performance vs knowledge trade-off [Image Courtesy: [91] ] . . . . .	39
3.6	Packet delivery ratio over Simulation time . . . . .	40
3.7	Packet Delivery ratio with varying buffer size . . . . .	41
3.8	Delivery ratio with different packet size . . . . .	42
3.9	Delivery ratio with intermitted links . . . . .	44
3.10	Packets Incoming and Queuing Sequence in Buffer . . . . .	48
3.11	Buffer Controller Blocks . . . . .	50
3.12	Packets Drop Percentage at Source Rover Node . . . . .	51
3.13	Comparison of PDP at Source Rover Node . . . . .	52
3.14	Buffer Queue Length at Source Rover Node . . . . .	52
3.15	Packet Delivery Ratio for Real Time Traffic . . . . .	54
3.16	Data Delivery Time Performance . . . . .	54
3.17	Packet Delivery Ratio for Best effort Traffic . . . . .	56
3.18	Mean percent change of maximum network goodput for Best effort Traffic . . . . .	56
4.1	Power Available in the Rover for 1 sol day for different Antenna size . . . . .	61
4.2	Basic Network Architecture . . . . .	63
4.3	Attenuation Measure on Mars surface between Lander and Rover . . . . .	67

4.4	Power Required for Transmission in Rover depending on data rate . . . .	68
4.5	Data rate depending on Bandwidth . . . . .	69
4.6	Logical scheme of the algorithm . . . . .	75
4.7	Algorithm results due to the power constraints for a sol . . . . .	78
4.8	Algorithm results due to the delay constraints for a sol . . . . .	79
4.9	Antenna Selection % upon two different rules . . . . .	80
4.10	Functional Blocks of Routing Protocol . . . . .	81
4.11	Next Hop selection % for each Algorithm . . . . .	88
4.12	Delivery ratio with different buffer size, varying visibility duration % . .	90
4.13	Packet Delivery time with different buffer size, varying visibility duration % . . . . .	90
4.14	Delivery ratio with different visibility duration %, varying power level .	91
4.15	Delay, link, buffer . . . . .	92
4.16	Performance comparison with PROPHET protocol for Best Effort Traffic	93
5.1	Protocol Transmission Sequence for Deferred NAK Mode [Image Courtesy [25]] . . . . .	100
5.2	Time-out Triggered NAK Retransmission [Image Courtesy [25]] . . . . .	101
5.3	Time-out Triggered EOF and Finished Retransmissions [Image Courtesy [25]] . . . . .	102
5.4	Deferred NAK mode: Where $T_{prop}$ stands for one-way propagation delay, and $RT_k$ stands for the duration of the $k^{th}$ retransmission spurt. ("Trans- mission spurt" refers to consecutive transmission of PDUs back to back.). [Image Courtesy [53]] . . . . .	103
5.5	Validation of Simulation Model . . . . .	111
5.6	File delivery Time over various BER in single hope scenario . . . . .	112
5.7	Counter Threshold . . . . .	113
5.8	Simulation Topology . . . . .	113
5.9	File Delivery Time over symmetric channel . . . . .	116
5.10	File Delivery Time over asymmetric channel . . . . .	116
5.11	Number of Retransmission required to complete 90% of file transfer for 10Mb file . . . . .	117
5.12	Throughput Performance over Number of PDUs for 50% & 30% Link availability period . . . . .	119
5.13	Number of File completely transfer where File Size = 1MB . . . . .	120
6.1	QoS Framework to select transport protocol . . . . .	124
6.2	Periodicity of Link . . . . .	126
6.3	Throughput for changing delayed SACK factor with different PER and RTT=120s . . . . .	135
6.4	Throughput for changing file size with delayed SACK factor d = 5 and 15 with framework re-configuration . . . . .	136
6.5	Goodput percentage as a function of checkpoint ratio for different loss rate	137
6.6	File transfer time of SCPS-TP as the function of packet size for different error rate . . . . .	139
6.7	Minimum buffer size at different load with link availability period of 80%	140
6.8	Throughput performance for different RTT with length of NIL packet is set as L=14 and L=6 . . . . .	141

7.1	Interplanetary Internet - Future Mars Mission [Image Courtesy: NASA [19] ] . . . . .	148
7.2	Scénario de référence . . . . .	158
7.3	Taux de livraison des paquets avec liaisons intermittents . . . . .	160
7.4	Comparaison des deux politiques " Priority " et " Drop Tail " . . . . .	163
7.5	Analyse Macroscopique . . . . .	165
7.6	Contrainte de puissance . . . . .	170
7.7	Bloc fonctionnel de protocole de routage . . . . .	170
7.8	Proportion de sélection du prochain saut pour chaque algorithme . . . . .	173
7.9	Taux de livraison des paquets avec durée de visibilité 70% et 30% . . . . .	175
7.10	Validation du modèle de simulation . . . . .	181
7.11	FDT sur canal asymétrique et symétrique . . . . .	182
7.12	Cadre de QoS . . . . .	185
7.13	Performance de débit pour différents RTT . . . . .	189

# List of Tables

3.1	Simulation Parameters . . . . .	40
3.2	Simulation Parameters . . . . .	50
4.1	Comparison of MER DTE and UHF Relay Link . . . . .	59
4.2	Link parameters relations . . . . .	64
4.3	DTE Link Parameters . . . . .	65
4.4	Rover - Orbiter Relay Link Parameter . . . . .	66
4.5	Rover-Lander Link Parameters . . . . .	66
4.6	know parameters at the Rover . . . . .	70
4.7	Expected Value of Power, Delay & Error Parameters . . . . .	72
4.8	QoS Metric for different interplanetary application . . . . .	73
4.9	Simulation Parameters . . . . .	77
4.10	Simulation Parameter . . . . .	87
4.11	Simulation Parameters . . . . .	89
5.1	Abbreviations used for various PDU types . . . . .	99
5.2	Notation used . . . . .	105
5.3	Simulation Parameters . . . . .	114
5.4	BER and the corresponding PDU error rate. . . . .	114
6.1	QoS Metric for different interplanetary application . . . . .	125
6.2	BER for different atmospheric conditions . . . . .	126
6.3	Mapping Table of Interplanetary applications . . . . .	129
6.4	Configuration Parameters . . . . .	132
6.5	Goodput and End to End Delay for Error rate of $10^{-3}$ . . . . .	138
6.6	Simulation Parameters . . . . .	138
6.7	Process delay introduced by the TCL Middleware . . . . .	142
7.1	Les tables de correspondance . . . . .	186
7.2	Retard de Traitement . . . . .	190

## Acknowledgments

I am heartily thankful to my supervisor, Joséphine Kohlenberg, whose encouragement, guidance and support from the initial to the final level enabled me to develop an understanding of the subject.

I would like to express my sincere gratitude to my advisor Prof. Gérard Hébuterne for the continuous support of my PhD study and research, for his patience, motivation and immense knowledge. His guidance helped me in all the time of research and writing of this thesis.

I owe my deepest gratitude to Prof. Monique Becker, who enlightening me the first glance of research and give me an opportunity to work in SAMAVOR lab. Without her support, advice, I could never have embarked and started all of this work. Ever since, I came to France for my masters, she gave her assistance & guidance and encouragement. And I am thankful for all her support.

I would sincerely thank my thesis reporters Prof. Pascal Lorenz (University of Haute Alsace) and Prof. Radu D. Rugescu (University Politehnica) for their encouragement, insightful comments and also accepting to be in my thesis committee. I wish to express my warm and sincere thanks to Prof. Guy Pujolle (Professor at UPMC), Prof. Zhili Sun (University of Surrey) for accepting to be jury for my thesis.

I would like to thank, Dr. Michel Marot for his professional support and taste of French language. And Dr. Vicent Gauthier, for his friendly advice, professional guidance and Mac support.

I would like to thank Dr Sandoche Balakrichenan, for his friendly support, and professional guidance throughout my stay in France. I also like to thank my labmate and friends, Ashish Gupta, Christina Moreno, Sanjay Kanade for all the fun & intellectual chats we have had in the last four years.

I convey special acknowledgement to department secretary Françoise ABAD for her friendly attitude and approachability with made all my administrative work easy.

It is a pleasure to express my gratitude wholeheartedly to Mme. Ravi's family & Priya for their love, affection, kind hospitality during my stay in France.

Where would I be without my family? My parents deserve special mention for their inseparable support and prayers. My Father, Mr. Prathaban, in the first place is the person who put the fundamental, my learning character, showing me the joy of intellectual pursuit ever since I was a child. My Mother, Revathi, is the one who sincerely raised me with her caring, love & spiritual Knowledge. My brothers and Revathi Anni thanks for being supportive and caring siblings.



And I offer my regards and blessings to all of those who supported me in any respect during the completion of my thesis.

# Chapter 1

## Introduction

Acquiring knowledge about space has fascinated humans since the antiquity. Since a few decades, mankind is able to send people and machines towards near space to acquire valuable knowledge concerning the history of our universe, astrophysics, etc. Considering the time required to travel to such distances, observers cannot simply collect data and bring it back to earth when they return. Transmitting measurements results as soon as they are acquired allows to exploit this data rapidly and eventually to reconfigure the sensing devices according to analysis results. Space networking has now become a reality with InterPlanetary Networks.

Figure 7.1 shows the visions of what InterPlanetary Networks (IPN) could be and how complex their architecture might be (in addition, surface communications are not shown). Spatial exploration projects become more and more complex, involving several different communicating nodes: satellites, robots (landers, rovers) or aerorobots (i.e. atmospheric sensors). Most of these nodes can communicate directly with earth. However, this causes issues regarding both bandwidth usage and energy-consumption related issues. For instance, in 2003/2004, demands for communications from/to Mars were at least 3 times higher than what current installations offer, hence impairing dramatically data transfers planned for the different missions involved. Furthermore, some inter-robot (sensor network) or robot-satellite communications are also required, which means that IPNs, by nature, no longer consist of plain point-to-point interplanetary links but consists in complex networks formed by heterogeneous nodes. Such networks are particularly challenging, considering that connectivity between nodes is usually transient due to orbital movements and planet rotations. As a result, communication with some parts of the network might only be possible during precise time periods.

Due to the growing number of research teams involved in such programs and

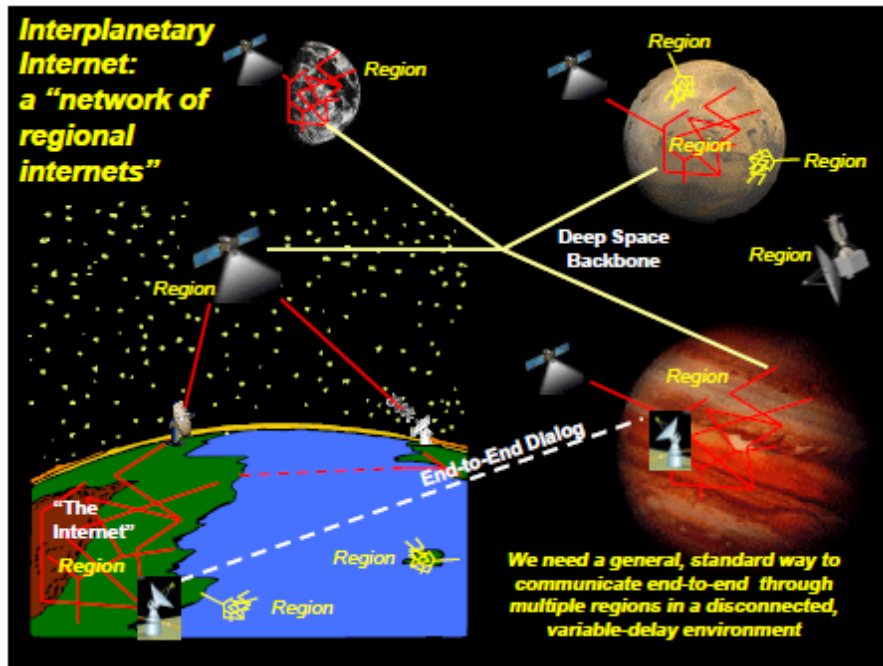


Figure 1.1: Interplanetary Internet - Future Mars Mission [Image Courtesy: NASA [19] ]

to the large amount of data collectable by space exploration devices, there has been and will be an increase in the number of communicating entities, in the services complexity and therefore in the bandwidth usage. Space missions exhibit communication needs shifting away from the pure telecommunication standpoint and therefore a full-fledged network stack should be designed to fulfill these requirements.

In the following Sections, we present in detail the context for interplanetary networking (IPN [5]), their research issues and our main contributions to this research domain.

## 1.1 Context

The Interplanetary Networks, is a conceived computer network in space, consisting of a set of network nodes which can communicate with each other. Communication would be greatly delayed by the great interplanetary distances, so the IPN needs a new set of protocols and technology that are tolerant to large delays and errors. While the Internet as we know, it tends to be a busy "network of networks" with high traffic, negligible delay and errors, and a wired backbone, on the other hand the Interplanetary Internet is a store-and-forward "network of Internets" that is often disconnected, has a

wireless backbone fraught with error-prone links and delays ranging to tens of minutes, even hours.

Space communication technology has steadily evolved from expensive, one-of-a-kind point-to-point architectures, to the re-use of technology on successive missions, to the development of standard protocols agreed upon by space agencies of many countries. This last phase has gone on since 1982 through the efforts of the Consultative Committee for Space Data Systems (CCSDS),[1] a body composed of the major space agencies of the world.

### 1.1.1 Why Not Terrestrial Protocols in IPN

One of the main aim of the these standardization bodies was to adapt the well developed terrestrial protocol (TCP/IP [90], FTP [76]) to the deep space communication network. The Internet as we use it now has been designed in the context where entities (end-hosts and routers) are mostly connected through wired links. Protocols and applications that were developed thirty years ago are still in use today. They rely on assumptions that we describe here:

- End-to-end connectivity: they assume that between any two nodes that can communicate with each other there is a continuous, bidirectional end-to-end path.
- Short round trip times: the assumed end-to-end connectivity, mainly relying upon wired links, leads to round trip times ranging from few milliseconds to a second.
- Symmetric data rates: data rates are assumed to be roughly symmetric. Even if access networks such as ADSL are asymmetric by nature, the level of asymmetry is suitable with regard to user needs. The up-link, which has the lowest capacity, has been sufficiently provisioned for most of the applications.
- Low error rates: transmission errors can occur for diverse reasons such as link failure or congestion, but are considered being unusual and they arise at a low rate.

Where else interplanetary network differ drastically from the terrestrial internet in a number ways that might arise individually or collectively:

- Intermittent connectivity: Connectivity may suffer from disruptions leading to link failure and network partitioning, for a large number of reasons like mobility issues, radio issues, and battery issues.

- Delay issues: Links could have a very high propagation delay or have such a highly variable delay that traditional protocol like TCP would fail.
- Asymmetric data rates: Links can suffer from highly asymmetric data rates or can be simply just unidirectional.
- High error rates: Some links may have high error rates. They could require a high level of correction and a large number of retransmissions, leading to the creation of tight bottlenecks.

The communication challenges posed by the interplanetary network dent the performance of these terrestrial protocols and opened the doors to the design of more effective protocol architectures.

In more detail, particular effort has been made by standardisation bodies such as the Consultative Committee for Space Data Systems (CCSDS) [1] and the Delay Tolerant Networking working group [3] within the Internet Research Task Force (IRTF) [4]. The former developed a full protocol stack (SCPS [7]) , alternative to the TCP/IP Suite, specifying protocol layers, from the application downwards to the physical, more appropriate to the deep space peculiarities. The latter has devised an overlay network architecture named Delay Tolerant Architecture (DTN [94]) , working over the transport layer and able to tolerate link disruptions and long delays, owing to the features offered by the Bundle Protocol [82].

## 1.2 Research Issues

In spite of the relevant efforts made by the scientific community, some research areas are still only partially explored. In more detail, some attention has to be drawn to the performance issues related to routing, and transport layers to be resolved. In this work we focus on scenarios related to interplanetary network (like Mars surface network) in which nodes are limited in resource and network connectivity is intermittent. In this context a number of networking issues need to be addressed:

- Transport: As we will see in the next chapter, it is widely accepted that nodes need to communicate in a store and forward fashion and that they need to have storage capabilities to carry data on behalf of the others. But a large number of open questions remain in this area. For instance, what kinds of acknowledgement, timer mechanism, flow control and congestion avoidance mechanisms should we use?

- **Resource Management:** As devices can have limited capacity in terms of batteries, computation power and storage, what kind of policies and strategies have to be defined for admission control and buffer management?.
- **Routing:** Being in an environment where nodes see each other intermittently, a complete path may never exist physically between a source and a destination. Routing is then challenging. Routing strategies that take into account the properties of such an environment are thus needed to ensure efficient delivery of messages.
- **Interoperability:** Interplanetary nodes may need to pass through Internet-linked networks and interact with legacy applications. Thus interoperability and security solutions are need for planetary devices and their protocols

In this work, we concentrated on performance issues by trying to understand properties of interplanetary nodes and of network environment that could serve protocols to make efficient routing or forwarding decisions. we have proposed some solutions for routing, transport and application protocols to improve their performance. The questions we address are the following: What kind of buffer management we need in a resource limited Rover?, Is it possible to have effective route selection based on environmental information?, Is it possible by the space application to select its underlying transport protocol according to its QoS requirement? To what extend?

### 1.3 Contribution

- **Buffer Management Policy for Mars Intelligent Proximity Network [69]** (Chapter 3) In order to meet the need for future Mars exploration missions, NASA [6] has come up with various proximity networks. These proximity network are relatively small, fairly short-range, often ad hoc, wireless network typically dedicated to tasks such as transporting in site sensing data. In this study we enlist the challenges in these proximity networks in general. In order to achieve data delivery in such challenging networking environments, researchers have proposed the use of store-carry-and-forward protocols which require good storage capacity for communication. However, the storage capacity in proximity network nodes (Mars rover for example) are unavoidably limited. Thus, efficient buffer management policies are necessary to decide which messages should be discarded, when node buffers are operated close to their capacity. In this study, we have proposed a buffer management policy in both real time and best effort traffic (image data) and compared

the performance of the proposed policy with other available policies and shown proposed policy is effective in given network.

- An Antenna Selection Algorithm for Mars Exploration Rover to Increase Data Return with Minimum Delay [67] (Chapter 4) Mars Exploration Rovers are one of most important entities in Interplanetary Network, their communication performance and data return ability affect considerably the performance of the whole system. To that extend, in this work, we have proposed an antenna selection algorithm for Mars Exploration Rover (MER), where MER can autonomously select which antenna to use between X-band Direct-To-Earth (DTE) and UHF for communication according to its constrains and limitation to effectively use their resources and thereby increase data return. When MER selects a path for communication, it simultaneously selects an antenna for transmission of data to earth. The proposed algorithm, defines a cost function for each possible path based on three parameters namely power required by the rover to transmit in that path, average delay, and error rate. A weighting is given to each parameter in the function based upon application Quality of Service (QoS) requirement. The algorithm has low computational complexity and provides optimum performance. This proposed algorithm make MER more autonomous in terms of communication system.
- Resource Aware Routing in Interplanetary Mars Surface Network [65] (Chapter 4) The networking and communication challenges posed by interplanetary environments make the design and the deployment of complex telecommunication infrastructures particularly difficult, especially with regard to routing and congestion control issues. To deal with communication challenges in Mars, an interplanetary overlay Network is proposed to establish a communication infrastructure among planets, natural and artificial satellites, and various mission elements such as spacecrafts and rovers. Mars surface nodes in such network have limited resources in terms of storage, energy, processing power. Thus in this work we have proposed an effective resource aware routing protocol for interplanetary Mars surface network, which uses knowledge about the connectivity and resource consumption of the nodes to make an effective routing decision to route all kind of traffics in IPN , where communicating entities are of different natures. Protocol provide better performance in terms of delivery and delay and makes mobile Rover nodes more autonomous.

- Dynamic NACK Timer Algorithm to Improve Delivery Latency of CCSDS File Delivery Protocol [66] (Chapter 5)

The Consultative Committee for Space Data Systems (CCSDS) File Delivery Protocol (CFDP [26]) is a new international standard developed by the space agency to meet the growing space communication need for effective transfer of information in a wide variety of mission configurations of inter-planetary space links. CFDP by principle work as traditional FTP with addition Automatic Repeat Request (ARQ) techniques suited for space environment. To make CFDP protocol as effective as FTP, various studies are carried on. To that extend, the performance of CFDP in the deferred NAK mode has been evaluated in a theoretical manner and found out that the delivery latency mainly depended on timer used in both sender and receiver. Thus to improve the performance we have proposed a dynamic timer algorithm for CFDP in deferred NACK mode, by which sender and receiver entities are able to set the timeout value for their timers dynamically according to the network conditions. Then algorithm have been implemented in Network Simulator (NS2 [62]) along with CFDP protocol itself and validated. Results furnished shows the performance improvement achieved by the proposed dynamic timer algorithm in file delivery latency in different network condition.

- QoS based Protocol Selection in Interplanetary Network (Chapter 6)

Recent developments in the design of deep space mission has give rise to numerous interplanetary applications, other than simple data transfer from a distant node. And the main focus of researchers is to provide communication infrastructure in a challenging environment. To that extend, CCSDS and DTNRG have developed protocol stack which perform optimal at the given environmental condition. They do however posse some limitations. In addition, neither of the CCSDS or DTN protocol stacks contains autonomy or flexibility. Interplanetary networks applications are designed in such a way that it can work under any lower layer protocols. To use effectively the protocol resource, CCSDS deployed protocol selection based on application layer knowledge. They proposed three transport protocols to respond to varying Quality of services. While the idea of multiple protocols in attempting to satisfy QoS levels is considered effective, the protocols deployed are not.

For this reason, we have indented here to combine both CCSDS and DTN protocol to provide more option in protocol selection to satisfy application QoS. And also make the selection process dynamic by taking in to account the environmental



---

condition during selection. To this extend, we have proposed a QoS framework which act as middleware reside between application and transport protocols, where the application layer select its transport protocol which has ability to achieve its QoS requirements.

## 1.4 Outline

This thesis is structured as follows. First, Chapter 2 proposes a state of the art in interplanetary networking by giving an overview of architectures, applications and protocols that have been studied or deployed. Then, Chapter 3, Chapter 4 , Chapter 5 and Chapter 6 present our contributions in the order they were introduced in previous section. Finally, Chapter 7 concludes this work by summarizing the contributions and discussing the future directions in which this work can be extended.

## Chapter 2

# State of the art

This chapter provides an overview of research works related to interplanetary networks, architecture, protocols and their standardization. The communication needs for future National Aeronautics and Space Administration (NASA) space mission have triggered the need for various research activities to address all of the communication issues of errors, delays, and intermittent links in interplanetary network. This made mission planners, technologists, system designers, engineers, and scientists from the space community to work together to provide feasible communication architecture. Thus, we first present network architectures in near & far term in interplanetary networks and their communication challenges in deep space environment.

Focus of these researches is mainly to answer, why not use the existing internet protocols into deep space to minimize cost and risk? Thus, we discuss here why TCP dose not suit in these kinds of networks and the need for new protocol standards. Finally present various protocol standards which are proposed as alternative and their limitations in the challenging environment.

### 2.1 Network Architectures

The developments in the space technologies are enabling the realization of deep space scientific missions such as Mars exploration. The vision of future space exploration includes missions to deep space that require communication among planets, moons, satellites, asteroids, robotic spacecrafts, and crewed vehicles. This vision involves in the design and development of next generation deep space networks, which is expected to provide Internet like service in the deep space planetary networks and defined as Interplanetary Networks [5]. And the vision is to transform deep space network into

interplanetary network [101]

In [19] [73], a different decomposition of interplanetary architecture was defined. However, the architecture is defined in terms of several types of communication sub-networks called architectural elements. These architectural elements are defined by NASA as:

1. Mars-Earth backbone network
2. Mars vehicle proximity networks
3. Local area networks (LANs) on-board the Mars vehicles

### 2.1.1 Mars-Earth backbone network

This architectural element provides the long haul data links directly between Mars vehicles [18] and the Earth as well as the Earth-based infrastructure elements. The remote assets include elements with long-haul capabilities (certain Mars surface vehicles, spacecraft in Mars orbit) and the in-space data relay network. The Earth-ground segments include the deep space network (DSN), NASA and other space agency Intranets and virtual private networks (VPN's), and the Internet. The critical technologies involve increasing the capacity and the autonomous operation of the remote long haul space link assets.

### 2.1.2 Mars vehicle proximity networks

Mars vehicle proximity networks - These involve the wireless links with distances that are relatively near to the planet. There are three types of proximity links.

1. Orbiter to/from Surface (called access network in [19]): This architectural element provides the data links between Mars surface/airborne vehicles and spacecraft in Mars orbit. The orbiters will typically contain the long haul links to Earth and will host a gateway to the backbone network to relay between the backbone elements and the mission spacecraft and/or vehicles, or will forward to another proximity vehicle.
2. Mars inter-spacecraft networks - This architectural element provides the data link between spacecrafts flying in formation, clusters, or constellations in the Mars vicinity. It also includes the communication interfaces between approaching/departing spacecraft. In particular, spacecraft in a communication constellation that intercommunicate to relay the data to Earth are included.

3. Mars surface networks (called proximity network in [19]): This architectural element provides the data links between surface vehicles (rovers, airplanes, aerobots, landers, and sensors) spread out in an ad hoc network.

### 2.1.3 Local area networks (LAN's) on-board the Mars vehicles

Local area networks (LAN's) on-board the Mars vehicles (included in the access network [19]): This architectural element interconnects the various modules of the vehicle through an internal LAN consisting of one or more types of serial or parallel interconnected busses.

## 2.2 Communication Architecture

As the interest in Mars exploration grows and new vehicles are deployed with more and better communications assets, the architecture will acquire greater capabilities [15]. The following sections, describe the potential near-term (out to 2010), mid-term (out to 2020) and far-term (beyond 2020) scenarios for the Mars network [73].

### 2.2.1 Mars Near-Term Communication Architecture

The near-term architecture for the Mars communication infrastructure is illustrated in Figure 2.1. This architecture provides moderately high-speed communication between Earth and Mars vehicles via the radio frequency (X-, Ka-band) equipment incorporated on each scientific orbiter sent to Mars as well as on-board the large Rovers. Each of these orbiting vehicles incorporates a moderately high data rate RF backbone link to the Earth-based DSN as well as a moderate data rate proximity link to other Mars surface vehicles. Many of the orbiters will also provide navigation information to the surface elements. In addition, the infrastructure will be augmented by a dedicated communications satellite (ASI Telesat) to be put in place to perform data relay functions for surface vehicles, such as the Scout-class that do not have the ability to communicate directly to Earth. As shown in the figure, low rate proximity networks will be employed between Mars vehicles, such as between Landers and rovers as necessary.

### 2.2.2 Mars Mid-Term Communication Architecture

The mid-term architecture for the Mars communication infrastructure is illustrated in Figure 2.2 includes the Robotic Outpost, representing the first permanent

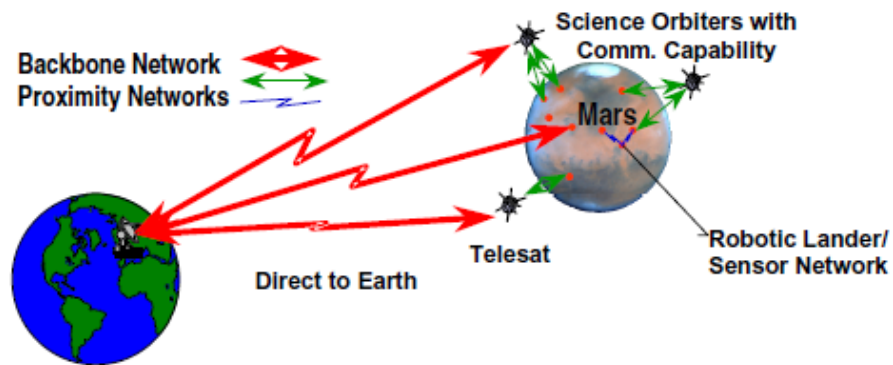


Figure 2.1: Mars Near-Term Communication Architecture [Image Courtesy: NASA [73]]

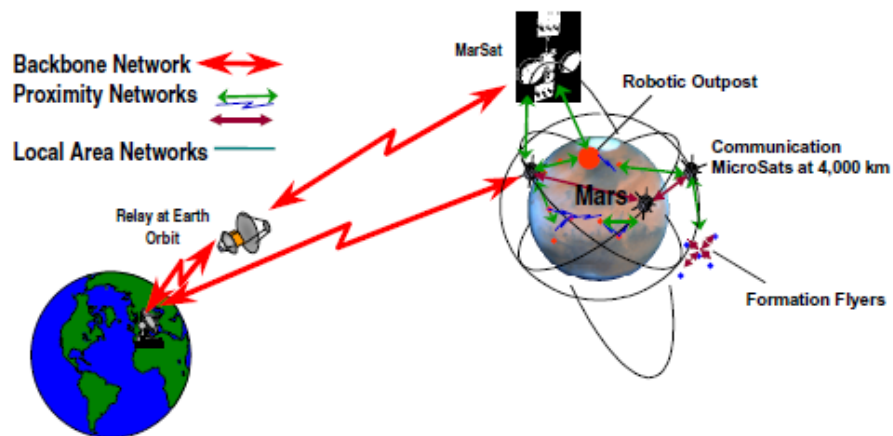


Figure 2.2: Mars Mid-Term Communication Architecture [Image Courtesy: NASA [73]]

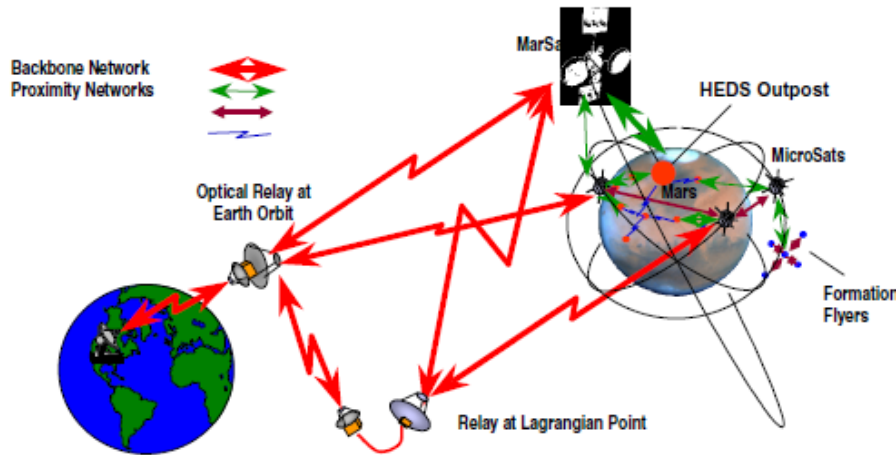


Figure 2.3: Mars Far-Term Communication Architecture [Image Courtesy: NASA [73]]

presence on Mars. Sensor networks will be used to monitor the local environment around the outpost. This architecture provides high speed communication between Earth and Mars vehicles via a micro-satellite constellation designed specifically for Mars communications and navigation purposes [29]. Each of these spacecraft incorporates a high data rate Ka-band package for the backbone connection to the Earth-based DSN as well as high data rate access links to Mars vehicles. One or more Mars communications satellites (MarSat) will be placed in Areosynchronous Mars Orbit (AMO) [89] and used to provide continuous coverage for the Microsats and ground vehicles in view.

It will support the high-rate communications to/from Earth. This MarSat would become an always available, high-speed link for the Robotic Outpost below it. In addition, there is likely to be one or more Earth orbiting relay stations that operate using high-speed RF or optical means to provide additional DSN capacity and availability.

### 2.2.3 Mars Far-Term Communication Architecture

The far-term architecture for the Mars communication infrastructure will be used in support of Human Exploration and Development of Space (HEDS [11]) missions and is illustrated in Figure 2.3. This architecture provides high availability, high-speed communication between Earth and the Mars Base and other Mars vehicles. The traffic patterns will drastically shift from the predominantly data-to-earth requirement to a high volume of two-way traffic, consisting of a wide variety of types (e.g., voice, video, data) to/from Earth to support the human presence. There will also be an increased need for local communications to handle the human generated local traffic. At this point

## **2.3. COMMUNICATION CHALLENGES IN SPACE ENVIRONMENT**<sup>14</sup>

---

in time, the Mars network will become the most important hub of the Interplanetary Network and handle data from other deep space missions.

In order to support communications at all times, relay stations placed at gravitationally stable Lagrangian points associated with the sun and the Earth may be employed to avoid blackouts during the periods that the sun is between Earth and Mars. These relay stations will also be backbone elements of the intra-solar system IPN communications network. Each of these relay stations may incorporate high data rate microwave and optical packages for the intra-backbone communications links. Multiple Mars communications satellites (MarSat) in Areosynchronous Mars Orbit (AMO) [89] will be used to handle the high data rate communications link from other communication assets and ground vehicles to Earth either directly or via the relay network.

### **2.3 Communication Challenges in Space Environment**

The unique characteristics of the communicating entities and links in Space scenario make the direct use of terrestrial protocol stack infeasible [32] [13]. This is due to the following constraints that are innate to this environment:

#### **2.3.1 High Signal Propagation Delays**

The enormous distances involved between the communicating entities and the relativistic constraint restricting signal transmissions to the speed of light, cause high signal propagation delays. The speed-of-light delay from Earth to Mars, for example, is approximately 4 minutes when Earth and Mars are at their closest approach [32]. The one-way light time can exceed 20 minutes when Earth and Mars are in opposition.

#### **2.3.2 High Data Corruption Rates**

Radio signal strength reduces proportionate to the square of the distance between the communicating entities; thus the large distances involved in the interplanetary communication environment cause the signal to be received at extremely low strengths. Further, this reception is subject to high probability of bit-errors in the channel due to random thermal noise, burst errors due to solar flares, etc.

#### **2.3.3 Meager, Asymmetric Bandwidth**

The bandwidth capacities are meager, and asymmetric in the deep-space environment. The uplink channel (Earth to Destination) tends to have much lower band-

width than the downlink channel (Destination to Earth), typically by one or two orders of magnitude. This is normally by design, due to signal power considerations and the fact that the uplink channel is expected to carry mostly light-weight command traffic while the more interesting data collection, analysis, reports etc. are expected in the downlink channel. Therefore the uplink channel tends to be designed as a reliable, lightweight communication channel. For example, the Cassini spacecraft has an uplink bandwidth of 1 Kbps while the maximum downlink bandwidth is 166 Kbps

#### **2.3.4 Blackouts or Intermittent Connectivity**

Since communicating entities in deep-space almost always are in motion relative to one another, the communication channels between them are intermittent. Periodic link outages may occur due to Orbital Obscuration; loss of line of sight because of moving planetary bodies, harsh environment (solar flares, sandstorm, meteoroid shower etc.), and interference of a spacecraft, ground station handovers, weather, changing network topology (wide fluctuation of round trip time, orbital dynamics), temporary or permanent link outages.

## **2.4 Characteristic of TCP over Mars-Earth backbone network**

Much of the study described here has its roots in a NASA research project to develop an interplanetary Internet, or interplanetary network (IPN). The main focus of various space agencies is to apply well known terrestrial protocols into the deep space network. For this reason, a study is made on most successful terrestrial protocol Transmission Control Protocol (TCP) [90] in interplanetary network. The basic idea is to try to make data communications between Earth and (very) remote spacecraft seem almost as easy as that between two people on different sides of the world. As it happens, before a network node can send any application data using TCP, a three-way handshake is required that consumes 1.5 round-trip times (RTTs). There is also a generic, two-minute timeout implemented in most TCP stacks: if no data is sent or received for two minutes, the connection breaks. Putting these facts together, we can see that once a spacecraft is more than a minute away (in terms of light-trip time), every attempt to establish a TCP connection will fail, and no application data will ever be transmitted. In the case of Mars, for example, at its closest approach to Earth, the RTT is roughly eight



minutes, with a worst-case RTT of approximately 40 minutes. Thus, normal variant terrestrial TCP cannot work at all for Earth-Mars communications.

The space environment in general offers a number of impediments to reliable data communications. This section, presents TCP behavior over challenges in deep space like, Error-Prone Links, Asymmetric Channels, Limited Link Capacity and Intermittent Connections.

### **2.4.1 TCP over Error-Prone Links**

In wireless communications in general, but especially in space communications, bit-errors caused by noise are not uncommon. To some degree, forward error correction [64] can compensate for such errors trading bandwidth for effective data rate, but the space link still is rarely as clean as those of modern terrestrial networks. TCP is designed to handle packet loss by identifying and retransmitting lost segments; however, TCP assumes the source of all packet loss is network congestion. Consequently, TCP invokes congestion control, reduces its congestion window, and in turn, its transmission rate as a result of any packet loss. TCPs congestion control algorithm works well in dealing with congestion-induced loss, but only results in reduced throughput on un-congested, noisy links without providing any benefits.

Various space standards SCPS-TP [87], TCP-Planet [12] are proposed with standard TCP augmented and with a set of extensions and enhancements. SCPS-TP [87] has two mechanisms for determining the source of packet loss. Like standard TCP, SCPS-TP makes a default assumption regarding the source of loss in the absence of any explicit information. TCPs default assumption is that all loss is caused by congestion; however, SCPS-TPs assume by default that any loss is due to errors. The second mechanism for identifying the source of loss is explicit signaling.

### **2.4.2 TCP over Asymmetric Channels**

Communications channels between spacecraft and the ground are frequently asymmetric in terms of both channel capacity and error characteristics. Often the forward link bandwidth (from the spacecraft to the ground) is substantially larger than the return link bandwidth, with ratios of 1000:1 not uncommon [32]. This asymmetry is a result of various engineering tradeoffs (such as power, mass, and volume), as well as the fact that for scientific missions, most of the data originates at the satellite and flows to the ground. The return link is generally used for commanding the spacecraft, not bulk

data transfer. While such high asymmetries are not as common in the wireless/cellular environments, reducing the required acknowledgment channel bandwidth is still a desirable goal since it reduces the energy emissions of a mobile unit when it is only receiving data.

Bandwidth asymmetry can limit TCPs throughput even when the data are flowing on the larger channel and the smaller channel is carrying only acknowledgments. A receiving TCP endpoint generally acknowledges every other segment, which dictates an acknowledgment channel capacity that is proportional to the data channel capacity and is a function of the segment size. With a 1024-byte segment size, TCP throughput is relatively unhindered by a forward-link to return-link bandwidth ratio of less than 50:1; however, throughput performance is limited by the acknowledgment channel capacity at higher ratios.

TP-Planet [12] uses a Blackout State procedure to deal with blackouts and the delayed SACK [57] strategy to deal with bandwidth asymmetry. More precisely, TP-Planet uses a rate-based Additive Increase Multiplicative Decrease (AIMD) congestion control, whose operation depends on the decision of the congestion detection mechanism. Deep-space communications, however, at least presently, operate with static, pre-scheduled management procedures, which are fixed long before the missions themselves. Therefore, congestion control is not really needed, since flow multiplexing over deep-space links does not exist. In that context, TP-Planet seems to be less-qualified for deep-space data transfers.

### **2.4.3 TCP over Limited Link Capacity**

Wireless communications channels tend to offer less available bandwidth than wired networks. In space arenas, this problem is coupled with the constraint that power is limited and bit-efficiency is important in terms of the cost of transmitting as well as in terms of link capacity. There is substantial bit-overhead associated with the TCP protocol, especially when using small segments to increase the probability of successfully transmitting a packet without incurring a bit-error. This overhead, at least 20 bytes of TCP header per packet, can consume a sizable share of a limited-bandwidth channel.

### **2.4.4 TCP over Intermittent Connections**

For IPN environment, connectivity on a given communications link is usually intermittent. Contact may be interrupted for a number of reasons, including ground

station handovers, changing network topology, antenna obscuration, weather, and orbital dynamics. When a satellite transitions from the visibility of one ground-station to another, the behavior is similar to a cellular hand-off between two base stations, although the handover time may be much longer.

Even short-term link outages pose a problem for TCP, resulting in poor throughput in the best case and aborted connections in the worst. In the absence of a steady flow of acknowledgments, TCP will invoke congestion control and repeatedly retransmit and back-off its retransmission timer. If connectivity is restored before TCP exceeds its maximum retransmission threshold, it will resume transmission where it left off. However, by the time the link is restored, the retransmission timer may be backed-off so far that minutes elapse before TCP recognizes it. If, on the other hand, TCP reaches its maximum retransmission threshold before connectivity is restored, the connection is aborted.

Another potential effect of a changing network topology is a wide fluctuation of measured round-trip delays due to increases or decreases in path hop-count and propagation distances. TCP can retransmit either too early or too late if its round trip time estimate is sufficiently far from the current actual round trip time. Premature retransmission timeouts result in unnecessary invocations of slow start, reducing throughput.

#### 2.4.5 Discussion

In summary, many mechanisms have been proposed, and studied to extend, or adapt TCP/IP for the near-space environment [68]. However, none of these mechanisms address the innate chattiness of TCP (taking 1.5 RTTs to finish the hand-shakes, for example), which becomes very expensive in deep-space channels where one-way signal propagation delays could be in the order of minutes and hours. Thus TCP variants with their current form are inefficient in interplanetary networks.

## 2.5 Protocol Standards in Deep space Missions

In this section, The current space/ground protocol stack used by the Consultative Committee for Space Data Systems (CCSDS) [1] is described in Section 2.5.1 while a proposed protocol suite for the InterPlaNetary Internet [79] is discussed in Section 2.6.

Since the beginning of the space era, NASA along with other nations' space agencies have been initiating organized efforts towards designing, testing and standardizing the conventions and protocols needed for space exploration. The Consultative

Committee for Space Data Systems (CCSDS) [1] is one of such cumulative effort and a successful result of collaboration. CCSDS File Delivery Protocol [26] acts as flexible and efficient transfer protocol for various data over space links. One of the final recommendations of the CCSDS comes in the form of Space Communications Protocol Standards (SCPS) [7]. SCPS act as a protocol suite designed to allow communication over challenging environments.

### 2.5.1 The Consultative Committee for Space Data Systems (CCSDS)

CCSDS is a multi-national forum for the discussion of common space communication issues. It was founded in 1982 by the most influential space agencies in the world. CCSDS acts as a global platform for leading space communications experts from 28 nations to discuss the most well-engineered space data handling standards. Its goal [1] is to enhance governmental and commercial interoperability and cross support, while also reducing risk, development time and project costs. More than 90 international space data standards have been developed; which have been used in more than 300 space missions and the number continues to grow. Figure 2.4 summarizes the space link protocols categorization into the five layers of the OSI model [107]. Note that the Session and Presentation Layers of the OSI model are rarely used over space links and so are absent in the figure. The CCSDS File Delivery Protocol (CFDP) [26] has been successfully used in some of the most recent missions including MErcury, Surface, Space, ENvironment, GEochemistry, and Ranging (MESSENGER) Mission [48] launched in summer of 2004 and the Deep Impact Mission [23] launched in Jan 2005 consisting of a larger Flyby spacecraft that carried Impactor spacecraft.

#### The CCSDS File Delivery Protocol (CFDP)

The CFDP is a File Transfer Protocol (FTP)-like protocol for transferring files between two entities separated by interplanetary distances [48], on the space links that have asymmetric bandwidths. As depicted in Figure 2.5, CFDP can be used over a wide range of underlying communication services. It can be configured for either ground station to spacecraft or spacecraft to ground station transfers, apart from other configurations of a network of platforms such as a spacecraft constellation or a series of planetary landers. The CFDP offers typical file operation commands such as delete, move, and copy that can be used to control a distant filestore e.g. the Solid State Recorder (SSR). The CFDP consists of core procedures and extended procedures. When a direct path is

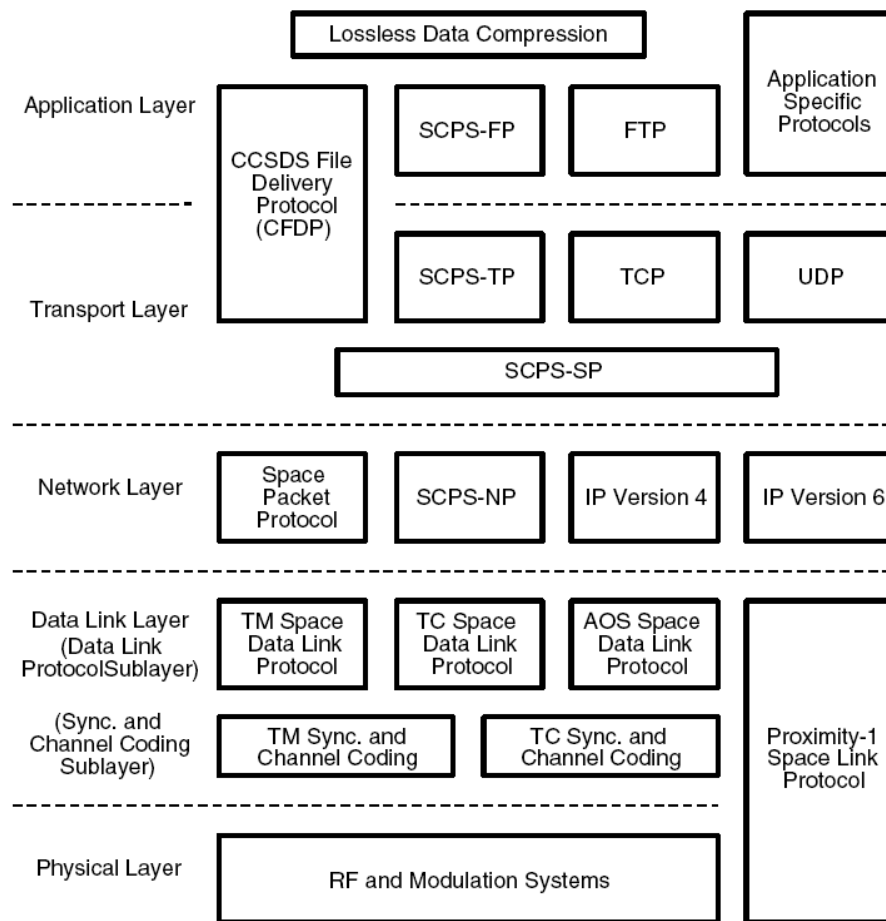


Figure 2.4: Space Link Protocols. [Image Courtesy: CCSDS [24]]

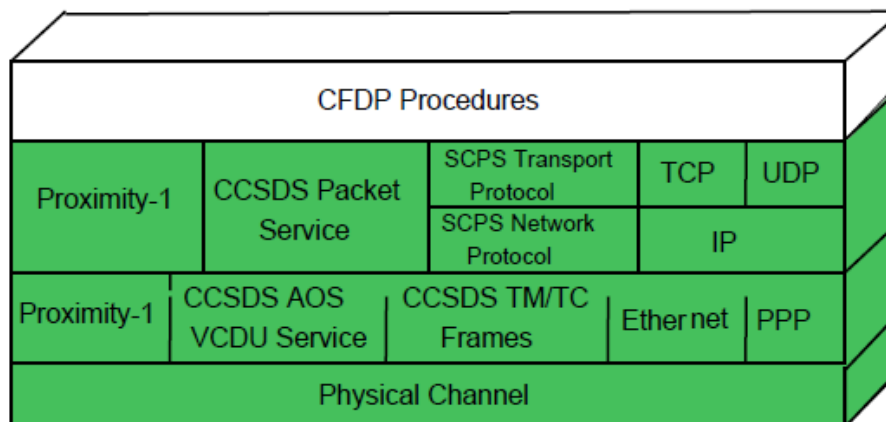


Figure 2.5: CFDP Operation [Image Courtesy: CCSDS [25]]

available between the two protocol entities, the core procedures are used to constitute the interaction. Extended procedures are used to automatically build an end-to-end file copy transaction when direct network connectivity between the source and the destination is impossible. The Extended procedures in turn use the Core procedures for an immediate next waypoint.

There are two modes of operation provided for CFDP viz. the acknowledged (reliable) mode and the unacknowledged mode. In first case the lost or corrupted data are automatically retransmitted. Several adopted measures present in CFDP's design enable robust operations of its ARQ system [70] in high-latency environments [79]. Proposed revisions to the CFDP Recommendation allow it to work in more complex scenarios e.g. where there might be multiple rovers and landers, multiple relay satellites, and perhaps even smaller devices that would relay files via rovers, etc. The store-and-forward model on which CFDP is built would enable it to be operated in more complex scenarios. Scott Burleigh, CCSDS working group chair and lead CFDP system engineer at NASA JPL in Pasadena, California commented, "The successes of CFDP on MESSENGER and the Deep Impact missions bring us closer to having an automatic interplanetary communication fabric that can support deep space science and exploration the way the Internet supports science on Earth." [48]. As shown in Figure 2.5, CFDP can be used over variety of underlying communication services, specifically including CCSDS packet services [25]. This includes the space datalink protocols such as the Advanced Orbiting Systems (AOS) [27] and the Proximity-1 Space Link protocol [28]. The AOS can be used over space-to-ground, ground-to-space, or space-to-space communications links by space missions. It enables exchange of video, audio, high-rate telemetry, low-rate transaction data, etc., through the space link in either direction due to provision of symmetric services and protocols for it. Thus the AOS Recommendation provides a space-adapted analog of the terrestrial concept of an "integrated services digital network" [27]. Its primary application areas are categorized as either observational science, experimental science, or the operation of the space vehicle ("core") systems. The Proximity-1 Space Link Protocol [28] is to be used [24] over short range, bidirectional or mobile radio links called as proximity links, generally used to communicate among fixed probes, landers, rovers, orbiting constellations, and orbiting relays. It contains recommendations for the Physical Layer of proximity space links.

20

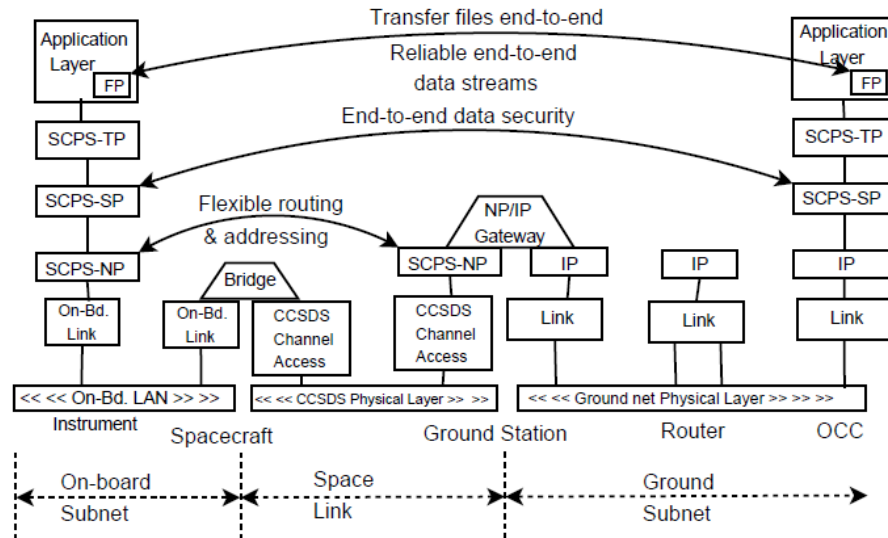


Figure 2.6: SCPS End-to-End Services. [Image Courtesy: SCPS [83]]

### 2.5.2 Space Communications Protocol Standards (SCPS)

The design of SCPS protocols is made by taking into consideration the need of interoperability across the spectrum of space missions, and between space data systems and the broader ground network environment [83]. The protocol data unit formats and options provided by SCPS protocols can be scaled to satisfy the communication needs of both complex and simple, resource-constrained missions. As the terrestrial protocols work on the assumptions that connection always exists which often does not hold in case while communicating with a remote spacecraft. The SCPS help to bridge the space and ground environments by supplementing current space link and ground protocols with end-to-end protocols, as depicted in Figure 2.6. Originally, the SCPS protocols were conceived for communication scenarios where spacecraft served as one endpoint of the communication. They also included requirements such as cross-linked satellite systems e.g. Brilliant Eyes, Teledesic, etc [72]. The SCPS suite provides following [83]

- SCPS-FP [84]: A file handling protocol that does the up-loading of spacecraft commands and software and the downloading of collection of telemetry data, optimally.
- SCPS-TP [87]: An underlying retransmission control protocol, that optimally provides reliable end-to-end delivery of spacecraft command and telemetry messages

between computers communicating over a network containing one or more potentially unreliable space data transmission paths.

- SCPS-SP [86]: A data protection protocol providing the end-to-end security and integrity of each message exchange
- SCPS-NP [85]: a scalable networking protocol that supports connectionless as well as connection-oriented routing of these messages through networks containing space data links

### 2.5.3 Towards Delay Tolerant Networking Architecture

As discussed in Section 2.5.1, CCSDS protocols are successfully used in most of the current deep space missions. The Deep Impact mission [78] is an excellent example of CFDP's usefulness in delivering the science data efficiently in most of the basic deployment scenarios in core architecture.

As mentioned in CFDP section, for more complex scenarios, revisions are proposed to the CFDP Recommendation [78]. Multiple elements of CFDP's architecture may in-fact require extension. Either the user applications (store-and-forward overlay) would have to be made more complex or the CFDP (the extended procedures) would have to bear the complexity. Furthermore, multiple UT layer implementations might be needed: e.g. Between Earth and orbiter CFDP communication, CCSDS Telemetry and Telecommand protocols might be utilized, but between a rover and an orbiter, CCSDS Proximity-1 protocol might be a better choice instead.

In these complex scenarios, the operation of CFDP is topologically similar to the operation of the Internet, except that the Internet's TCP/IP protocols suffer from large propagation delays and interruptions in end-to-end connectivity. The constraints that, the operation of CFDP over the Internet UT layer is quite similar as basic deployment of CFDP in space: the end-to-end route provided by IP functions as a single link from CFDP point of view. To overcome these limitations Delay tolerant networks architecture is proposed.

The following sections describe the architecture for the Delay and Disruption-Tolerant interoperable networking (DTN) environment [50] [103]. The architecture is developed taking into consideration the networks with occasional-connectivity that may suffer from frequent partitions and that may be functioning on more than one divergent set of protocols or protocol families. It is robust against disruptions and facilitates fire-and-forget networks and provides a new general-purpose network delivery service [79].



It is centered on a new end-to-end overlay protocol called Bundle Protocol [82]. The Delay Tolerant Network Research Group (DTNRG) [3], an IRTF working group [4] is a publicly accessible forum for discussions related to DTN architecture. There is separate effort being funded by Defense Advanced Research Projects Agency (DARPA) [2] in the form of Disruption Tolerant Networking program. The inception of DTN architecture lies in the work related to InterPlaNetary Internet [37] that focused primarily on the deep space communication environment.

### 2.6 Delay and Disruption-Tolerant Networking (DTN)

The Delay and Disruption Tolerant Networking (DTN) architecture [50] [94] [38] was conceived as a derivative of the IPN project [5]. The interplanetary networking environment is a specialized example of various operational environments for which DTN architecture could be utilized. The DTN architecture also applies to other networks including sensor-based networks [100] [58] using scheduled intermittent connectivity, terrestrial wireless networks that cannot ordinarily maintain end-to-end connectivity, satellite networks [22] with moderate delays and periodic connectivity, underwater acoustic networks [14] with moderate delays and frequent interruptions due to environmental factors and also Airborne Network [44] and Small Satellite Networks [77]. Its design methodology relaxes most of the assumptions with which the Internet Protocols work (as discussed in Section 1.1). The design principles can be summarized as follows:

- using variable-length messages that are possibly long as opposed to streams or limited-sized packets as the communication abstraction that could optimize the path selection decisions when possible
- using a naming syntax that can support wide range of naming and addressing conventions to improve the interoperability
- making networks capable of storing messages for a better period of time to support store-and-forward operation over multiple paths and over potentially long timescales. This would also take the end-to-end reliability requirement out from the architecture
- providing the security mechanisms that protect the infrastructure from unauthorized use

### 2.6.1 DTN Architecture

The DTN architecture [94] defines an end-to-end message-oriented overlay called the "Bundle Layer". The bundle layer can run over variety of transport and lower layer protocols. The devices implementing the bundle layer are called DTN nodes. DTN nodes are identified by one or more Endpoint Identifiers (EIDs). Each Endpoint ID is expressed syntactically as a Uniform Resource Identifier (URI) [75]. the EIDs are comprised of two parts : scheme and a scheme specific part(SSP). The interpretation of the SSP of an EID takes place relatively late in the delivery process of a message and is referred to as Late Binding.

The bundle layer provides a mechanism for persistent storage to combat the network interruption. It includes a hop-by-hop transfer of reliable delivery responsibility and optional end-to-end acknowledgment. The DTN-enabled applications can send messages called Application Data Units. These messages are transformed into bundle protocol data units called Bundles. This virtual message switching using Store-and-Forward operation works quite similar to postal service model.

Guided by the principle of postal communications, the DTN architecture offers three relative levels of delivery priority. The priority classes typically imply some relative scheduling prioritization among bundles in queue [94]:

- Bulk - This signifies the least effort delivery priority. Bulk bundles are shipped at the last as compared to other classes.
- Normal - Normal bundles are shipped prior to bulk class bundles.
- Expedited - Expedited bundles are given the highest preference and are shipped prior to Normal and Bulk bundles.

The priority levels also play role while discarding bundles as part of congestion control system The bundle layer provides six Classes of service (CoS) for a given bundle:

- Custody Transfer [34]: DTNs support node-to-node retransmission of lost or corrupt data by means of Custody Transfers [35]. The sending node can delegate the retransmission responsibility of the given bundle to an accepting node. This helps the sender node recover its retransmission resources. The accepting node replies the previous custodian node with a custodial-acceptance acknowledgment.
- Return Receipt: This act as a confirmation to the sending node that the bundle has been received by the implied destination application.

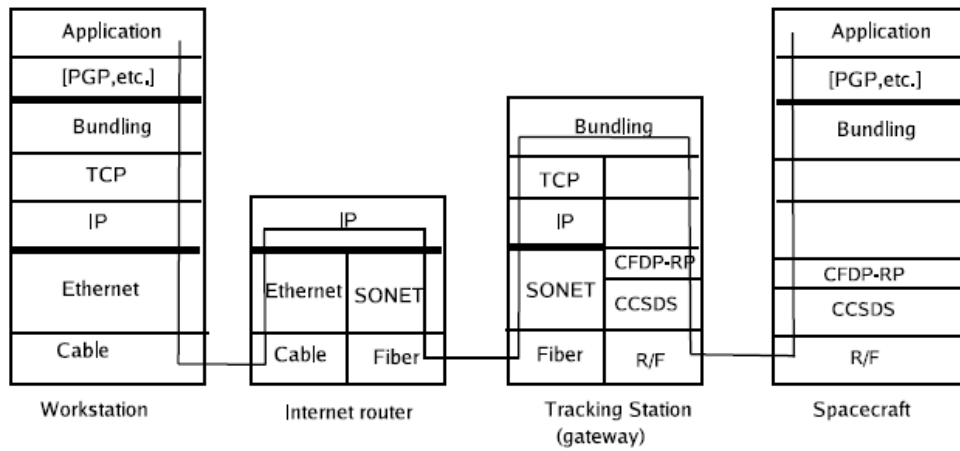


Figure 2.7: An Example of Data Flow in a Bundling-based Interplanetary Internet. [Image Courtesy: JPL/ NASA [79]]

- Custody-Transfer Notification: When a node accepts a custody transfer of the bundle, a notification is sent to the source.
- Bundle-Forwarding Notification: When the bundle is forwarded to another node, the current node sends a notification to the source.
- Priority of Delivery: As discussed above there are three priority levels viz. Bulk, Normal or Expedited.
- Authentication: It encompasses the methods such as digital signature that can be used to verify the sender’s identity and the integrity of the message.

Figure 2.7 shows an example of how the data flow between a workstation and a spacecraft in a bundling-based Interplanetary Internet might look [79].

### 2.6.2 Bundle Protocol (BP)

The Bundle Protocol [82] provides effective mechanisms for communicating in and/or through highly stressed environments such as those with intermittent connectivity, large and/or variable delays, and high bit error rates. For providing its services, BP forms a store-and-forward overlay network by sitting at the application layer<sup>2</sup> of some number of constituent internets. The protocol data unit is called as a bundle and it comprises a sequence of two or more blocks of protocol data, which serve various purposes. These protocol data blocks come in the form of primary bundle header and extension headers. The bundle priority can be specified with the Class of Service Flags (COS)

## 2.6. DELAY AND DISRUPTION-TOLERANT NETWORKING (DTN) 27

---

octet in the primary bundle header. The COS flags consists of two bits that represent the default priority classes followed by six bits that are reserved for future. The two bit priority field indicates the bundle's priority, the higher values being of higher priority: 00 = Bulk, 01 = Normal, 10 = Expedited, 11 being reserved for future use. The bundle carries the application data in the form of bundle payload. Following are some of the important concepts of the bundle protocol [82]:

- Bundle node: A bundle node is any entity that can send and/or receive bundles.
- Bundle protocol agent (BPA): It's a node component that offers the BP services and executes the procedures of the Bundle Protocol.
- Convergence layer adapters (CLA): The interface between the common bundle protocol and a specific internetwork protocol suite is termed a Convergence Layer Adapter. A CLA does the actual job of sending and receiving bundles on behalf of the BPA by utilizing the services of some native internet protocol that is supported in one of the internet's within which the node is functionally located.
- Application agent (AA): It's a node component that utilizes the BP services. The two components of AA are viz. an administrative element and application specific element. The application-specific element acts as a liaison between the application and BPA and works in constructing bundles and extracting application-specific application data units. The administrative element helps for the flow of control information between the application and BPA.
- Bundle endpoint: A bundle endpoint is a set of zero or more bundle nodes that all identify themselves for BP purposes by a single text string, called a bundle endpoint ID.
- Registration: A registration is the state machine characterizing a given node's membership in a given endpoint. A given registration can either be in active or passive states.

The current bundling architecture is shown in figure 2.8. Various research are currently being carried out to study different aspects related to bundling such as security mechanisms [92], routing [74] [91] [105], reliability mechanism [51] and Transport [40]. Key differences between Bundle protocol and CFDP are as follows [9]:

- The Bundle protocol is not limited to just support file transfer. Its design allows handling of virtually any end-to-end space application. Eventually, CFDP will

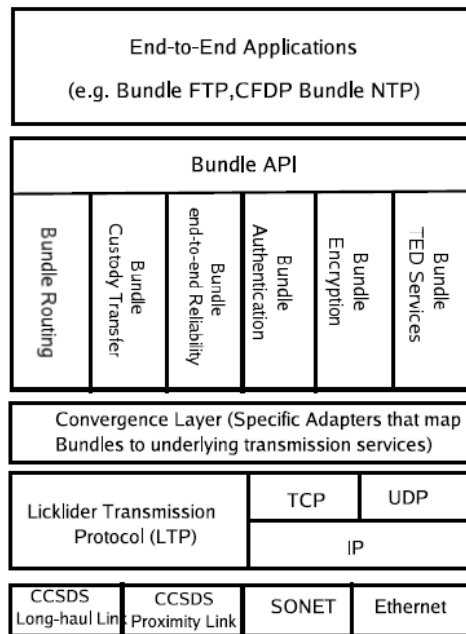


Figure 2.8: Current Bundling Architecture. [Image Courtesy: JPL/ NASA [82]]

be moved up the stack and would become one of those applications. A rich and comprehensive set of application services will be provided by the bundle protocol, including a more mature custodial transfer capability than is achievable with CFDP.

- The Bundle protocol is more concretely designed in terms of layering principles as compared to CFDP; which in turn would help it to evolve over time.

### 2.6.3 Licklider Transmission Protocol (LTP)

The Licklider Transmission Protocol [80], a.k.a. Long-haul Transmission Protocol [59], is principally aimed at supporting the links, characterized by extremely long message route-trip times and/or frequent interruptions in connectivity. It has emerged as one of the best candidates as a convergence layer protocol in the case of interplanetary space communication. For single hop deep-space RF links, LTP is intended to serve as a retransmission-based reliable mechanism and does ARQ of data transmissions by soliciting selective-acknowledgment reception reports. It's stateful, and has no negotiation or handshakes. LTP's design notions are directly descended from the retransmission procedures defined for CFDP.

LTP is defined in three Internet drafts that are close to getting published as

## 2.6. DELAY AND DISRUPTION-TOLERANT NETWORKING (DTN) 29

---

RFCs. Some of the key features of LTP can be summarized as follows [80] :

- **Massive State Retention:** Being a reliable transport service employing ARQ, LTP maintains the state information. All portions of the given data block are retained for possible retransmission. This further implies that the LTP implementations should consider non-volatile storage of some kind for the data retention to safeguard against very long delays, long communication disruption and brief communication opportunities.
- **Absence of Negotiation:** The deep-space communication demand minimum chattiness from the protocols to be used due to long round-trip times involved.
- **Partial Reliability:** For non critical data, LTP provides a best-effort mechanism in the form of green-part and red-part transmission. In this mechanism, LTP allows the upper layers to get any of their header and the meta-data transmitted reliably, i.e. being subject to acknowledgment and retransmission as per the necessity, in the form of red-part and have the actual data transmitted unreliably as green-part.
- **Laconic Acknowledgment:** The LTP sessions are unidirectional, which implies that there are no piggybacked acknowledgments on data segments. The LTP data acknowledgments - reception reports are carried in a separate segment type and are sent infrequently with comprehensive report of all data received within some specified range of offsets from the start of the transmitted block. This mainly saves the consumption of low and asymmetric transmission bandwidth.
- **Adjacency:** LTP is designed for providing point-to-point reliability on a single link. Traffic between two LTP engines is expected not to traverse any intermediate relays unlike TCP. In deep space communication, the point-to-point communication is simply exchange of radiation between two mutually visible antennae. LTP doesn't expect any underlying network-layer protocol; the underlying communication service is assumed to be a point-to-point link-layer protocol such as CCSDS Telemetry/Telecommand or Peer-Peer protocol for terrestrial applications. It's expected that the link-layer frames delivered to LTP arrive in order in which they were transmitted, though possibly with any number of gaps due to data loss or corruption.
- **Deferred Transmission:** LTP gets notifications about availability (or otherwise) of underlying communication service through link state cues. In case of communication interruption, LTP can suspend the ongoing sessions so to be invoked at later

stage when the link becomes up. This is as opposed to continuous duplex communication in case of TCP, which simply does not work in case of link unavailability.

Although LTP is being designed mainly taking the IPN environment into consideration, it does have applications in other environments as well.

## 2.7 Discussion

In this chapter, we have identified what exactly is interplanetary network and explored various proposed network and communication architecture for interplanetary network. As a first step we have analyzed various limitation posed by the interplanetary environment which directly affect the performance of the communication protocols especially TCP. Second, we studied the limitation of various variant of TCP protocol over interplanetary network and the need for new protocol stacks. Thirdly, we have explored various space protocol standards proposed by space standardization bodies like CCSDS and their challenge in providing required QoS. Finally we explored newly developed protocol family called "Delay Tolerant Network, which has been developed as alternative to exciting space protocols.

From this exploration study, we have identified various performance limitation pose by the exciting form of the protocols and defined a problem statement for our thesis and provided some effective solution and recommendation for selected protocols to achieve good performance, in the following Chapters.

## Chapter 3

# Performance of DTN protocols in Interplanetary Network

In the last few years, Delay- and Disruption-Tolerant Networking (DTN) have grown from relatively obscure research activities to a healthy research topic attracting both network designers and application developers. DTN is now a recognized area in networking research, due in part to practical experiences with planetary networks that are required to operate in situations where continuous end-to-end connectivity may not be possible.

While the architectural principles for DTN were synthesized and collected together about a half-decade ago, only recently have these principles been reviewed by a larger community and put to the test in a number of real-world pilots. With renewed interest in network architecture research, it appears timely to examine the DTN architecture retrospectively, highlighting some of its more unusual and controversial aspects, with the goal of providing concrete suggestions for capabilities applicable to network architectures that might be considered in evolving the DTN architecture or other networking architectures.

In this chapter we review many of the principles of the DTN architecture [50], highlighting performance of DTN protocols over Planetary Networks and their limitations. As a first step, we analyze the performance of those routing protocols which seems adaptable for deep-space network, aiming to study the impact resource limitations (buffer) over the performance of the protocols.



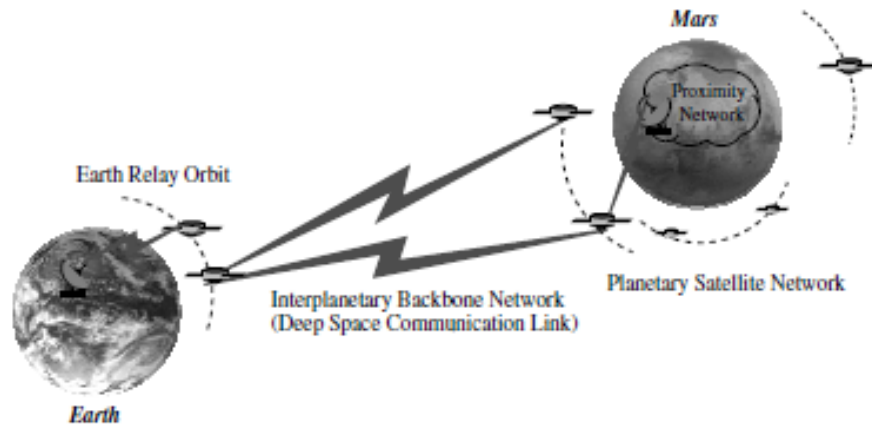


Figure 3.1: Interplanetary Architecture [Image Courtesy: NASA [19] ]

### 3.1 Deep Space Network Architecture

A typical deep-space network architecture shown in Figure 3.1 is proposed for the Mars Exploration mission [19]. In this section we describe deep space network architectural elements and define the reference scenario considered in the study. Architectural elements of the proposed infrastructure can be summarized as

- IPN Backbone Network
- Planetary Satellite Network
- Planetary Proximity Network

#### 3.1.1 IPN Backbone Network

Interplanetary Backbone Network includes the direct link or multihop paths between the outer-space planets and the Earth, as well as the Earth-based infrastructure elements such as a ground station for the deep-space network. One of the main focus will be on IPN backbone network, where the source and sink end-points are basically the ground station at the Earth and the planetary gateways connected to the relay satellites orbiting around the outer-space planets as shown in Figure 3.1. This is because the IPN backbone network plays a significant role for the performance of entire deep-space communication. The most important characteristics and the challenges posed by the IPN backbone network are Very long propagation delays, High link error rates, Blackouts, Bandwidth asymmetry.

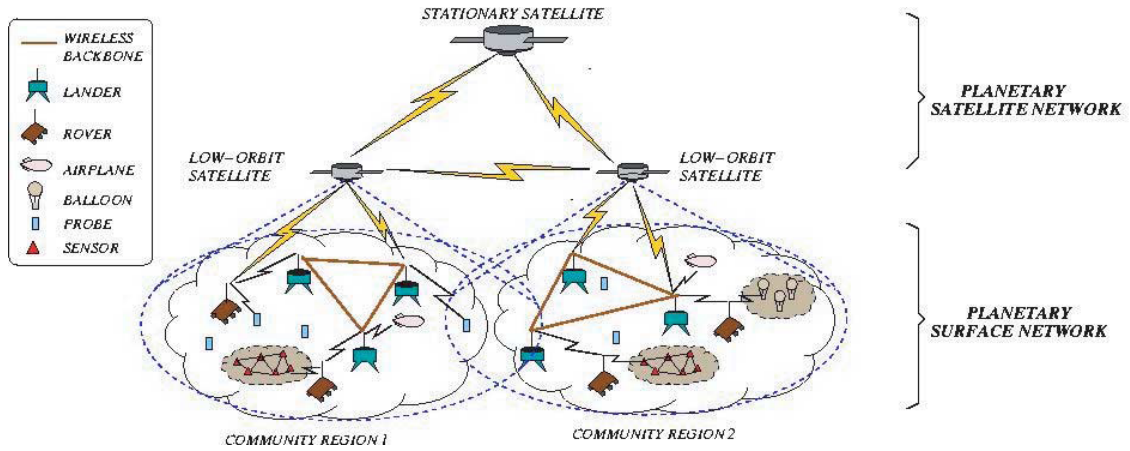


Figure 3.2: Planetary Network Architecture [Image Courtesy: NASA [19] ]

### 3.1.2 Planetary Network

Planetary network comprises of planetary satellite network and proximity surface network as shown in Figure 3.2. Planetary satellite network consists of the satellites orbiting the planets to provide communication relay and navigation services to the surface elements. On the other hand, NASA Proximity networks are relatively small, fairly short-range, often ad hoc, wireless networks typically dedicated to tasks such as transporting in situ sensing data. The number of nodes contained within a proximity network is expected to be comparatively small, perhaps tens or hundreds of nodes at most. While "short-range" is relative, many proximity networks will have a physical diameter on the order of hundreds or thousands of meters (although some authorities have suggested that a few of these networks might be as large 100-400 km [47])

Planetary surface proximity networks including surface elements, such as rovers and landers are often organized in clusters and spread out in an ad hoc manner, e.g., sensor nodes and balloons. Hence, the routing problem in planetary surface network is relevant and similar to the terrestrial mobile intermitted ad hoc networks and sensor networks; hence some of these emerging terrestrial technologies can be applied to the planetary surface network.

### 3.1.3 Reference Scenario

The scenario considered in this work as depicted in Figure 7.2 is composed of two main portions: planetary and backbone regions. In more detail, each planetary region comprehends planetary nodes (white circles) working as both traffic source and

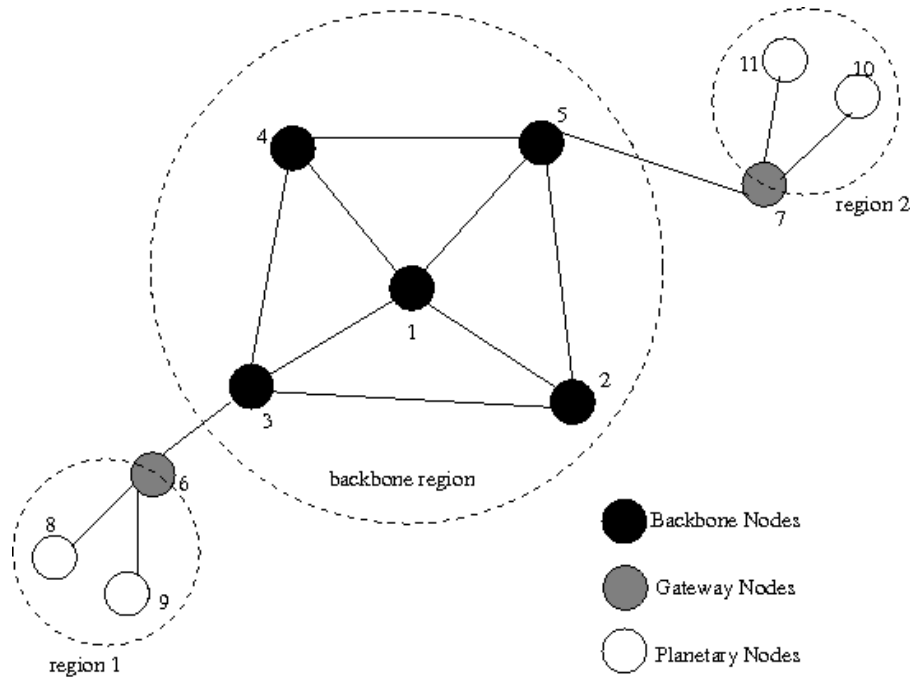


Figure 3.3: The Reference Scenario

destination nodes. The backbone region is composed of interplanetary nodes (black circles), serving as relay nodes, connected one with another through a star topology. Finally, specialized gateway nodes (grey nodes), responsible for data forwarding, connect the planetary regions.

The Figure 7.2 reports the case of two planetary region and one backbone region. Where planetary region composed of two planetary nodes which can be either Lander or Rover who work as traffic sources and destination, where Landers are fixed base station node and Rovers are mobile with non deterministic random movement. Planetary nodes are connected to backbone region by gateway nodes which are the Mars orbiters which orbit around Mars in a defined orbital movement. And, we have backbone region composed of backbone nodes which are Earth ground stations and Earth satellites. Earth ground station work both as traffic source and destination. The Figure 7.2 shows Node 1 as Earth ground stations; Nodes 2-5 are earth relay satellites in backbone region; Nodes 6,7 are Mars relay orbits (gateway nodes); Finally Nodes 8-11 are planetary nodes which can both send and receive data.

## 3.2 DTN routing protocols performance

The subject of DTN routing has almost become an independent research area. There have been more than a dozen different routing schemes proposed [74] [91] [43], a few PhD theses [54] [106], and a number of papers [42] mostly simulation studies that explore the particular features of one or more algorithms. This is perceived as a healthy situation, and was anticipated during the development of the DTN architecture. A survey of such schemes appears in a useful review by Zhang [105].

The DTN architecture [79] claims applicability to a wide range of operating environments, and was therefore intended to support pluralism between the naming formats, routing algorithms, and network technology. The routing problem can be coarsely divided into whether the routing graph is assumed to be connected or not, with DTN typically aiming at the latter. In addition, methods for routing bundles may involve creation and deletion of single or multiple copies of a bundle, various degrees of knowledge about the topology and traffic pattern (e.g., past, current, and future contact, traffic load, and buffer occupancy), fragmentation, various levels of granularity in decision making, resource reservations, different routing for different class of service or custody bundles, and different options for the loci of the routing computation (e.g., at the edges with source-route forwarding versus routing computations at each node).

### 3.2.1 DTN Routing

In [105] author review some of the recently developed routing protocols in DTN and categorize them in to two:

- Deterministic Routing [91] : Protocols assume that future movement and connection are completely known (that is, the entire network topology is known ahead of time)
- Stochastic or Dynamic Routing [95] [56] [20] : Protocols work under condition where the network behavior is random and not known. These protocols depend on decisions regarding where and when to forward messages. The simplest decision is to forward to any contact within range, while other decisions are based on history data, mobility patterns or other information

There are various studies conducted on the performance of these protocols [55] over Ad-hoc network [60], intermitted sensor network [104], and space networks [30]. In our study we intended to analyze the performance of few well documented protocols over

interplanetary network (reference scenario) over several constrains like power and buffer resource.

### Forwarding Strategy in Dynamic Routing

Suppose a source node generates an original data packet that is delivered to its destination node. When a relay node without its packet copy encounters (establish a radio communication) the source node, the relay node accepts a packet copy (i.e., it is infected) with probability  $q$  ( $0 \leq q \leq 1$ ). Further when a relay node without a packet copy encounters an already infected relay node (with the packet), it accepts a packet copy (i.e., it is infected) with probability  $p$  ( $0 \leq p \leq 1$ ). When the destination node encounters the source node or one of infected nodes for the first time, it accepts the packet with probability 1.

Various authors have proposed different set of routing protocols based on different forwarding strategies; where

- Direct Source-Destination Delivery ( $p = q = 0$ ) [105]; The source node directly delivers a packet to the destination node, without any help of relay nodes.
- Two-Hop Forwarding ( $p = 0, q = 1$ ) [105]; Each relay node accepts a packet copy only from the source node. Thus the packet will reach the destination node with two hops at most.
- Probabilistic Forwarding ( $0 < p = q < 1$ ) [56]; [105]; Each node forwards a packet copy to a relay node with probability  $p$ .
- The conventional Epidemic Routing ( $p = q = 1$ ) [91]; [105]; Each node forwards a packet copy to a relay node with probability 1.

In this study we have analyzed performance of the protocols which falls in the above forwarding strategies, namely simple Epidemic Routing [95] which was the first work in DTN routing; Well drafted PROPHET routing protocol [56]; and Minimum Estimated Expected Delay [91] that uses topology information for routing decisions.

### Epidemic Routing

Epidemic routing [95] supports the eventual delivery of messages to arbitrary destinations with minimal assumptions regarding the underlying topology and connectivity of the underlying network. In fact, only periodic pair-wise connectivity is required

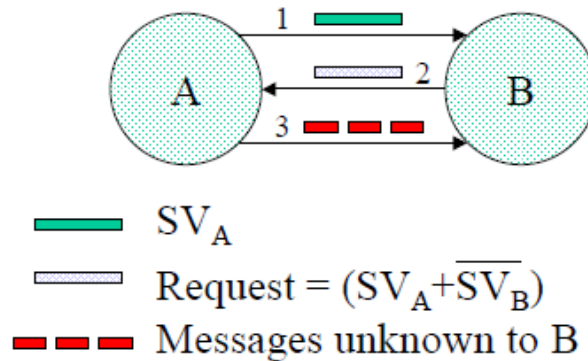


Figure 3.4: Message Exchange in the Epidemic Routing protocol [Image Courtesy: [95]]

to ensure eventual message delivery. The protocol relies upon the transitive distribution of messages through the networks, with messages eventually reaching their destination. Figure 3.4 depicts the message exchange in the Epidemic Routing protocol. Host A comes into contact with Host B and initiates an anti-entropy session (During anti-entropy, the two hosts exchange their summary vectors to determine which messages stored remotely have not been seen by the local host). In step one, A transmits its summary vector,  $SV_A$  to B.  $SV_A$  is a compact representation of all the messages being buffered at A. Next, B performs a logical AND operation between the negation of its summary vector,  $\overline{SV_B}$ , (the negation of this summary vector, representing the messages that it needs) and  $SV_A$ . That is, B determines the set difference between the messages buffered at A and the messages buffered locally at B. It then transmits a vector requesting these messages from A. In step three, A transmits the requested messages to B. This process is repeated transitively when B comes into contact with a new neighbor. Given sufficient buffer space and time, these anti-entropy sessions guarantee eventual message delivery through such pair-wise message exchange.

Packet level simulator called The Network Simulator (ns-2) is used for all simulation study. In order to analysis epidemic routing, we have adopted the source code implantation of epidemic routing developed in Duke computer science school [95], where the Epidemic Routing protocol is implemented using the Monarch [71] extensions to the ns-2.

## PROPHET

Another well drafter DTN routing protocol is Probabilistic Routing Protocol using History of Encounters and Transitivity (PROPHET) [56]. PROPHET first estimates a probabilistic metric called delivery predictability,  $P(a, b)$ , at every node  $a$ , for each known destination  $b$ . This indicates how likely it is that this node will be able to deliver a message to that destination. The operation of PROPHET is similar to that in Epidemic Routing [95]. When two nodes meet, they exchange summary vectors, and also a delivery predictability vector containing the delivery predictability information for destinations known by the nodes. The summary vectors are obtained in the same way as in [95] (where the vector is called a list). This additional information is used to update the internal delivery predictability vector as follows:

$$P_{(a,b)} = \begin{cases} P_{(a,b)_{old}} + (1 - P_{(a,b)_{old}})P_{init} & \text{if a b meets} \\ P_{(a,b)_{old}}X\gamma^k & \text{otherwise} \end{cases}$$

where  $P_{init}$  in  $(0,1)$  is an initialization constant (with all  $P(a, b)$  being set at  $P_{init}$ ) and  $\gamma$  in  $(0,1)$  is an aging constant. The information in the summary vector is used to decide which messages to request from the other node. To evaluate the protocol, we have implemented abstract PROPHET protocol in ns-2 which incorporates delivery prediction calculation and forwarding strategy using the source code developed by authors of the protocol.

## Minimum Estimated Expected Delay (MEED)

Jain et al. present in [91] a routing framework that takes advantage of increasing levels of information on topology, queue state and traffic demand. Four knowledge oracles are defined in Figure 3.5. The contacts summary oracle provides time-invariant aggregate or summary statistics on inter-node contacts, such as average waiting time until a next contact. The contact oracle provides full information for all contacts, such as start time and duration, enough to build a time-varying contact multigraph. The queuing oracle answers for the instantaneous queue state and current waiting times at all nodes. The traffic demand oracle gives information on any present and future messages injected in the network.

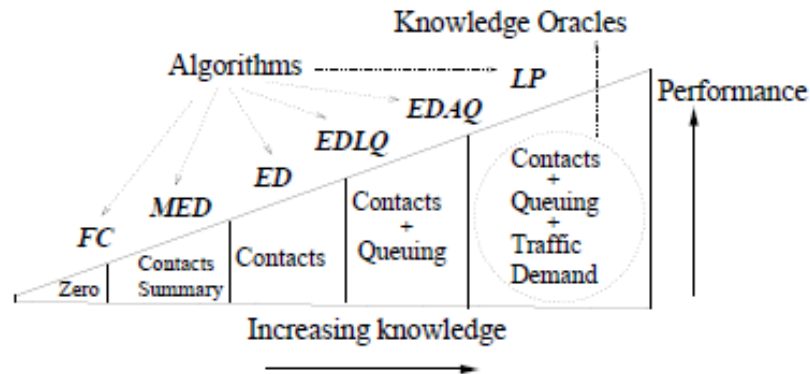


Figure 3.5: Conceptual performance vs knowledge trade-off [Image Courtesy: [91] ]

### 3.3 DTN bundle protocols performance Comparisons

In this section, the performance analysis of the selected DTN routing protocols over reference scenario (Figure 7.2) is presented. All simulations are conducted using Network Simulator NS2 version 2.29. As the nodes in interplanetary network are storage limited, all the performance analysis are aimed to evaluate the impact of buffer size of the nodes over the protocol performance. Though various research works [55] [74] have already done on various network topology like Intermittent Ad hoc Network, Mobile Sensor Network and Satellite Network, our intention here to understand the working of DTN based protocol over the resource limited planetary nodes in the interplanetary network scenario, from which future tuning of the protocol can be established (like, a buffer management, resource aware decisions etc.) to improve performance.

#### Performance Metrics

As DTN routing protocols works in store and forward paradigm, we have considered *Packet delivery ratio* as performance metric in our analysis. Data packet delivery ratio is defined as the number of packets received by the destination node divided by the number of data packets transmitted by the source node. The performance metric is evaluated verses three different parameters namely buffer size, packet size and link availability.

- Buffer size = Size of the buffer of all intermediate nodes
- Packet size = Size of the data packet transmitted by the source node
- Link Availability = Defined as up/down time of the links



### 3.3. DTN BUNDLE PROTOCOLS PERFORMANCE COMPARISONS40

Table 3.1: Simulation Parameters

Simulation Parameters	Values
Application rate	2 pks/sec
Up/Down link %	75/25 %
No of Nodes	11
Packet Size	100 KB
Buffer Size	1000 (pks)
Transport Protocol	UDP
Simulation Time	2500sec

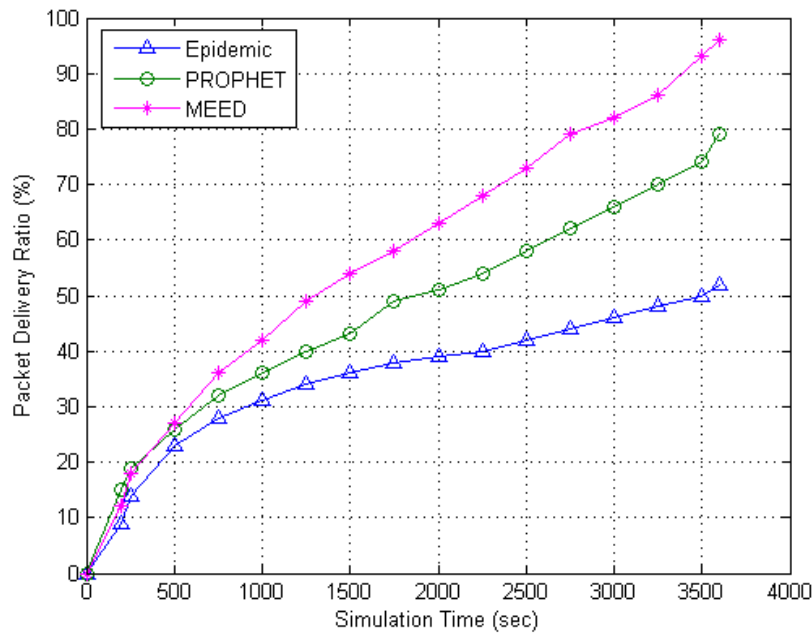


Figure 3.6: Packet delivery ratio over Simulation time

#### 3.3.1 Packet Delivery Ratio

In the first set of simulation, we have measure the transient packet delivery performance of the routing protocols over finite time horizon. In Figure 3.6, the packet delivery ratio percentage is plotted, using the default parameters listed in Table 3.1. Since we are interested in the behavior of the protocol over only a finite time horizon, the initial conditions can have a large impact on the performance. To counter the transient nature, we have generated 10 i.i.d replicates of PDR by running independent simulation of the system under study. In each simulation uses same initial conditions, and average of each PDR is plotted in the graph.

As shown in Figure 3.6, MEED routing protocol achieve good packet delivery

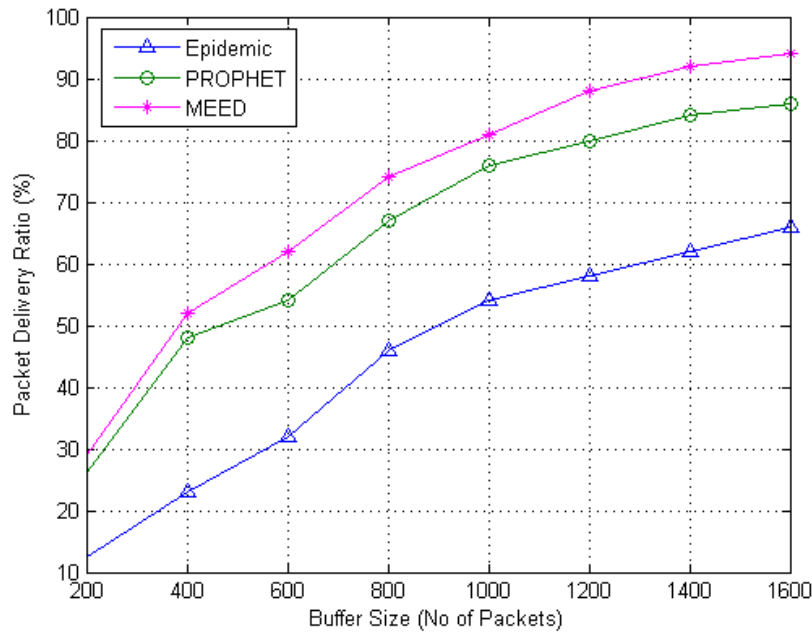


Figure 3.7: Packet Delivery ratio with varying buffer size

ratio compared with other 2 protocol. Nearly 85% packets reached their destinations before the deadline (2400s). Compared with MEED, the second best routing protocol is PROPHET, followed by Epidemic in the same simulation environment. The underlying reason is that both of them are sort of "probabilistic-based" routing protocols, and they assume that the nodes have no knowledge about network. However MEED reasonable assume oracle information for routing, thus achieve reasonable good delivery ratio compared with other routing protocols. Hence in extreme network scenario with sparely few nodes, the routing protocol with which utilize pre-network information achieve good file delivery ratio.

### 3.3.2 Impact of Buffer Constraint

It has been demonstrated that buffer constraints can severely affect the relative and absolute performance of DTN routing schemes and consequently applications. For example, a number of studies have clearly shown that Epidemic routing has minimum delivery delay under no buffer constraints (and no bandwidth constraints), but performs poorly when buffer sizes are limited. However, to understand this we have done simulation to determine the file delivery ratio as a function of maximum storage available per node for our reference scenario. Looking at the graph of the delivery ratio over buffer

### 3.3. DTN BUNDLE PROTOCOLS PERFORMANCE COMPARISONS42

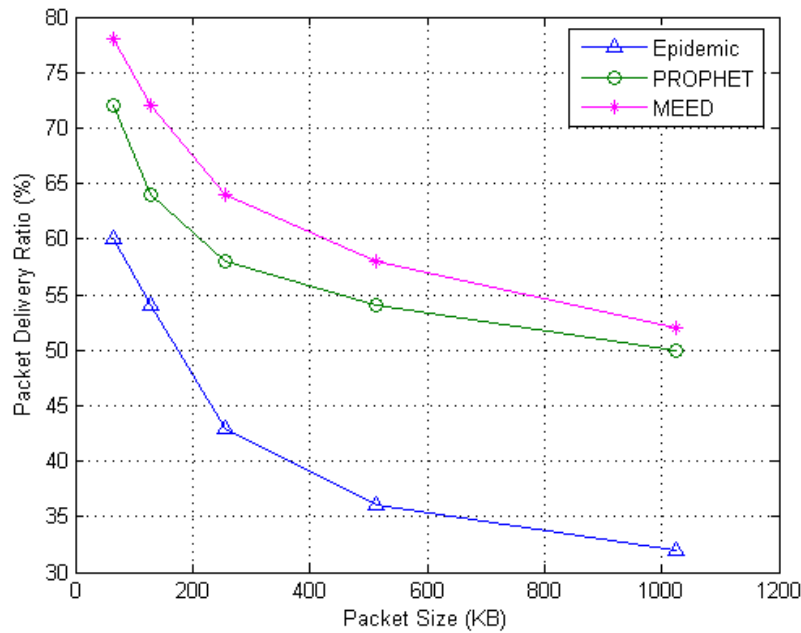


Figure 3.8: Delivery ratio with different packet size

size in Figure 3.7, we can immediately see that the buffer size has significant impact on the Epidemic protocol. As the buffer at each node becomes larger, the delivery ratio increases. The Epidemic protocol only guarantees delivery if there is sufficient buffer to have a copy of every message at every node. If there is insufficient buffer, then some messages must be dropped. Thus, there is a very predictable relationship between the buffer size and the delivery ratio. The other protocols require much less buffer space because they use a single copy routing. Only when the buffer size drops below 10% of the total traffic generated does the delivery ratio start to decrease. In a buffer constrained network, MEED has much better performance than Epidemic and PROPHET routing. PROPHET does not take neighbor buffer storage into consideration while taking routing decision and Epidemic routing uses multiple copies. For this reason both underperform when compared to MEED. Thus in a buffer constrained network, single copy routing protocols perform better than other types of protocols and also the results show the need for good buffer management policy to overcome buffer congestion in intermittent networks like Interplanetary networks.

#### **3.3.3 Impact of Packet Size**

In this section, we analyzed the impact of packet size over the file delivery ratio of the routing protocols. Figures 3.8 show the delivery ratio of each protocol as the size of the packets increases from 64KB to 1024KB. Increasing packet size decreases delivery ratio for all the routing protocols. It is difficult for peers to make use of all transfer opportunities when transferring larger packet, because some of connections are too short in time duration. For this reason all routing protocol underperforms when the packet size is large. Since both MEED and PROPHET have predicted and single copy routing, they outperform flooding based epidemic routing protocol.

To overcome this problem DTN architecture has proposed proactive and reactive fragmentation schemes. However handling reactive fragmentation, with delivery and reassembly of packets (called bundles in DTN) split in the network, whose fragments take different routes to the destination, is an open problem with reliability implications. Hence reactive fragmentation schemes should carefully considered in protocol implementation. Since no fragmentation scheme is implemented in none of the three protocols, their performance is marginal, when packet size is larger.

#### **3.3.4 Performance over Intermitted Link**

Due to rotation of orbiters, movement of Rover and Earth satellites, the communication links are mostly intermitted. Hence it is interesting to analyze the performance routing protocol over intermitted link. In this simulation, we have emulated the intermitted nature by scheduling the up and down time (emulate orbit pass over Rover) of the links in the NS2. We have calculated file deliver ratio for various link up/down time for all three protocols The graph in the Figure 7.3, shows that all three protocol have limited performance when down time is higher, in which MEED and PROPHET perform comparatively better than Epidemic. When down time is higher more packets are generated for a given buffer size and buffer at source and neighbor are filled rapidly. This condition causes congestion at source node and the intermediate nodes and this leads to packet losses which reduce the delivery ratio. Since either of these protocol consider buffer into account during routing decision to avoid buffer congestion. For this reason, their performances are optimal when buffers are limited.

The performance analysis of the protocol over intermitted network shows the need for effective buffer management policy to avoid buffer congestion in buffer limited nodes over intermitted network. Thus in the study we are intended to propose a new

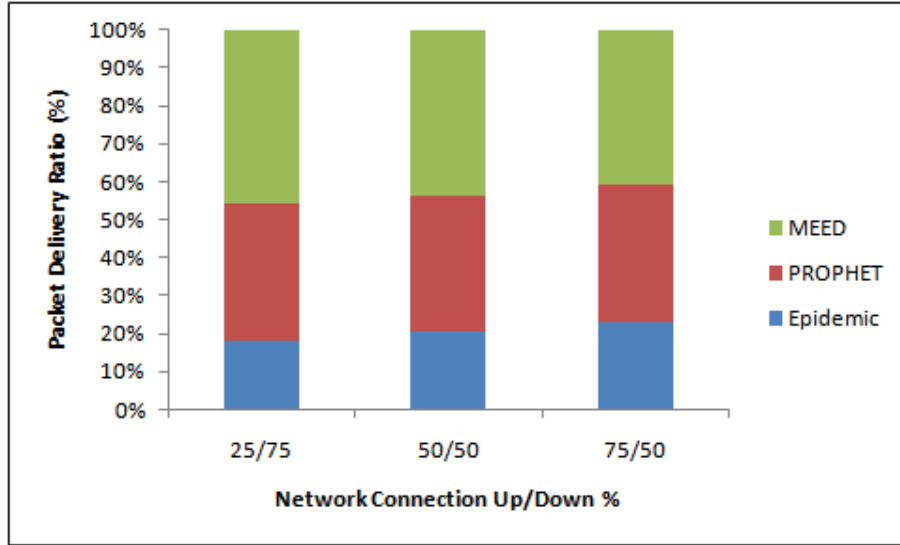


Figure 3.9: Delivery ratio with intermitted links

buffer management policy to improve packet delivery and delay. In the next section, we elaborated the proposed buffer management policy.

### 3.4 Buffer Management Policy for DTN

A buffer management policy defines which message (packets) to drop if the buffer of a node is full when a new message has to be accommodated. Each message (i) in the buffer (B messages in total) has a set of information stored with it ( $S_i$ ): the source id, the time since the message was generated, the Time-To-Live (TTL), etc. In the DTN architecture [50], the TTL value is a timeout value, which specifies when a message is no longer useful and should be deleted. Let us assume a new message  $S_{new}$  arrive at a buffer that is full, then buffer management policy can be defined by a function:  $f(S_1, S_2, \dots, S_B, S_{new}) = S_j, j \in \{[1, B] \cup \{new\}\}$ . That is, based on the information of all messages in the buffer, policy function decide to drop a message  $S_j$  which can either be a message already in the buffer or the newly arrived message.

Future spacecraft and Mars Rovers will have the ability to acquire increasingly larger loads of scientific data. Imaging and multispectral instruments will be deployed in order to send as much information as possible to Earth, for fruition by both the scientific community and the general public. However, the onboard buffer size is unavoidably limited, and part of the data need to be thrown away to avoid buffer congestion. To makes things worse, in interplanetary communications, link availability is severely

limited, due to the long latency; the scarce opportunities for uplink and asymmetric bandwidth. Moreover, the performance analysis conducted over various DTN routing protocol shown the need for good buffer policy to effectively utilize the storage to achieve good throughput with minimum storage latency.

Researchers came up with various solutions to overcome these problems of asymmetric bandwidth, buffer limitation by techniques like good image processing algorithm, compression algorithms and some storing and forward policy. There are researches carried out on image processing [81] and compression algorithms [88] and also buffer management separately. But in this study, we intend to provide a simple and effective policy which combines both images processing and buffer control techniques. Hence new buffer control policy with the combination Image Encoding & buffer control is proposed to maximize data transmission by minimize data lost due to buffer overflow.

Interplanetary network traffic can be classified in to two: real time traffic and best effort traffic. The information which real time traffic carries is very critical and sensitive to delay. For example the traffic of command, control, and navigation systems of various entities in the space network are of such type. They are given higher priority over other traffics and generally the resources in terms of energy, bandwidth, processing power and memory are reserved for them. While the best effort traffic carries the information which are not sensitive to delay and no special resources are reserved for such traffic. For example: transmission of image data taken by space telescope. The proposed set of buffer control rules for both real time and best effort traffic is elaborated in the following Section.

### 3.4.1 Control strategy for Real Time traffic

In case of real time traffic, for simplicity a local knowledge base drop policy is introduced, where any node in the network on sensing buffer congestion, drop messages according to certain rules without any global knowledge of the network, Since global knowledge based drop policy is difficult to be implement. Given the above limitation, a key question to answer is: When the buffer in the node gets congested, which message should be dropped so as to optimize a specific routing metric?

Let us assume that all the messages routed in the network with a finite TTL value. In DTN routing, the source node keep a copy of the messages until TTL value of message get expired. Intermediate nodes are not obliged to do so. Let us consider a time instant  $t$  where a new message copy arrives to new node during an encounter, to find node buffer full. To answer, which is the best message to be dropped (locally),

among the messages already in the buffer and the newly arrived message, in order to avoid buffer congestion following policy is proposed.

**Drop Policy** Let assume K number of messages occupied in buffer of the node N where  $R_j$  represents remanning TTL value of message j such that  $j \in [1, K]$ , and  $E_j$  represent the elapse time for the message j, which is time since the message j was generated by the source node. Where  $E_j$  is give by difference between time when the packet is generated by the source with time spent by the packet in the network before reaching destination (queueing time):  $TTL_j - R_j$ . When a new message i arrived at node N, at the moment when the drop decision by the node is to be taken, then Node N drop the message  $j_{min}$  satisfying:

$$j_{min} = \arg \min_{j \in [0, K]} E_j, \quad (3.1)$$

Where j is oldest message generated by the source node in the network. Since in real time traffic, oldest message are almost always obsolete compared to newly generated messages, dropping the oldest message seems appropriate.

In case of forwarding decision, the packet from the tail of the queue is transfer first. Interplanetary application like command & control, where status of Mars Rovers and other Planetary assert are controlled. Most recent information in these types of application has significant value when compared to older information (current status of Mars Battery System). For this reason, in this type of traffic, packets from tail (recently arrived packet or bundle in terms of DTN node) are transmission first, when a communication window is available.

### 3.4.2 Control strategy for Image data traffic

Planetary exploration missions often have the capability of collecting more data volume than the mission is allocated in downlink bandwidth. Currently, instruments often collect and transmit data at a rate that equals their bit limit, remaining idle much of the remaining time. One scenario to take advantage of instrument capability is to permit continuous data collection and to autonomously select onboard the most scientifically valuable images for downlink. Given the same bandwidth constraint, this approach should increase the diversity and value of the transmitted data. It is clear that the data gathered in a mission cannot be all transmitted to Earth. The question is: What part of the data should be downlinked, and what should be trashed?

Various researches are done by space agency to define a metric for measuring the relative importance of gathered information by effective image processing and encoding algorithm. Besides providing a metric for the importance of a data segment, we need to be able to describe the dynamics of the communication system from a global standpoint, and to optimize the system performance in terms of science return. It should be clear that, due to the possibly variable rate of data acquisition in deep space mission, as well as to the bit rate bottleneck of the downlink channel, buffering is an extremely important operation. Buffer memory is a limited resource, and we should make the best use of it by means of a specific control strategy. The goal of the combined encoding/buffer system is to maximize the value of data transmitted and minimize the value of data lost due to buffer overflow. We introduce a control strategy that maximizes the buffers usefulness by keeping it constantly full and overflowing. With the help of

---

**Algorithm 3.1** Drop Policy for Image traffic

---

When a packet is generated the TTL value of the packet is set according to its priority index. (High TTL value is set for High priority Packets). And let  $K$  number of packets are occupied in buffer at the node  $N$  and are stored in the ordered list according to their priority index value  $i$  such that  $K = k^i, k^{i-1}, k^{i-2}, \dots, k^{i-n}, i \in [1, n]$ . And a new packet  $k^x$  arrives at the node  $N$  with priority index of  $x$

**if** { Buffer storage of node  $N$  is not FULL } **then**

Insert the packet  $k^x$  in the ordered list according to the order of its priority index value with respect to other packets priority index in the buffer.

**else if** Buffer storage of node  $N$  is FULL **then**

Compare the priority of incoming packet  $k^x$  with priority index of last packet at end of ordered list  $k^{i-n}$  and

**if** {  $x \geq i - n$  } **then**

Drop packet  $k^{i-n}$  and insert the new packet  $k^x$  according to the order of its priority index.

**else**

Drop the incoming packet  $k^x$

**end if**

**end if**

---

effective image processing algorithm [81], the data stream of an image is prioritized according to its significance in the image, and the most significant segments of data are give more priority than other data in a give image. A data segment (or buffer packet)



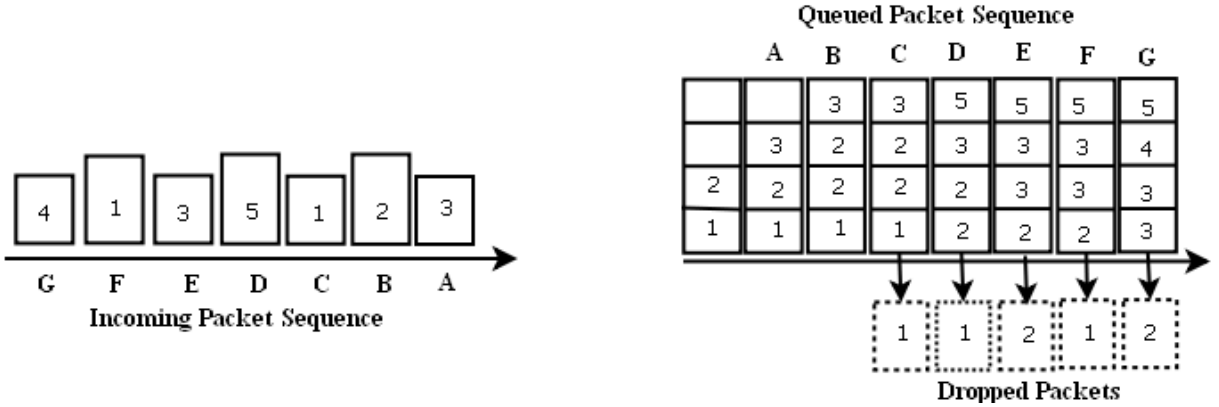


Figure 3.10: Packets Incoming and Queuing Sequence in Buffer

is the smallest data unit handled by the buffer. We have proposed a very simple buffer management strategy which takes the advantage of effectively drafted image processing algorithm which prioritizes the data packets. In our strategy we queue the incoming packets in the buffer in a ordered list according to their priority index provided by the image processing algorithm. During the generation of packet TTL values are assigned according to the priority index of the packets.

Let us say, if  $K$  packets are stored in the buffer of a node  $N$ , they are ordered according to their priority index  $i$  and their TTL value are assigned during the generation of packet according to their priority index value. Lets  $K = \{k_j^i, k_{j-1}^{i-1}, k_{j-2}^{i-2}, \dots, k_{j-n}^{i-n}\}$ , packets are generated at source node with priority index  $i$  given to each packets by the image processing algorithm, then the TTL value are assigned to each packets according to their priority index  $i$  such as:  $TTL_{k_j} > TTL_{k_{j-1}} > TTL_{k_{j-2}}$ , where  $j \in [0, K]$ . Thus the packet having higher priority index  $k_j^i$  is given higher TTL value  $TTL_{k_j}$  and inserted at top of ordered list in the node buffer, which are then de-queued first. Least priority packets are queued at bottom of the ordered list with high drop probability.

When a new packet arrives at the node  $N$  with the priority index value  $x$  and buffer is not full, then buffer controller compare the priority index value of the incoming packet  $k^x$  with the priority index value  $i - n$  of last packet  $k^{i-n}$  in the ordered list. If the priority index value of the incoming packet is greater than priority of index of the last packet in list as per equation 3.2

$$x \geq i - n \tag{3.2}$$

then the incoming packet  $k^x$  is inserted into the ordered list according to the order of its

priority index value  $x$  with respect to priority index value of other packets in the queue

If the packet  $k^x$  arrives at moment, when the node buffer is full, the buffer controller check whether equation 3.2 is satisfied, and if the equation 3.2 satisfied, it drop the packet  $k^{i-n}$  with lowest priority index value  $i - n$  and insert the incoming packet  $k^x$  into the order list according to the order of its priority index value  $x$  with respect to priority index value of other packets in the queue. And if the equation 3.2 is not satisfied then its drop the incoming packet  $k^x$

The Figure 3.10 shows both incoming and queued packet sequence in the node buffer. Here we can see the way in which packet are queued and dropped. From the Figure we can also see that, more then one packets have same priority index value and they are ordered at bottom of the list. For example, when packet with high priority index value compared to the index value of packets in the bottom of the list arrives at the moment of drop, then buffer controller drop the packet from the bottom of the order list such that the packet with minimum TTL value among the packets with same priority index value is selected.

As the high priority index value packets are assigned higher TTL value and are ordered at top of list, increase the delivery probability with low drop rate. Thus we can argue that, proposed policy improve the throughput efficacy of high priority packets with given bandwidth constrain over intermitted network. And, since the packet with high priority index values are more than enough to retrieve the whole image information, even though few low priority packets are lost due to buffer congestion improves goodput performance. In our buffer management policy, when a very high-priority packet arrives at the buffer, they are transmitted with minimum storage delay, as packet from top of the ordered list are forwarded first, which reduce the global delay of whole image date, thus the policy indirectly improve overall latency.

### 3.4.3 Performance of the buffer management policy

In this section, we have evaluated the performance of the proposed buffer management policy for both the traffics classes. The evaluation of the proposed solutions has been performed through NS-2, by properly extending the DTN modules developed by [41] and implementing priority generator and buffer management policy within the Bundle Protocol layer. In particular, priority index is random number between [1, 9] and it is randomly assigned to the packets generated. TTL values equal to 250s, 150s, 50s are given for the packets with the priority index value between, [9-7], [6-4], and [3-1] respectively. In order to understand the impact of buffer policy over planetary

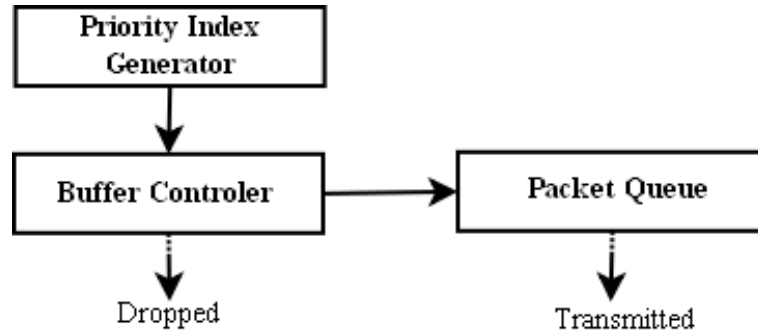


Figure 3.11: Buffer Controller Blocks

Table 3.2: Simulation Parameters

Simulation Parameters	Value
TTL Value	[1-3] 50s [6-4] 150s [9-7] 250s
Traffic	CBR
Buffer Size	400 bundle
Bundle size	64Kbytes
Bandwidth	2 Mbit/s
Application rate	4 bundles/s

node, we have used a static routing scheme in the simulation. We have considered a interplanetary network topology for the simulation as shown in Figure 7.2. In more detail, the propagation delay amongst interplanetary backbone nodes has been set to 20 s. Moreover, each node implements a bundle layer buffer size equal to 400 bundles. On the other hand, the propagation delay between planetary nodes and gateway nodes has been set to 0.5 s, whereas the available link capacity to 2 Mbit/s. Constant Bit Rate (CBR) traffic sources are considered: they are kept active for 150 s of simulation and generate data bundles of 64 Kbytes at rate of 4 bundles/s, yielding 2.048 Kbit/s. In all set of simulation Bundle protocol is used a transport agent and each node in the topology are defined a DTN Agent node [41]. To introduce intermitted nature of the network, a link failure model (rt-model in NS2 [62]) is added in to simulation scenario. The simulation parameters consider for almost all studies are shown in Table 3.2.

The performance analysis has been two-fold: Macroscopic and Microscopic. In the former, the investigation looks into performance provided by the whole network. In this light two metric are considered: Packet Delivery Ratio (PDR) and Goodput. In this first is defined as ratio between the number of received and of transmitted packets. The second accounts for the number of useful bits per unit of time. In the case

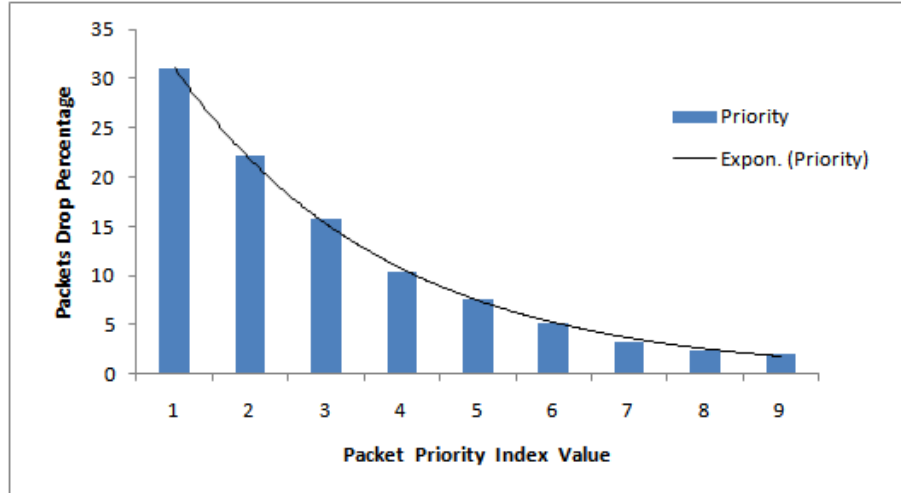


Figure 3.12: Packets Drop Percentage at Source Rover Node

of Microscopic Analysis, attention has been paid to look into performance provided by buffer controller within each node. To this end, two metrics have been introduced to assess the effectiveness of the mechanism: Packet Drop Percentage (PDP) and Buffer Queue Length (BQL). Where, PDP gives the Number of packets dropped by buffer controller in the order of their priority index. BQL figures give indications on how treatment for the packets are given according to their priority.

### Microscopic Analysis

We have considered only the drop policy for best effort traffic in this analysis. The analysis of Packet drop percentage provides meaningful insight into the dynamics of the drop policy within source node. As the Rover nodes are most limited node in terms of storage and bandwidth, we have calculated PDP metric for Rover node, which is source node in the network. The graph in the Figure 3.12 shows an exponential drop in packets according to their priority index value. Lower the priority index of the packet, higher the percentage of PDP, and we can see that drop percentage of higher priority packets (index = 9) are quite low. Drop probability of low priority packets are set exponentially towards the high priority index packets in the policy for this reason we can see an exponential drop distribution with the priority index. Thus we can attribute that our proposed buffer management policy works in the defined manner.

To understand the influence of our buffer management policy, we have compared the PDP metric with DropTail (DT) policy. In drop tail policy each packet is treated identically. With tail drop, when the queue is filled to its maximum capacity,

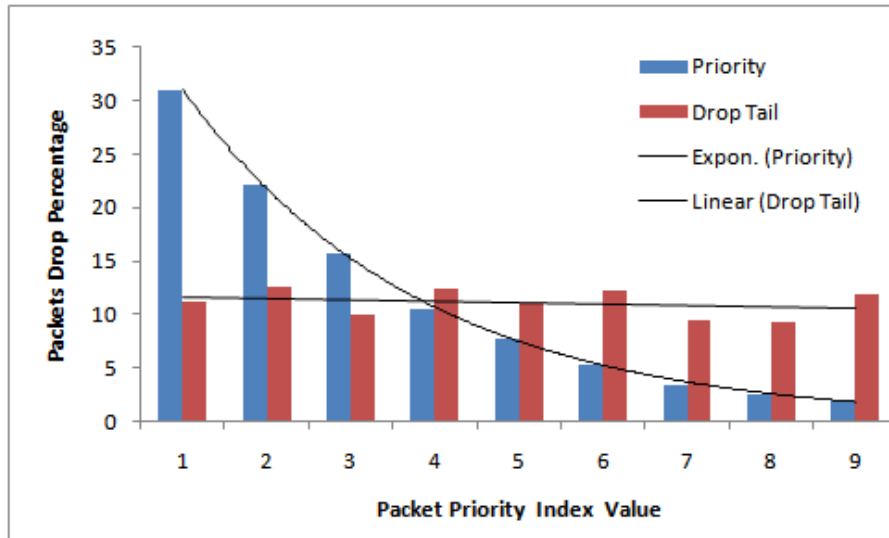


Figure 3.13: Comparison of PDP at Source Rover Node

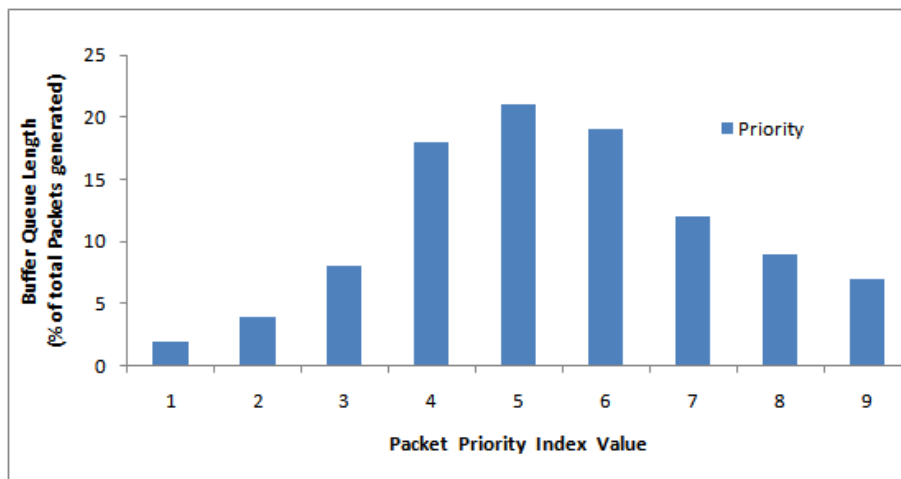


Figure 3.14: Buffer Queue Length at Source Rover Node

the newly arriving packets are dropped until the queue has enough room to accept incoming traffic. From the Figure 7.4 we can see that PDP of DT policy have a linear distribution with respect to the given priority index of the packets, reason being that drop tail policy treats all packets equally and have equal drop probability. Where else in our policy, we can see an exponential distribution with respect to the priority index value of the packets.

Buffer Queue length shows the length of queue with the given priority index at the given node buffer. The graph in the figure 3.14 shows the packet queue length percentage at the source buffer in the network. And we can see from the graph that packets with very low and very high priority index have less queue length when compared with medium priority packets. This is reasonable in our buffer policy as the packets with lower priority are dropped either after queuing or before queuing with high probability. Packet with high priority index has higher probability of delivery compared with other index packets hence lower queue length percentage. Thus the packets with medium priority index value stay longer which results in higher queue length.

### Macroscopic Analysis for Real Time Traffic

In macroscopic Analysis, we have evaluated the performance of drop policy for both the real time and best effort traffic. For both the simulation we used the same topology and parameters (Table 3.2) used in microscopic analysis. The difference being, for real time traffic we don't use priority generator module while generating the packets. In this subsection, we have presented the performance analysis of real time traffic. For real time traffic, in addition to the Packet Delivery Ratio metric, we also considered the data delivery time metric which accounts for the time interval required to complete the data delivery to destinations. We have compared the proposed drop policy with other drop policy like Drop Last (DL) and Drop Oldest (DO) to understand the efficiency of the proposed policy. Drop Last is the most common used policy, it simply removes the new received message and In Drop oldest, the message with the shortest remaining life time is the first to be dropped. Figure 7.5(a) shows the PDR for all drop policy considered in the simulation. We can see from the graph that DL perform poorly in real time traffic, since it drop the new arriving packets (significant information) and store packets having lower remaining TTL value, which will get expired before reaching the destination. Though both DO and proposed policy drop packets according to their TTL value, DO drop packets by their remaining TTL value (oldest packet in the queue), where else proposed policy drop according to their elapse time (oldest generated packet).

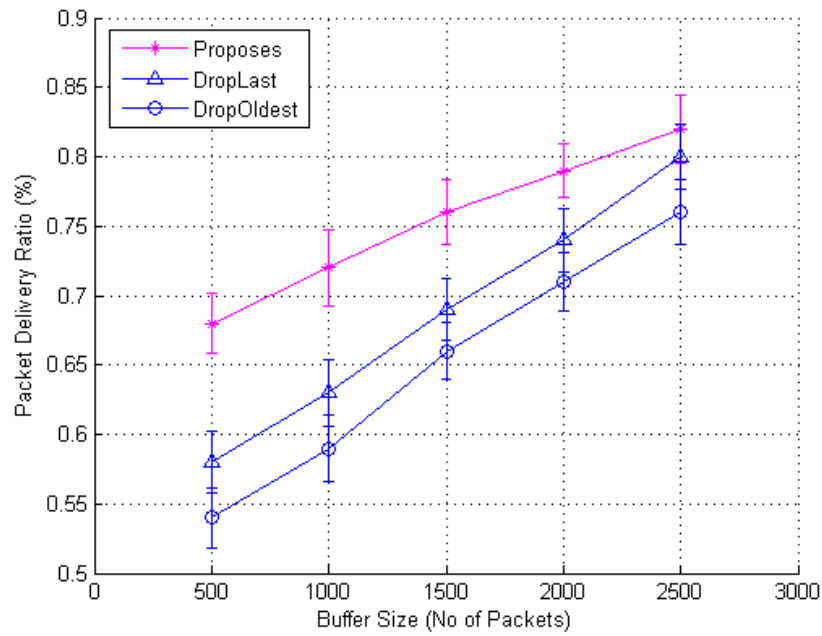


Figure 3.15: Packet Delivery Ratio for Real Time Traffic

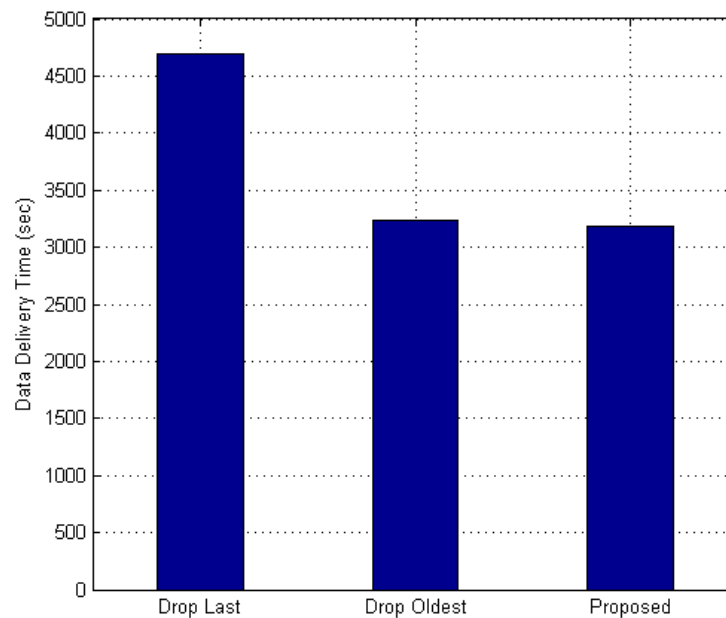


Figure 3.16: Data Delivery Time Performance

But they differ in forwarding policy, where DO forward the packet from the head, where proposed policy forward the packet from the tail. The packets in head have lower elapse time than the packet in the tail and they have lower probability of delivery to the destination, thus DO have less PDR when compared to the proposed policy.

To calculate the data delivery time (ddt) metric, we have used the same simulation setup as before and result is plotted in Figure 3.16. From the graph, we can see that DL has high DDT when compared to DO or proposed. The reason being, that in DL newly arrived packets are being drop in order to avoid buffer congestion, thus packets which are delivered have high queuing delay, resulting in high data delivery time. On the other hand, both DO and proposed used TTL value for drop decision, leading to lower DDT value. The marginal improve achieved by the proposed policy is due to the forwarding scheme adopted in the policy. Thus we can say that proposed policy achieve good PDR and optimum DDT when compared to other policies in interplanetary network scenario.

#### Macroscopic Analysis for Best Effort Traffic

In this subsection, we have presented the performance analysis of our buffer policy over best effort traffic. In this analysis we can considered two metric: PDR and Goodput. We have used the same simulation scenario, which we have used in microscopic analysis. Figure 3.17, present the packet delivery ratio for all considered buffer management policy. From the figure it can be seen that the proposed policy outperforms existing policies DO & DL. For DL same inference can be made as real time traffic for its poor performance. DL drops in coming packets to the congested buffer which may have higher TTL value than packets in the buffer having less Remaining TTL value. For this reason, they have poor Packet delivery ratio. The proposed policy assign different TTL values for the packets according to their priority index, where in DO, TTL value are set equal for all generated packets. As both the policy forwards the packets from the head of queue, in DO packets in head of the queue are packets which are stayed quite long compared to packets at the tail of the queue, hence they have lower probability of delivery rate. Where else in proposed policy packets at head of the queue are high priority packets with high TTL values having higher probability of delivery, thus in the Figure we can see that proposed policy achieve higher PDR when compared to DO & DL policy.

Figure 7.5(b) shows the mean percent change of maximum network goodput versus buffer size. As buffer size decrease we can the degradation of network goodput



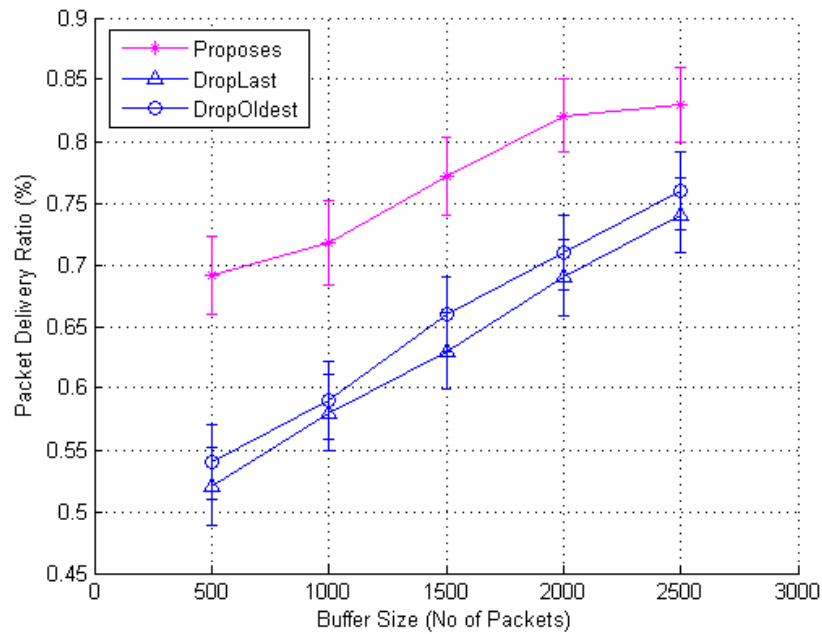


Figure 3.17: Packet Delivery Ratio for Best effort Traffic

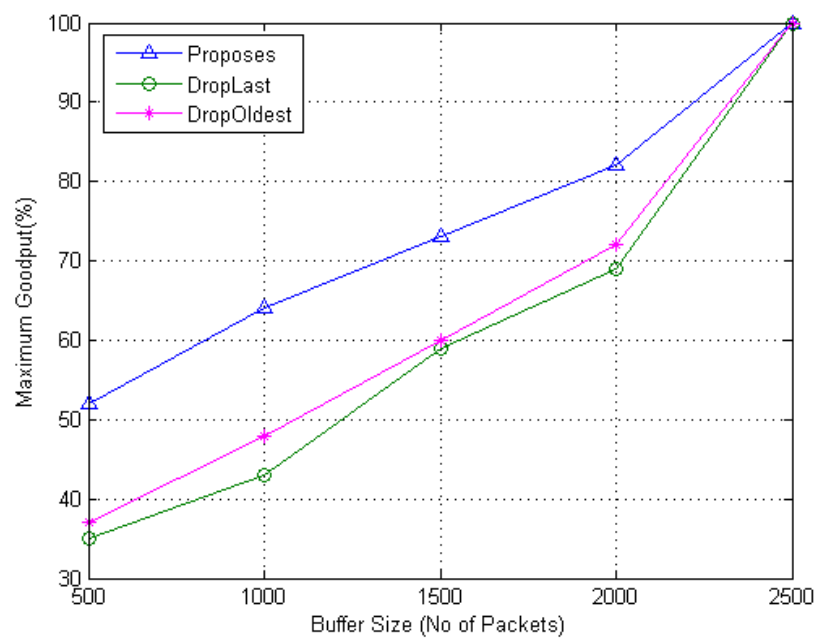


Figure 3.18: Mean percent change of maximum network goodput for Best effort Traffic

performance for all three drop policy. But proposed policy performs better compared to other two. Image data sent by rover can be reconstructed fully with high priority indexed packets (thanks to image processing algorithms), and since proposed policy delivery high priority packets with high probability (By setting high TTL value), it achieve good network goodput when compared to other policy (DO & DL). In both DL and DO, the indifference showed in packet priority results in lower goodput performance. Thus network goodput degradation percentage is less for proposed policy when compared to other two drop policy.

### 3.5 Discussion

In this chapter, we have analyzed the performance of the existing DTN routing protocol over interplanetary network scenario. Based on the performance analysis of the routing protocol over the constrained network scenario, we have identified that there is a need for effective buffer management policy for the resource limited nodes in the interplanetary network. Thus, we have proposed a simple buffer management policy for these resource limited nodes for both real time traffic and best effort traffic (in particular image data traffic). We have then evaluated the proposed buffer management policy over other policies and shown that proposed policy achieve good performance in terms of delivery and delay using simulation. Though the performance of the buffer management policy mostly depend upon the routing protocol, we then shift our attention to a resource-aware routing protocol which adapt the proposed buffer management policy in the next chapter.

## Chapter 4

# Resource Aware Routing for Interplanetary Mars Surface Network

Chapter 4 accomplish the research work done in chapter 3, by proposing a resource-aware routing for Mars proximity network by taking into consideration the limitations of Rover/Landers nodes like Power and Storage. As a first step, an antenna selection algorithm for Mars Exploration Rovers is proposed, which act as a base for the design of Resource-aware routing protocol.

### 4.1 Motivation

Mars surface exploration missions up to date have featured humans and robots operating essentially independently. The communications requirements for these missions have been relatively simple, the surface elements needed simply to communicate back to Earth, either directly or via an orbiting relay satellite. In contrast, future Mars surface exploration missions will incorporate teams of humans and robots working together to achieve science and engineering goals on planetary surfaces. For example, a geologist collecting samples on the Mars surface may work with a robotic assistant to annotate or analyze those samples. As a result, flexible and dynamic planetary communications are critical to the success of Mars space exploration vision.

The use of direct wireless communications among local surface elements will be necessary to achieve optimal communications efficiency. However, the surface elements are mobile and may lose communication with one another, due to traveling either out of

Table 4.1: Comparison of MER DTE and UHF Relay Link

Parameters	Direct-to-Earth(DTE)	Relay
Band [39]	X-band (8.4 GHz)	UHF (0.4 GHz)
RF transmit power (W)	15	12
Antenna	28 cm HGA (24.8 dB) LGA (6.9 dB)	Monopole (2 dBi)
Telemetry data rate	3.6 kbps to DSN 70m (@ end of prime mission 2.1AU)	128 kbps nominal to MGS/ODY (256 kbps for favorable ODY over flights)

range or behind an obstruction. This problem can be addressed through the application of a mobile ad hoc like routing protocols [10], which allowing nodes unable to communicate directly to remain in contact by relaying data through one or more intermediate nodes. Though ad hoc routing partially solved the problem, still resource limited issues should be addressed.

Currently Mars surface nodes are remotely operated from Earth with detail configuration and operation plans uploaded into the rovers. Alternatively, it would be interesting to embed some intelligence in the surface nodes (Rover/Landers) to operate autonomously, requiring manual configuration or intervention only rarely and under exceptional circumstance. Moreover, Mars surface nodes are resource limited (Buffer, Power).

Thus a novel unicast routing protocol is proposed in the chapter, to provide intelligence in the Rover/Lander nodes to take routing decision autonomously. To enhance performance, the protocol confined the routing decision taken based on resource available at the give node.

As first step, different type of antenna in-build in Rover/Lander are explored for their characteristic features and condition in which, selecting one of the antenna can achieve good performance results.

## 4.2 Why to Study Mars Rovers

Typically in Mars Missions, Mars Rovers [93] are the main traffic source, and its mission is to explore the planet surface and gather raw science data and analysis results (e.g., chemical and geological analysis of rocks and soil samples) to send back to Earth. Various researches [102] [46] are going on to improve the data return ability of Mars Rover. As the amount of data return from Mars increases, the need for robust communication architecture and routing are obvious. And it must not only meet the

requirement of the current mission, but also enable to adapt to future missions as well. Moreover Rover design limits the communication ability [96]. Thus it is interesting to study the characteristic limitation of Mars Rover which impacts the communication ability of the Rover. For this reason, in the following Sections we have explored the Mars Rover and its possible link characteristic as a function of its resource limitations, which act as basic requirement for our proposed resource aware routing protocol.

### 4.3 Mars Exploration Rovers - Antenna

In this section, Antenna of Rover/Lander are explore based upon their character and ability. Typically a Lander/Rover will carry two communication systems [93] [21]. The first one is an X-band Direct-to-Earth (DTE), Direct-from-Earth (DFE) system, designed to operate at lower data rate with higher reliability. The other system is a higher-data rate UHF radio with omni-directional antenna designed to communicate with the satellites orbiting around Mars like, the Mars Express and the Mars Odyssey [33]. Key Rover telecom parameters are summarized in Table 4.1.

The main objective of Rover communication system is to return the investigated data back to earth. When a Rover has some information to send, it sends the data using the available communication link either DTE or Relay link, intuitively Rover select an antenna to transmit the information back to earth. Hence a focus is given in the direction to embed some intelligence in Rover by which Rover can select by itself the communication link for transmission according to its requirement and availabilities (like power or link capacity).

Almost 90% of the data returned by the Rovers (Spirit and Opportunity) has been communicated over the Mars odyssey relay link and only 3% of data is returned using DTE. UHF link has been heavily favored. But with the recent advancement [16] in X-band transmitters, prompted us to look in to different direction. As proposed in [16], X-band appliques provide the Electra UHF transceiver with dual UHF and X-band capability. For example if the Lander/Rover equipped with appropriate transmitter appliqué can be able to communicate using both DTE and UHF at same time.

#### 4.3.1 Resource Limitations

The Mars Rovers are typically highly constrained in the mass, volume, power, and energy available for communications. The power and energy sources in a Rover greatly affect the Rover design, capabilities [96], and mission architecture. The power

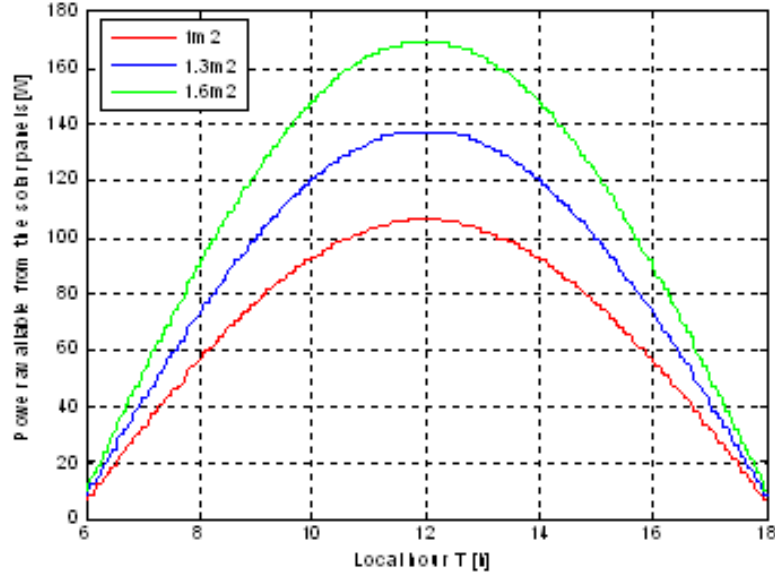


Figure 4.1: Power Available in the Rover for 1 sol day for different Antenna size

and energy requirements of a resource limited Rover can be broken down into three categories: internal, external, and mobility. In which only power required for communication (external) is considered in this study.

On both Rovers Spirit and Opportunity, solar panels are considered as main source for power generation. There are various factors which affect the generation of power in the Rover; like eccentricity of the Mars orbit; quantity of dust in the atmosphere; seasons; time of the day; panel orientation. In order to understand the impact of these factors over the power generated in the Rover, the power level is measure in the Rover for one Mars day (1 sol) between 6hr-18hr (day time) using the formula give in equation 4.1.

$$P = G_h * \eta_{solar} * A_{solar} \quad (4.1)$$

Where  $P$  is power generated by the solar panel;  $\eta_{solar}$  is efficiency of converting solar radiation into power;  $A_{solar}$  is the area of the solar panel and  $G_h$  is the global irradiance on the horizontal surface given by the equation 4.2.

$$G_h = G_{bh} + G_{dh} \quad (4.2)$$

Where  $G_{bh}$ , is the direct beam irradiance on the horizontal plane and  $G_{dh}$  is the diffused irradiance of the horizontal plane. As we can see from the Figure 4.1, due to the variance of solar irradiance throughout the day, the output of solar panel has similar effect in

power generation. The result is a window of operation, in which the Rover has enough power to perform tasks.

### 4.3.2 Communication Limitations

Another main challenge in MER is the communication window selection. In the mission, systems analysts on the strategic team (both rovers, longer-term, Earth time) produced most communication windows, and systems analysts on the tactical team (dedicated Rover, sol-by-sol planning, and Mars time) modified, added, and deleted communication windows based on how their Rover was doing. The Telecom analysts (all on the tactical team) checked window parameter values and the consistency with DSN station tracking times or UHF orbiter overflights, using a set of flight rules and the X-band and UHF predictions. After the windows were validated, they formed part of the command upload sent to each Rover. Checking was manual. And there is always a possibility of human error.

Suppose that during a fine Martian sol, one portion of a Rover's onboard data storage device fails. With stereo image data flowing in, the remaining buffer space may well get overwhelmed before the next opportunity to submit diagnostic telemetry back to Earth. As a result, the final reading from a spectrometer of the sol's rock-grinding experiments might get discarded. After all, it can take up to an Earth day for ground control to react and reconfigure the mission.

Various researches are going on to automate the selection process to eliminate some of the manual effort and human error. Given these issue; It is intended in this study to propose some intelligence in MER (selection algorithm), by which it can select by itself how it can send data back to Earth with respect to its own resource conditions and application requirements to improve data return ability. To provide some intelligence in MER to select the appropriate link for transmission, it is necessary to identify all possible communication links and their characteristics.

## 4.4 Rover Communication Links - characters

In this Section, all possible communication links and their characteristics are explored as the function of their resources: mainly power.

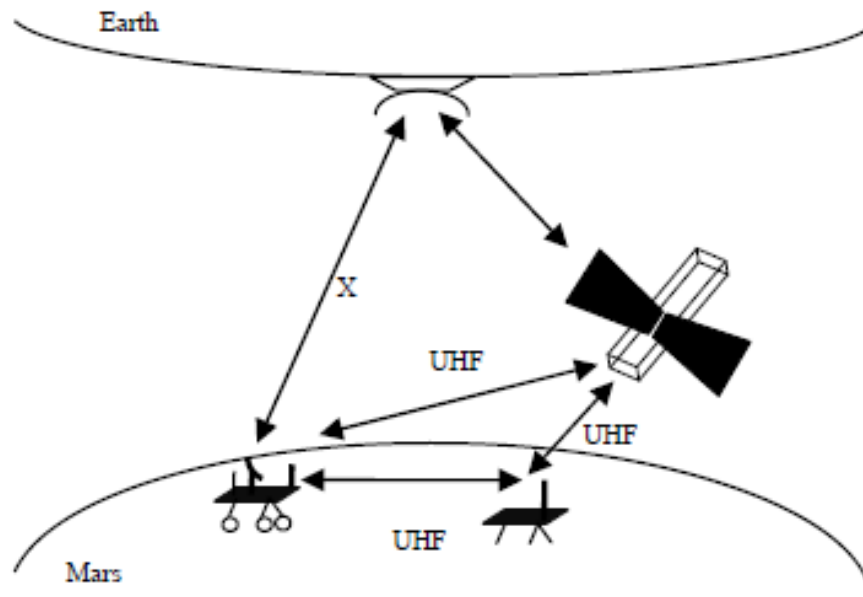


Figure 4.2: Basic Network Architecture

#### 4.4.1 Network Architecture

In this subsection Mars surface network architecture is describe to categorize all possible Rover links to the earth ground station and the parameters which affect their communication performance. The investigated scenario is derived from the ongoing space missions aimed at exploring the Mars surface, collecting images and measurements and transmitting them to Earth gathering centers via a suitable communication infrastructure. In detail, the network architecture as shown in the Figure 4.2 consists of Rovers and Landers placed on the Mars surface, responsible for taking measures and pictures, which communicate either through Direct-to-Earth (DTE) or through relay Orbit to earth ground station.

The parameters which affect the performance of the communication system of Rover were, transmission power, bandwidth delay product, bit error rate, and periodicity of the links. Relation between there parameters are briefly described in the table 4.2; In the following subsection, the characterizers of all possible link are analyzed as function of power. Main aim here is to tabulate all parameters which are required to calculate the power required for transmission using corresponding antenna in each link.



Table 4.2: Link parameters relations

Parameters	Detail
System Power	Power generated at the Rover depends on the design of the Rover solar panel
Attenuation	Attenuation depended mostly on the distance between two entity and atmospheric conditions
Bit error Rate	Bit error rate is mostly depend on the atmospheric changes (rain; wind; clear sky)
Periodicity of Link	Periodicity of the link depends on geometry of the orbit and position of Rover on the Mars surface
Bandwidth & Data Rate	Bandwidth and data rate are depended on the type of modulation used in each link or channel.

#### 4.4.2 Direct-To-Earth Link

The Rover, roughly the size of a golf cart, included antennas for both relay and DTE communications. However, due to size limitations, the HGA, used for DTE Communication between MER and DSN has been at X-band for all mission phases. X-band (7.2 to 8.5 GHz) [39] is attractive because of the existing near-Earth and deep-space frequency allocations and also onboard hardware could be shared for both the deep-space and local links.

Rover is very much limited in power; here all possible links are characterized as function of power requirement for data transmission. As the power mainly depends on the Rover design and their modulation technique adopted for their bandwidth. For this reason, various type of modulation technique are analyzed and Differential Phase-Shift Keying Modulation (DSPK) which provides simple demodulation technique with minimum transmission loss where chosen as modulation technique for DTE link [45]. DTE link performance is evaluated as function of power received at DSN. Where the power received at DSN is calculated as per the equation 4.3.

$$P_r = P_t + G + L_{loss} \quad (4.3)$$

Where  $P_t$  and  $P_r$  represent the power transmitted and power received at each end with an antenna gain  $G$  and link loss  $L_{loss}$ . This equation 4.3. can be rewritten in detail us:

$$P_t = N + NF + 10 \log \left| \left( \frac{Eb}{No} \frac{R}{B} \right) \right| - G_{tx} - G_{rx} + L_s + L_{add} \quad (4.4)$$

Where  $NF$  = Noise factor (measure of noise introduced by the receiver),  $R$  = Date Rate,  $G_{tx}, G_{rx}$  = Transmission and Receiver antenna gains,  $L_s$  = Free space losses,  $L_{add}$  = Additional losses due to the atmosphere, polarization, the accuracy of the transmission,

Table 4.3: DTE Link Parameters

Parameters	Value
Receive Antenna Gain	75dBi
Emitter Antenna Gain	2dBi
Frequency	8439.444446 MHz (return link)
Bandwidth	8000Hz
Noise Factor	5dB
Noise Temperature	10.7k
Attenuation Losses	8dB
Modulation	DSPK

and Noise power  $N$  is given equation 4.5

$$N = k * T * B \quad (4.5)$$

Where  $K$  is constant;  $T$  = Noise temperature of the System;  $B$  = Noise Bandwidth.

The parameter required to calculate the power required for transmission (equation 4.4) is tabulate in 4.3. Give the type of modulation (Differential Phase-Shift Keying Modulation (DSPK) [45]) and error model, single to noise ration ( $\frac{Eb}{No}$ ) in DTE link can be calculated. Power calculation also influenced by attenuation from both earth atmospheres and Mars atmosphere. Where cumulative atmospheric losses not exceed 8dB under normal condition, and attenuation in Mars surface is approximately 3dB. Attenuations occurring in the environment in interplanetary space have been neglected because they are too small.

#### 4.4.3 Rover-Orbiter Relay Link

A relay link to an orbiting satellite at Mars provides dramatic performance improvement compared to a direct-to-Earth link. The range of Rover-orbiter distance is in the order of  $> 20,000$  times smaller compared to the maximum Rover-Earth distance (= 2.6 AU). When considering that the achievable data rates vary by the square of the distance, the improvement is clearly significant. In spite of the dramatic difference in using a relay link to an orbiting satellite at Mars, Rovers and other Landers still require careful telecommunication design to ensure robust command (forward) and telemetry (return) data paths. In the similar way, as DTE link, Rover-Orbiter relay link is analyzed as function of power and parameters are summarized in the table 4.4. Quadrature Phase Shift Keying (QPSK) modulation technique [61] is selected here to provide high bandwidth efficiency in UHF frequency. Since range between Rover and Orbiter is small, the attenuation losses quite less compared to DTE link.

Table 4.4: Rover - Orbiter Relay Link Parameter

Parameters	Value
Receive Antenna Gain	5dBi
Emitter Antenna Gain	2dBi
Frequency	401.585625 MHz
Bandwidth	128000Hz
Noise Factor	3dB
Noise Temperature	550k
Attenuation Losses	0.1dB
Modulation	QSPK

Table 4.5: Rover-Lander Link Parameters

Parameters	Value
Receive Antenna Gain	2dBi
Emitter Antenna Gain	2dBi
Frequency	435 MHz
Bandwidth	64000Hz
Noise Factor	3dB
Noise Temperature	20k
Attenuation Losses	128dB
Modulation	QSPK

#### 4.4.4 Lander-Rover Link

Communication between Rover and Lander are considered as simple wireless link shown in Figure 4.2. Where Rover communicates with Lander using UHF antenna, when they are in the communication range. This link is modeled similar to Rover-Orbiter Link. Same QPSK modulation technique is used in UHF frequency band.

In this link, RF signal suffers reflection, refraction and take on multiple routes due to relief on Mars Planet and therefore a simple free space attenuation model cannot be applied. The irregular terrain model like Longley-Rice model is used since it suits best in this condition for Mars surface environment. This choice is viable to deliver the mitigation route for frequencies range of 20MHz to 20GHz. This model estimated losses signal for a radio spot with irregularities and for frequency is VHF, UHF and SHF as depending on the distance and vary the signal in time and space. Figure 4.3. exemplifies attenuation between Rover and Lander with their distance between them by using this irregular terrain model. The parameters of the link are summarized in Table 4.5.

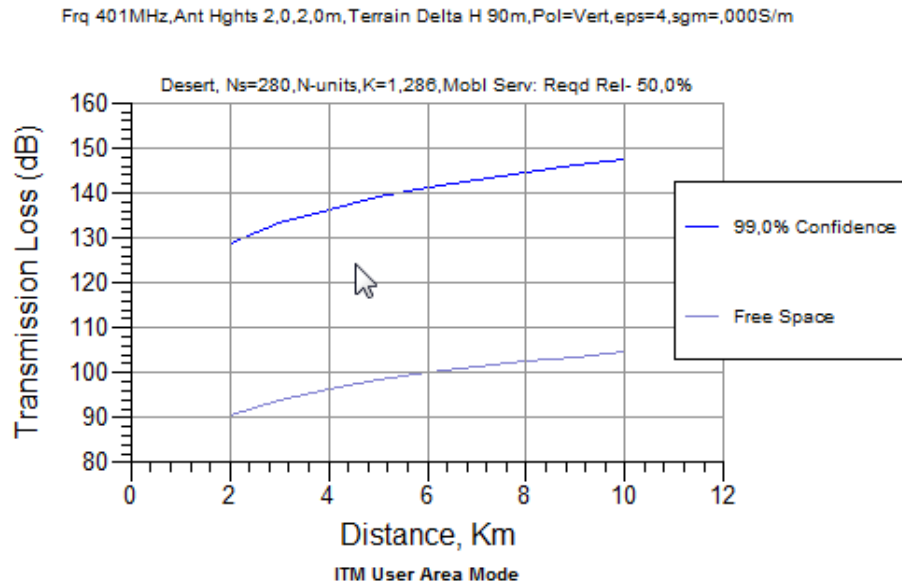


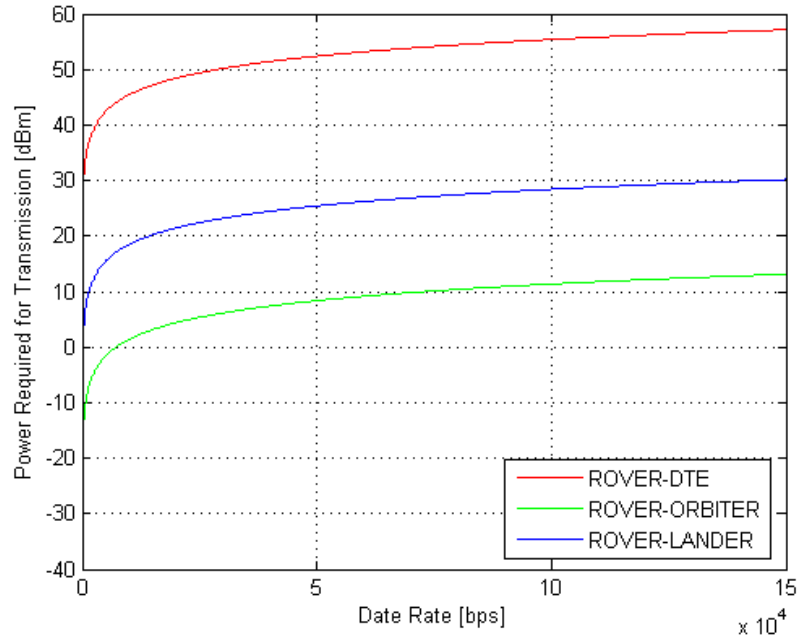
Figure 4.3: Attenuation Measure on Mars surface between Lander and Rover

#### 4.4.5 Power Estimation

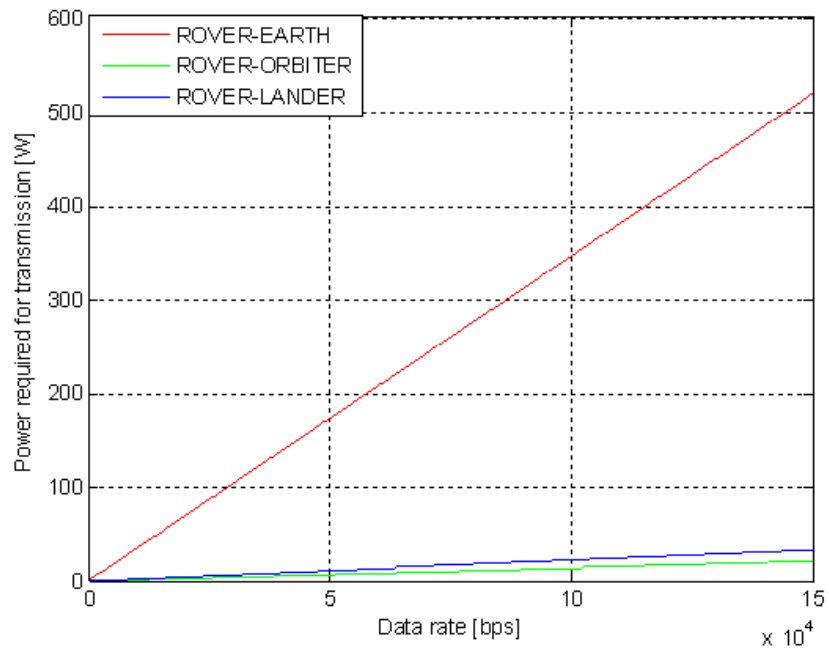
Based on the numerical parameters modeled for each link, power requirement for the transmission in each link for Rover is determined using equation 4.3. To understand the dependency of each parameter, power requirement for transmission in each links is plotted with data rate, shown in Figure 4.4(a). As expected power required for the DTE link is higher compared to Orbiter and Lander link due to distance between Rover and Earth. The power require for Rover-Lander and Rover-Orbiter are almost the same. One explanation is that, even the distance between Rover and Lander is less than Rover-Orbiter, the attenuations due to the environment are stronger than attenuations in free space for 400km (as it was considered the altitude of the satellite).

Figure 4.4(b). reveals the linear dependence between power and data rate, useful for characterizing the link and for establishing their parameter for a certain RF connection (link).

Historically, the data rates for deep space spacecraft are power-limited, not bandwidth-limited; hence, the DSN selected modulation methods that allow capture of all the power, rather than bandwidth. Thus DSPK is selected for power limited DTE and for Relay link, bandwidth-efficient QSPK modulation is selected. Figure 4.5 reveals the theoretical dependence of data rate and the digital bandwidth and as it may be seen the occupied bandwidth increases as data rate increases.



(a) Power Required for Transmission (dB) in Rover depending on data rate



(b) Power Required for Transmission (W) in Rover depending on data rate

Figure 4.4: Power Required for Transmission in Rover depending on data rate

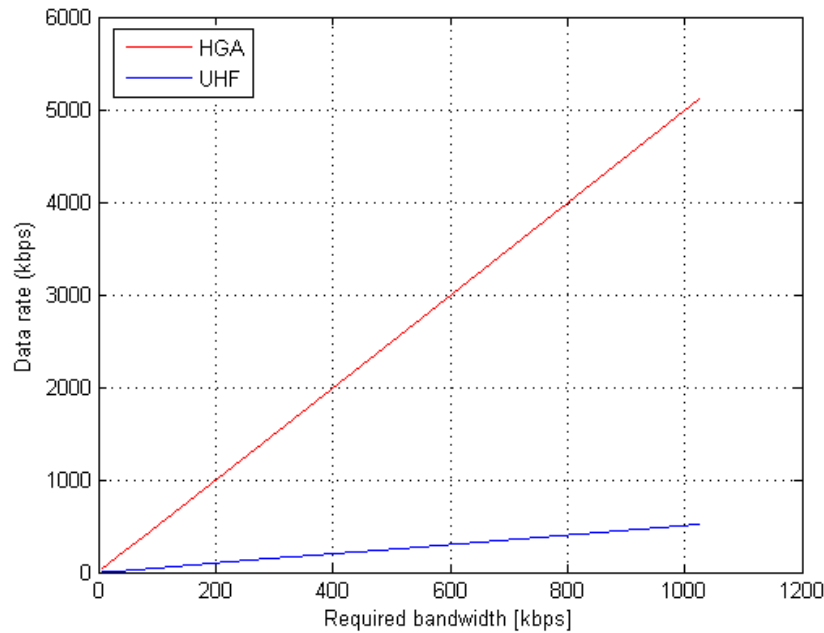


Figure 4.5: Data rate depending on Bandwidth

## 4.5 Cost Based Antenna Selection Algorithm

After characterizing each possible link as function of power, an antenna selection algorithm is proposed. To make Rover more autonomously select its communication antenna for transmission, according to its available resource, this cost based antenna selection algorithm is proposed, which is elaborated in this section. Set of rules are defined with which Rover selects its antenna for communication. Basically these rules assignees the cost value for all available antenna and allow Rover select cost effective antenna for transmission.

### 4.5.1 Basic Assumptions

Algorithm is indented to provide some intelligence in Rover by which it can decide when and by which link it can able to return data effectively back to earth. In order to facilitate the selection process, power generated by the solar panel of the Rover is assumed to be known in advance (before any transmission windows). Orbital movements over Mars surface are periodic due to their geometry of the orbits. For this reason orbital relay link have periodic connection with constant link duration. Thus it is assumed that Rover knows in advance, the time and duration of next possible

Table 4.6: know parameters at the Rover

Parameters	Description
Buffer Size ( $\beta$ )	With recent improved in Mars communication protocol which uses store-and-forward mechanism, storage become a critical parameter to achieve good throughput. Here we assume that the rover know in advance the buffer space available at the Lander ( $\beta_l$ ), orbiter ( $\beta_o$ ) and DSN ( $\beta_{dsn}$ )
Link Duration	Link duration is a measure of pass duration in case of orbiter ( $\zeta_o$ ), visibility period, in the case Lander ( $\zeta_l$ ) and kind of wired link, in the case of DSN ( $\zeta_{dsn}$ )
Propagation delay	Total propagation delay is approximately calculated using ephemeris data [8], where total propagation delay between rover and Earth station is given by ( $\delta_{dsn}$ ) and that of Lander and orbiter is given by ( $\delta_l$ ) and ( $\delta_o$ ).

Orbiter pass. This assumption is made with strong belief that feedback mechanism is possible between Rover and Orbiter; since when Orbiter passed overhead a Rover, the Orbiter signal the Rover/Lander via a hailing message defined by the CCSDS Proximity-1 Space Link Protocol [28] and establishes radio communication for the duration of the pass. With this hailing message; other information like buffer space availability at the Orbiter, possible delay incurred in relaying the Rover data back to earth (which is calculated from ephemeris data [8]) and future orbit pass time. For simplicity constant pass duration for all the passes are assumed. It is argued that Rover used the same antenna for both Rover-Lander path and Rover-orbiter with two different frequencies to avoid interference [52]. With these information's available at the Rover before any communication window make the selection process effective. The know parameters at the Rover before the communication window is tabulated below 4.6

#### 4.5.2 Network Model

The network topology considered under this study has very few nodes, namely; rover; Lander; orbiter and DSN. Hence it is modeled as oriented graph  $G$  with points and edges, where each path has specific cost for transmission.

$$G = (N, E) \quad (4.6)$$

Where  $N$  is number of point node in the graph and  $E$  represent it corresponding edges which connect these nodes. Our intuition is that, selection of a edge for communication

is as equivalent as selection of an antenna for communication in the Rover, thus the cost based antenna selection algorithm is proposed.

### 4.5.3 Cost Function

The path for communication is selected with help of a simple cost function. A transmission cost is associated with each path in the graph and Rover selects the path according the cost it can afford, in general low cost path. The cost function associate with each path is a real valued function give by the equation 7.1.

$$CostC \rightarrow R^+ \quad (4.7)$$

Cost function is derived based on Rover communication parameters; namely the propagation delay, transmission power and packet error rate, which is given by the following equation 7.2

$$CostC(p) = \omega_1\delta(p) + \omega_2\psi(p) + \omega_3\xi(p) \quad (4.8)$$

where  $\omega_1, \omega_2, \omega_3$  are the weighting for each parameter; these values are set depending upon the QoS requirement of the application and are expected to be determined in an heuristic method;  $\psi(p)$  is the power necessary for the Rover to transmit the information on the respective path;  $\delta(p)$  is average total delay in each path  $p$ . Delay is composed of three delay components: 1) queuing time, 2) transmission delay, and 3) propagation delay. Queuing time includes both the time waiting for an edge to become available (waiting time) and the time to drain messages already scheduled for departure on that edge. Minimizing delay lowers the time messages spend in the network, reducing contention for resources; therefore, lowering delay indirectly improves the probability of message delivery. The;  $\xi(p)$  is packet error rate (PER) which depending on the bit error rate (BER) of the path  $p$ ; give by

$$PER = 1 - (1 - BER)^N \quad (4.9)$$

Where, N is number of bits in each packet transmitted.

For simplicity in calculation of the path cost, each parameter in the cost equation is approximated between  $[0, 1]$  according to expected minimum or maximum value of respected parameters (power, delay, error) As thumb rule (heuristic method Table 4.7), value of  $\psi(p)$ ,  $\delta(p)$ ,  $\xi(p)$  are assigned according the expected min/max value in the



Table 4.7: Expected Value of Power, Delay &amp; Error Parameters

Parameter	Expected Value	Description
Transmission Power Figure 4.4(b)	$> 4 \text{ W/Kbps}$	Power required by X-Band Antenna for communication
	$< 1 \text{ W/Kbps}$	Power required by UHF Antenna
Propagation delay [8]	$> 10 \text{ min}$	Propagation delay in DTE path
	$< 10 \text{ min}$	Propagation delay in Relay & Lover path
Bit Error Rate	$10^{-2} - 10^{-4}$	Error rate in deep fade period
	$10^{-5} - 10^{-7}$	Error rate almost clear sky condition

respected path according the equations 4.10, 4.11, 4.12

$$\psi(p) = \begin{cases} 0.5, & \psi_{req}^p > 4W/Kbps \\ 1.0, & 0.5 < \psi_{req}^p < 1W/Kbps \end{cases} \quad (4.10)$$

$$\delta(p) = \begin{cases} 10, & E[\delta] > 10min \\ 5, & E[\delta] < 10min \end{cases} \quad (4.11)$$

$$\xi(p) = \begin{cases} 1.0, & 10^{-2} > E[\xi] > 10^{-4}BER \\ 0.5, & 10^{-5} > E[\xi] > 10^{-7}BER \end{cases} \quad (4.12)$$

For example let us take DTE path where power required by the rover is more than 4W/Kbps with expected delay around 0.8AU, with error rate of  $10^{-3}$  then cost function is give by the following equation

$$CostC(DTE) = 0.5(\omega_1) + 10(\omega_2) + 1.0(\omega_3)$$

#### 4.5.4 Selection Rules

The weight-ages for each parameter in the cost function are given according to the application traffic and their QoS requirements. The main traffics with application QoS requirement are list in the table 4.8 A simple set of rule are defined to allocated the weightage for each parameters in the path, are divided into 3 according to their parameter constrain, the power, delay, and error. Set of rule are defined below:

- Power constrain: At a given instance, when power generated at Rover is low then, the power  $\psi(p)$  is given heights weight compared to other parameters in each path according to their power requirement for transmission, so that path which require less power is selected
- Error Constrain: If the application data is required to be transferred with minimum error; then high weightage is give to  $\xi(p)$  value in each path which has

Table 4.8: QoS Metric for different interplanetary application

Application	QoS Metric	Description
Telemetry	Delay ( $D_{max}$ ) and Throughput ( $B_{min}$ )	This is the primary application which report health and status of spacecraft. An important aspect of telemetry systems is its delivery characteristics, which are either "stream-oriented" or periodically delivered.
Scientific data	Best Effort	Scientific data delivery is one of the prime application interplanetary network, which are in general not particularly time sensitive, have high tolerance of poor timing estimation and error coping abilities.
Multimedia	Delay ( $D_{max}$ ) and Throughput (constant)	This type of application is required to deliver great volumes of audio and visual information about the local environment to Earth. They are time sensitive applications, which require high data volume.
Command & Control	Delay ( $d_{Max}$ ) and Low loss rate	Command and control refers to the closed-loop control of remote systems. These types of applications have low tolerance of poor timing estimation and error coping abilities.

minimum probability of error with respect to other so that path with minimum error rate is selected.

- Delay constrain: If the application data is required to be transferred with minimum delay; the same principal as error constrain is carried away here, for  $\delta(p)$  value, so that path with minimum transmission delay is selected by the application.

#### 4.5.5 Selection Procedure

Selection of path is done in 5 steps. The logical scheme of the algorithm is shown in Figure 4.6.

1. When the data is available for transmission at Rover, it calculates the weightings  $\omega_1, \omega_2, \omega_3$  for each path according to the selection rules
2. Rover then calculates the cost for each path by applying the cost function and select the path with low cost for transmission
3. Check the buffer space availability and link availability period for the selected path

according to the following equations 4.13 4.14 using the algorithm 7.2.

$$\beta_i^j(t) > \sum_{i=1}^N P_{len}^i \quad (4.13)$$

$$\zeta_i^j(t) > TT_i^j \quad (4.14)$$

Where  $\beta_i^j(t)$  is buffer space availability at nodes  $j$  at the time of transmission should be greater than length of total packets ( $N$ ) to be transmitted and the link availability period (pass duration)  $\zeta_i^j(t)$  should be greater than transmission time  $TT_i^j$ . In fact,  $TT_i^j$  is the ratio between the total packet size (expressed in bit) and the link capacity in [bit/s] available in link between node  $i$  and its neighbor  $j$

---

**Algorithm 4.1** Logic for path indexing

---

Sort  $\text{cost}C = \{c(p_j) < c(p_{j+1}) < c(p_{j+2})\}$ , where  $c(p_j)$  is cost of path  $j$ ,  $j = \text{index of the path}$ ,  $\text{Path}=0$ ;

Let  $N = \text{total packet size in bits}$ , and  $R = \text{data rate}$ ;

**while**  $j \leq 3$  **do**

**if**  $\{ \text{timeWindow}(p_j) \geq \frac{N}{R(p_j)+Mrg} \ \&\& \ N \geq \beta(p_j) \}$  **then**

$\text{Path} = p_j$

**return**(Path)

**else**

$j = j + 1$  (increment the index)

**end if**

**end while**

**return**(Path)

---

4. If the verification in step 3 passes then Rover use the selected path, else it repeats the step 3 for next closest path

5. selected path is used for transmission of data bundles

As it is assumed that, Rover knows the power generated from the solar panels at the given moment of time, after receiving hailing message from the Lander/Orbiter, Rover arranges the path according to their cost in table  $C(P_j)$ . It can be considered that this table is specific to each model of Rovers. The table is read for three times, at most. If the link availability period is larger than the time necessary to transmit a  $N$  bundles, then whole bundle( $n$ ) are transmitted with a data rate (DR) only if buffer space is

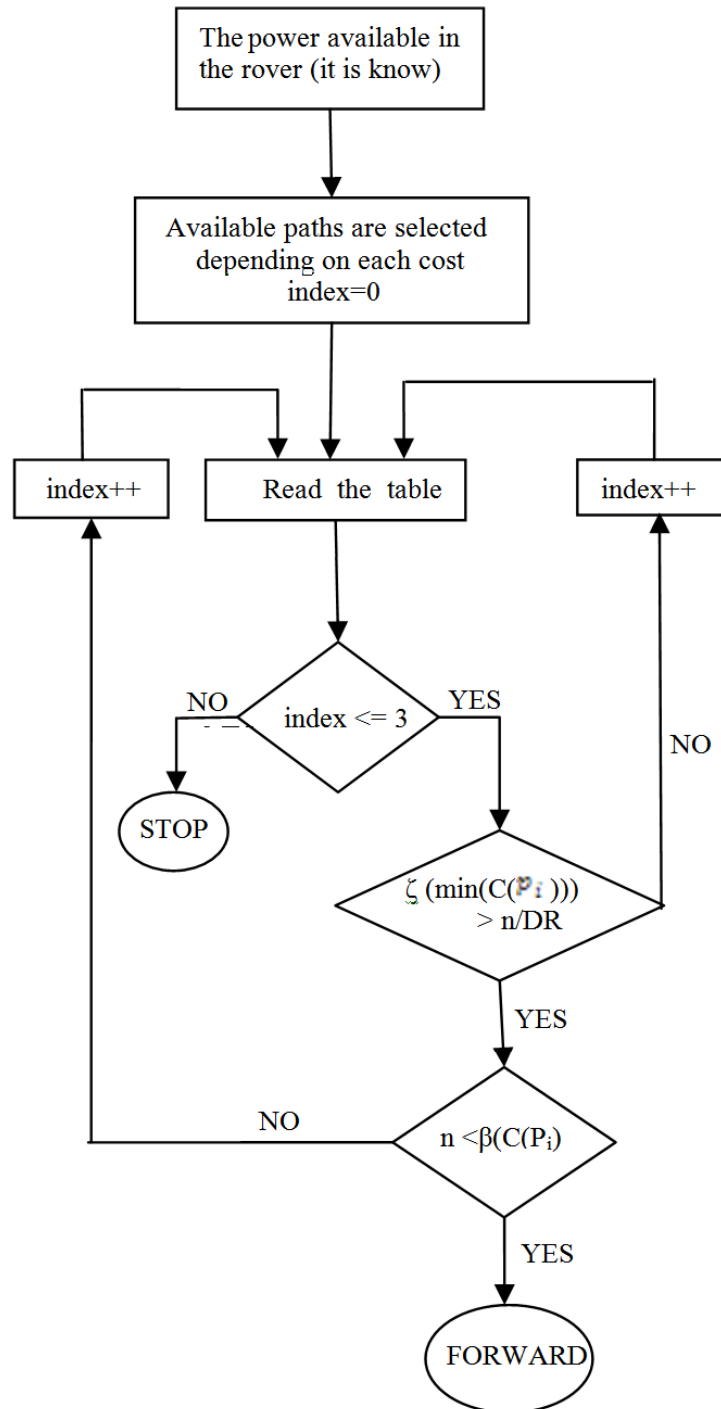


Figure 4.6: Logical scheme of the algorithm

available to accommodate entire bundle ( $\beta$  available space in the receiver equipment) see Algorithm 7.2. When conditions are satisfied, the algorithm is applied and it go to forward state where transmission is started. In the case of either one of the conditions is not , the algorithm go to the cost table where the index is incremented to select the link with the next minimum cost. Again the verification of link availability and buffer space is conducted before selecting the path. If no satisfactory result is obtained, then algorithm will stop and wait a predefined period of time until a new broadcast signal received by the Rover and new cycle for verifications is done for each path, which will redefine the cost table.

## 4.6 Implementation and Analysis

The algorithm was implemented in Matlab with a function that returns the path selected by the rover's communication system, as the result. In order to understand the behavior of the algorithms, the size of the message is fixed, there by buffer requirement; average value of power required for transmission in each path (data gathered using Figure 4.1). And the data rate is chosen according to power availability in the system. Eight orbital pass with fixed time period for communication (data gathered from ephemeris table is considered. Finally, a simple two state error model is used to define error rate in each path and the weightings for each parameter are calculated using the cost function.

Since the research can be considered for layer one and two of the networking stack, the values have been approximated for the equipments that were taken as models and as references. Our intuition is that the Rover will choose Rover-Lander link, since it is available almost all time and required less power, where both parameter have greater impact in cost function. It will select Rover-Orbiter link only if there is not enough space in the Lander's buffer. The proximity links appear to be the favorites since it required less power and offer less delay. Also, since Lander use two different frequency for communication between rover and orbiter, it will be able to communicate with Orbiter as well as the Rover at the same time without interference. The power-data rate factor of the link is calculated as transmission power divided to data rate. The smaller is the value; the better is the path (due to the consumption of power versus the data rate).

The link Orbiter-DSN can easily be considered like the Rover-DSN considering their distance. The significant difference is that the transmission part is supported by the

Table 4.9: Simulation Parameters

Parameter	Value
Total Packet Size	[5 - 10Mb]
Orbiter Used	MRO
Contact Duration	10 min
Contacts	near 4 – 5 AM LMST near 4 – 5 PM LMST
Power/Data Rate Factor	Rover-DSN 3.5 W/Kbps Rover-Lander 0.210937 W/Kbps Rover-Orbiter 0.140625 W/Kbps
Bit Error Rate	Rover-DSN $10^{-4}$ Rover-Lander $10^{-6}$ Rover-Orbiter $10^{-5}$
Propagation delay	Rover-DSN Path $0.8AU$ Rover-Lander Path $0.3AU$ Rover-Orbiter Path $0.2AU$
Buffer Space	Rover 5Mb Lander 8Mb Orbiter 12Mb

Orbiter with higher power availability than power limited Rover. The path to DSN, as it was expected, is the most expensive (the biggest value for the power-data rate factor) and it is selected as an opportunity only if there is no Orbiter in the visibility or/and the buffer space in the Lander is full. For simulations, it was considered the MRO as the Orbiter in the scenario. The orbiter has two contacts, near 4-5 AM LMST and 4-5 PM LMST, each is about 10 minutes duration. Rover-DSN link is available all time (Rover is assumed to the side of the planet oriented to the Earth). The global coverage on the surface of the planet Earth is assured by the three complexes of antenna (Canberra, Madrid and California)-especially by the 70m-antennas. Simulation parameters are shown in the table 4.9.

In order to understand the impact of each rules in algorithm, first simulation were done considering the decision based on power constrain. Simulation is done for 1 Mars day (sol) from 6hr to 18hr. Result of the simulation is shown in the Figure 7.6. From the graph as expected it can be seen that, Rover-Lander link is selected for quite long period. But due to the buffer over flow in the Lander, at 11hr LMST Rover select DTE (high power) link ahead of Rover-Orbiter (which is not visible). Then when orbiter is available at 17hr LMST, algorithm recalculate the cost function (As halling message is transmitted between Rover-Orbiter) and select Rover-Orbiter instead of high power DTE link as expected.

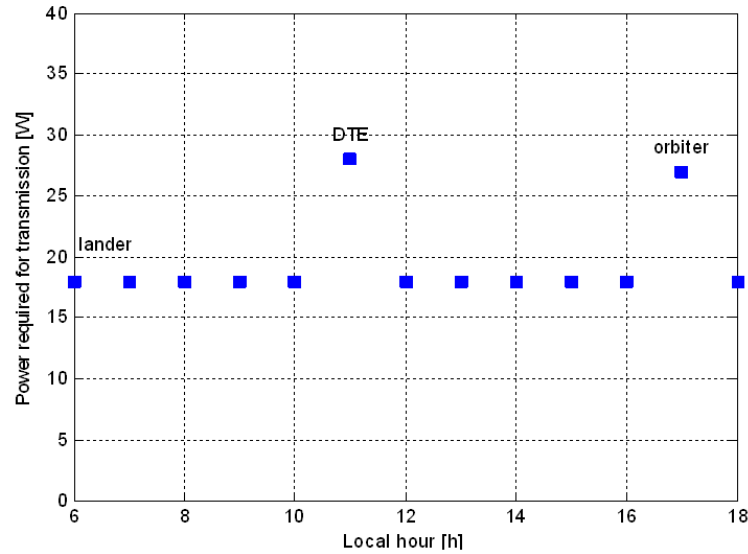


Figure 4.7: Algorithm results due to the power constraints for a sol

Relay UHF link seems to be the most favorable paths due to its energy efficiency, but there are some scenarios where both Rover-Lander link (buffer space) and Rover-Orbiter link (no pass) are not available, where algorithm alternatively select DTE link for communication (if power available), which make rover more autonomous. From the above result we can infer that Rover-Lander link is used 85% of time when compared with Rover-Orbiter (8%) and DTE (7%), which proves that relay link are more effective.

For the next simulation, delay sensitive application scenario is taken. Though presently in deep space network only file delivery application uses, where delay is not a big issue, but with the recent advance in application domain where audio and video applications are proposed have higher degree of impact on delay. Taking those applications in to consideration the algorithm rules are formulated. Here both propagation delay and storage delay is considered for calculation. Though the propagation delay is very less for both Lander and Orbiter; they have quite higher storage delay in comparison with DTE link (no storage delay). This is due to the scheduled contact between Lander and Orbiter to transfer data and Orbiter still have to wait for scheduled contact with earth station to transfer data; hence higher storage delay. In the simulation buffer space in both Lander and Orbiter set same as that of Rover in order to understand the impact of the buffer space. Figure 7.2 illustrate the simulation result for this scenario for one sol. As we can see from the graph, DTE link is selected ahead of both relay links when delay is considered as application QoS parameter. But Rover-Orbiter is selected though

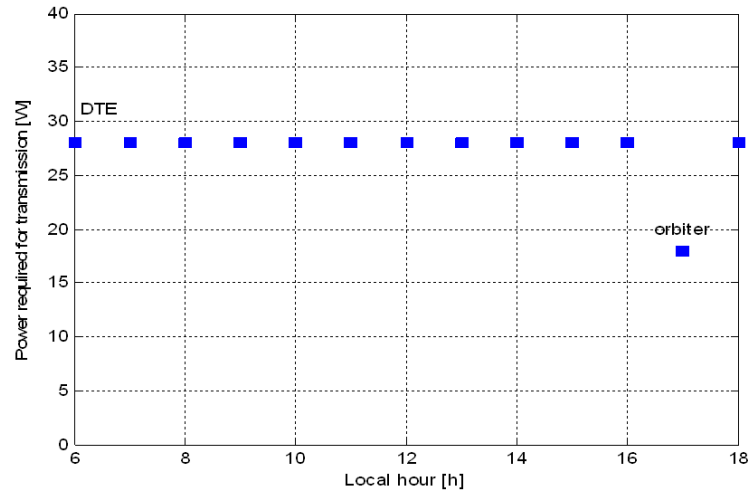


Figure 4.8: Algorithm results due to the delay constraints for a sol

they have high storage delay is due to power limitation in Rover which forced antenna selection algorithm to look for alteration. The cost function is recalculated when Orbiter is visible to Rover (kind of an update). This we can see from the graph at 17hr. Rover used around 92% of time DTE antenna when delay parameter is considered. In general in Mars communication DTE link is a least preferred link, thus selection algorithm performance is very optimal in this delay constrain scenario.

The Rover's schedule for a sol depends upon the mission of the robot. At the give time of day, for implementing this algorithm, Rover should know the available power from the solar panels and the link parameters (known from link design) also, standard values for the weightings should exist corresponding to the equipment. Since the DTE link is the less-efficient link, the solution for transmitting messages with optimal delay is to design a full-coverage orbiters constellation (ideal situation; like the GPS one for the Earth, but with another scope) also, these link (power-efficient) will be used as delay-efficient links (in the situation that the orbiters will transmit immediate after receiving the message). The surface equipments will use DTE link only for emergency situations, efficient for the moments when the DSN would not be able to control the Martian robots (the link will be selected automated). Figure 4.9 illustrate the selection percentage of each antenna for two different scenarios. In case of power constrain scenario the results seems to be closer to reality (which is happening in case of spirit and opportunity now). For the second delay constrain scenario, where DTE link is preferred seems quite unnatural, since DTE link is more expensive in case of power (it is desired to be used only for emergency cases) thus less optimal for delay sensitive application.



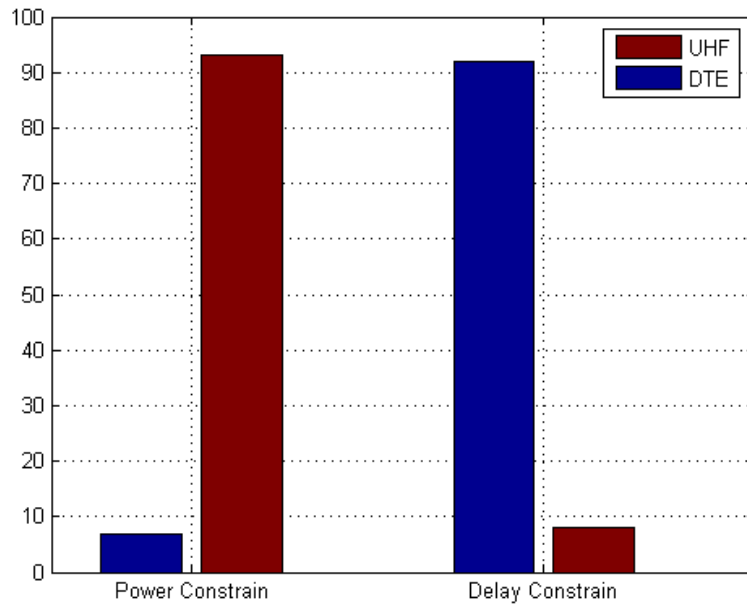


Figure 4.9: Antenna Selection % upon two different rules

In the next scenario, application with minimum delay with no enough power available for DTE link is considered. When simulated, the algorithm returns 0, where Rover will do nothing and wait until power level in system is up for any possible transfer. Since the algorithm implies to check two conditions, the power requirement and link availability, if any one condition fails the algorithm returns 0. Thus the algorithm proves to be efficient only for best effort traffic.

Rover's antenna selection intuitively leads to the selection next hop for transmission of data like Lander, Orbiter or Ground station. This study anticipates developing a new source routing protocol. Source-based routing means that the source specifies how the data is to travel through the network. In Mars scenario, Rover selects the next hop with which it transmits the date to the Earth. In next section describe the proposed source routing called "Resource Aware routing".

## 4.7 Resource Aware Routing Protocol

A unicast source routing protocol is proposed in this section. The three functional blocks of routing protocol is shown in the Figure 7.7. The aim of the routing protocol is to achieve higher delivery ratio with minimum delivery latency using all pos-

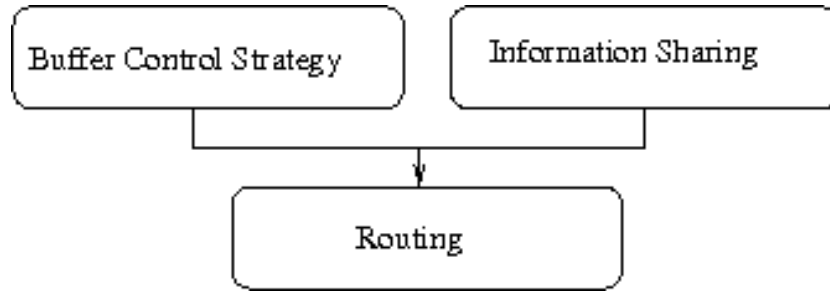


Figure 4.10: Functional Blocks of Routing Protocol

sible network knowledge available and also taking the buffer size availability of nodes into consideration. Each module of the proposed routing protocol is elaborated in the following subsections.

#### 4.7.1 Buffer Control Strategy

Most of the source nodes in the Mars network are resource limited. In store and forward networks like Mars surface network, when if a end-to-end path is not available then instead of dropping the packet, the nodes store the packet in the buffer until it finds destination or some other node. By doing this for long time, if the storage of nodes becomes full, then it has to drop a packet in the buffer to make room for new packet or deny the incoming packet. There are multiple policies to drop or deny the packet when buffer is overloaded. In the protocol two different policies for the two data type class i.e. real time traffic and best effort traffic are used.

In case of real time traffic (e.g. telemetry) which is time sensitive traffic; when a new packet arrives at the node and if buffer is full, then oldest packet from the buffer is dropped, i.e. the packet which has stayed for longest amount of time in the buffer among all the stored packets. The reason for this is, oldest time sensitive packets which stay in buffer for long become obsolete and of no use and also there is very low chance of forwarding the same in future.

In case of best effort traffic (i.e. Image data) the priority based buffer control strategy ([69]) is adopted. In which, the data stream of image is prioritized according to an effective image processing algorithm, where the most significant segment of data is given higher priority than least significant data and packets are stored in an ordered list in the buffer according to their priority level. When a new packet arrives and if the buffer is full, new packet is inserted into the ordered list according to its priority level and packet with least priority in the list is deleted. In this way, the strategy ensures that

only packet with lower priority is discarded and higher priority packets which are more than enough to constitute the full image are reliably transmitted.

#### 4.7.2 Information Sharing and Antenna Selection

In sparse, intermitted networks like Interplanetary Mars surface network, knowledge of network information like network connectivity, orbital pass duration, possible propagation delay, resource availability etc. help to make good routing decisions. In the proposed routing protocol these network information are shared between nodes when they encounter each other. This information's are known by the nodes with the history of their past encounter (i.e. pass duration which is constant value for all passes are build with help of past encounter). Whenever two nodes are within the communication range (especially surface nodes and orbital nodes) they share following information before starting data transfer request: (i) buffer space availability which is difference between occupied buffer and total buffer size of the node,  $\beta_i^j(t)$  is the value of  $\beta_{total}^j - \beta_{occ}^j$ , valid at the time instant t for node j notified to its neighbor node i. (ii) link availability period which is the expected time duration in which both nodes of a edge (link) have communication availability. As observed in the previous case;  $\zeta_i^j(t)$  is the value of the link availability period from the time instant t for node j and its neighbor node i. (iii) Next Contact time  $N_i^j(t)$  is expected time duration after which a pair of node encounter each other, which is calculated from past history of encounter between the pair i, j. After sharing these information in their first encounter, each nodes create table of these values, and calculate possible propagation delay ( $\delta$ ) between them (using expected RTT), and also the source node calculate Power available in it at time instant t and called it as  $P(t)$ .

As we know, in Interplanetary Mars surface network, source nodes are predominantly surface nodes (i.e. Rover/Landers) where, they are equipped with two communication system namely, X-Band Direct-to-Earth (DTE) system designed to operate at lower data rate with higher reliability and higher-data rate UHF radio communication system to operate in near proximity distance. These surface nodes can be modeled as typical DTN nodes which are best described as multigraph (a graph where vertices may be interconnected with more than one edge). Thus selection of a edge means selection of transmission antenna (DTE or UHF). In the previous work [67], a cost based Antenna selection algorithm for such Mars Surface nodes (Rover) is proposed, by which it select a most resource efficient edge. In this work, same principal is adapted to define the power generated in the nodes.

### 4.7.3 Routing

Based on the type of application data and its QoS requirement (table 4.8), different forwarding rules are proposed to use resources effectively. If for instance, surface nodes have packets to send to destination (e.g. Rover wanted to send terrain image to ground station), it selects the next hop node according to its resource availability (power, buffer) and QoS (best effort, maximum throughput, minimum delay) requirements of the application. Three sets of rules were proposed for three different applications Table 4.8 in the following Algorithms 7.3, 7.4, 7.5.

#### Network Model

As the protocol is designed for Mars surface network, the network consists of 4 different types of nodes: mobile surface nodes (Rover), fixed base station nodes (Landers), resource-rich satellite nodes (Orbiters), and fixed ground station nodes (DSN). Let  $M = \{m_1, m_2, \dots, m_n\}$  be the set of mobile nodes in the network.  $S = \{s_1, s_2, \dots, s_n\}$  be the set of all fixed surface nodes, such that  $SM = M \cup S$ . Let  $O = \{o_1, o_2, \dots, o_n\}$  be the set of all orbital nodes. And let  $G = \{g_1, g_2, \dots, g_n\}$  be the set of ground stations on Earth. Mobile nodes ( $m_i$ ) are resource-limited in terms of power and buffer, which have non-deterministic random movement, with at least one fixed surface node ( $s_i$ ) in the communication range. Fixed surface nodes ( $s_i$ ) which have comparatively high resources than mobile nodes act as base station nodes in the network. Orbital nodes ( $o_i$ ) have fixed orbital movement with periodic link connections with nodes in the  $SM$  set. They usually have higher resources than  $SM$  nodes. Finally, the ground station nodes ( $g_i$ ), which are usually the fixed sink nodes on the Earth surface, have a scheduled link with orbital nodes ( $o_i$ ) and a DTE link with one of the mobile nodes ( $m_i$ ) source nodes.

#### Algorithm 1

When the source node  $m_i$  has data to be transmitted with high reliability to the destination (e.g. Rover wanted to transfer telemetry data to Earth ground station), that is when application data requires a minimum loss percentage as a QoS parameter (algorithm 7.3) then, the source nodes  $m_i$  check its power level available in the system. Where the source nodes (Rovers) measure the power level in the system for 1 sol (1 Mars day) between 6hr-18hr using the following Equation:

$$P_t = G_h * \eta_{solar} * A_{solar} \quad (4.15)$$

**Algorithm 4.2** Protocol Logic for traffic which require high reliability

Let mobile node  $m_i \in M$  have application data which required to be send reliably to a ground station node ( $g_i$ )

**if** {sufficient power available  $P_{rq}^{m_i-g_j}(t) < \rho$  in  $m_i$  for DTE} **then**

Select a node  $g_j \in G$  as next hope which is in the line of site of  $m_i$  (path 1).

**else if** power is not sufficient  $P^{m_i}(t) \leq \rho$  **then**

Select a node  $o_j \in O$ , as the next hop which has sufficient buffer space availability  $\beta_i^j(t)$  and link availability period  $\zeta_i^j(t)$  (path 2).

**else**

Select a node  $s_j \in S$  as the next hop which are in the communication range of the source node  $m_i$  with sufficient available buffer space availability  $\beta_i^j(t)$  and have minimum next contact time  $N_i^j(t)$  (highest likelihood to encounter orbital node in near future) (path 3).

**end if**

where,  $P_t$  is power generated by the solar panel at time instant  $t$ ;  $\eta_{solar}$  is efficiency of converting solar radiation into power;  $A_{solar}$  is the area of the solar panel and  $G_h$  is global irradiance on the surface. Due to the variance of solar irradiance throughout the day, the output of solar panel has similar effect in power generation. The result is a window of operation, in which the rover has enough power to perform tasks. Knowing the power level available at the system, source nodes  $m_i$  select the Direct-to-Earth (DTE) link (using X-Band Antenna) which is the Earth ground station node  $g_j$  as the next hop, if and only if it has direct line of sight with  $g_j$  and satisfy the following Equation:

$$P_{rq}^{m_i-g_j}(t) < \rho < P^{m_i}(t) \quad (4.16)$$

Where  $P_{rq}^{m_i-g_j}(t)$  is the power require for the transmission in DTE link;  $\rho$  is power threshold, beyond which link is unreliable (higher error rate due to lower transmission power) and  $P^{m_i}(t)$  is power available at node  $m_i$  at given instance  $t$ . If the Equation 4.16 is not satisfied then source node  $m_i$ , select a orbital node  $o_j$  from the set  $o_j \in O$  as the next hop if it satisfies the following conditions;

$$\beta_i^j(t) > \sum_{i=1}^N P_{len}^i \quad (4.17)$$

$$\zeta_i^j(t) > TT_i^j \quad (4.18)$$

Where  $\beta_i^j(t)$  is buffer space availability at orbital nodes at the time of transmission should be greater than total length of the packets ( $N$ ) to be transmitted and also the

link availability period (pass duration)  $\zeta_i^j(t)$  should be greater than transmission time  $TT_i^j$ . In fact,  $TT_i^j$  is the ratio between the total packet size (expressed in bit) and the link capacity in [bit/s] available in link between node  $i$  and its neighbor  $j$ , which can be defined as in Equation 4.19.

$$X_i^j = \begin{cases} X_i^j : j = \arg \max \beta_i^j(t), j \in [1, O] \\ X_i^j : j = \arg \max \zeta_i^j(t), j \in [1, O] \end{cases} \quad (4.19)$$

If both the DTE node or orbital nodes are not available then source nodes select any Fixed surface nodes ( $s_j \in S$ ) which is in its communication range and satisfies the Equation 4.17 and also has minimum next contact period with any orbital nodes toward destination as defined in Equation 4.20.

$$s_j = \min_{j \in [1, S]} N_j^i(t) \quad (4.20)$$

On the other hand, if the buffer of source nodes senses any congestion, and both DTE and Orbital nodes are not available then it selects any fixed node as next hop, which satisfies just Equation 4.17 and not necessarily Equation 4.20, to avoid source buffer congestion.

---

**Algorithm 4.3** Protocol Logic For Best Effort

---

**if** the traffic class is of best effort **then**

Select a node  $s_j \in S$  as the next hop which are in the communication range of the source node  $m_i$  with sufficient available buffer space  $\beta_i^j(t)$  and have minimum  $N_j^i(t)$ .

**else if** orbit pass is available to source node **then**

Select a node  $o_j \in O$ , as the next hop which has sufficient available buffer  $\beta_i^j(t)$  and link availability  $\zeta_i^j(t)$  period.

**end if**

---

**Algorithm 2**

Second kind of traffic in the network is best effort traffic, which are mostly the image data sent from Rover to Earth station. For this type of traffic class source node (Rover) selects next hop according to the set of rules defined in Algorithm 7.4, where the source node  $m_i$  selects any fixed surface node  $s_j \in S$  which is in the communication range as next hop if it satisfies Equation 4.17 and Equation 4.20. Else it selects an orbital node  $o_j$  such a way that orbital node satisfies both Equation 4.17 and 4.18. Hence in the case of best effort traffic DTE link (path 1) which requires higher power is not used, by this

was power resource is guarded. During congestion in any node in the network for best effort traffic, priority base buffer management policy is applied (section 3.4).

---

**Algorithm 4.4** Protocol Logic For time sensitive traffic
 

---

**if** the traffic class require minimum delay QoS parameter **then**

Select a node  $o_j \in O$  , as the next hop which has sufficient available buffer  $\beta_i^j(t)$  and link availability  $\zeta_i^j(t)$  period.

**else if** no orbit pass available **then**

Select a node  $s_j \in S$  as the next hop which are in the communication range of the source node  $m_i$  with sufficient available buffer space  $\beta_i^j(t)$  and have minimum  $N_i^j(t)$ .

**else if** sufficient power available  $P_{rq}^{m_i-g_j}(t) < \rho$  in  $m_i$  for DTE **then**

Select a node  $g_j \in G$  as next hope which is in the line of site of  $m_i$ .

**end if**

---

**Algorithm 3**

Though the delay in interplanetary distance is inevitable, for time sensitive data traffic, we have proposed to select orbital relay path considering the orbital dynamic and buffer space availability which minimize both propagation and buffer storage delay considerably (algorithm 7.5). Thus source node select orbital nodes as next hop which satisfy both Equation 4.17 and 4.18. With the assumption that orbital nodes are always visible to the ground stations which mostly are the final destination nodes, provides minimum delay among other possible links in the network. If the orbital node pass is not available then it select a fixed surface nodes which satisfy Equation 4.17 and 4.20. Here next contact time  $N_i^j(t)$  is vita parameters since  $N_i^j(t)$  value is directly proportional to buffer storage delay. And if the fixed surface nodes do not have higher  $N_i^j(t)$  then source node select ground station nodes  $g_j$  which is in line of sight as next hop, if it satisfy the Equation 4.16. By this was its can minimize the overall latency in transmission.

## 4.8 Functionality Analysis

In order to understand the implication of our routing algorithm, we have implemented the routing functions in Matlab. The routing function returns a next hop selected by the mobile nodes in the network according to the given scenario. To understand the behavior of the algorithm, we have predefined the size of the messages

Table 4.10: Simulation Parameter

Parameter	Value
Data Rate	10 Kps
Packet Size	1000 bytes
Power Required - DTE Link	240W
Buffer	Exponential decay

(packet), data rate, and bandwidth. We have implemented separate program for each algorithm. Program runs the simulation of one Mars day (6hr - 18hr) with 1hr interval between each transmission is assumed. Program at give instant of time (say 6hr for example) calculate available power at rover nodes using equation 4.1. And power required for transmission in each given link is calculated from the graph 4.4(b) by fixing the data rate. Simulation parameter is listed in the table 4.10

#### 4.8.1 Next Hop selection % for different scenarios

The graph in Figure 7.8 shows the % of selection of next hop for each algorithm (best effort, reliable transfer, and time sensitive traffic). It can be seen from the graph that in the traffic class which require higher reliability (Algorithm 7.3), the source node select the Ground station node as next hop with high percentage when compared to other 2 nodes. Orbital nodes or fixed node are selected as next hop only if the power available is less than threshold value. Thus these 2 nodes are selected either at beginning of the day (sol) or at end the day when power available is comparatively low in source nodes. On the other hand for best effort traffic, fixed nodes are predominantly selected as next hop unless the buffer is full. And in the time sensitive traffic, orbital node is selected as next hop with high percentage compared to fixed or Ground station nodes, which is done to avoid buffer delay and power constrain in the system. Thus it can be argued that routing decision are taken according to resource constrains.

## 4.9 Performance Analysis

The protocol is implemented in version of 2.30 of Network Simulator (ns2) [62]. Protocol is implemented as a table driven source routing, where consistent and up-to-date routing information to all nodes is maintained at each node. Interplanetary network consist of fewer sparely connected nodes. For this reason, it can be assumed that the sender knows the topology of the network and hop-by-hop route to the destination.



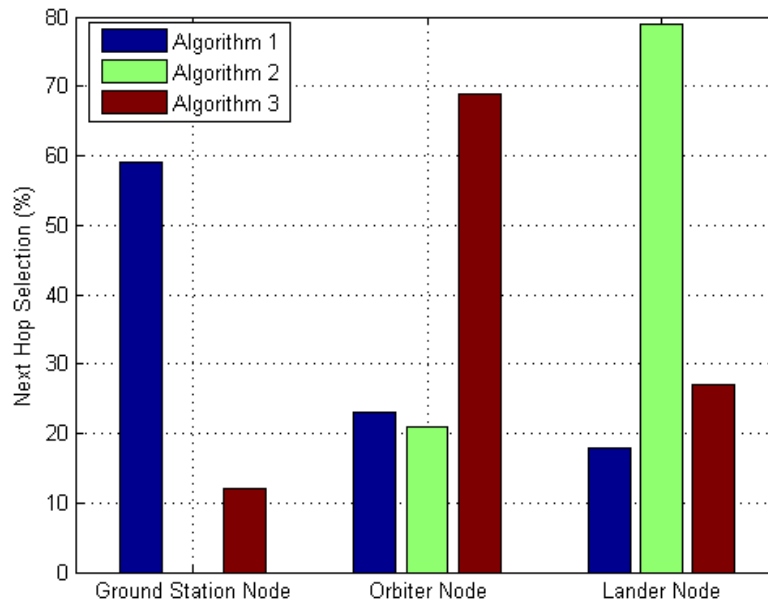


Figure 4.11: Next Hop selection % for each Algorithm

These routes are stored in a route cache. The data packets carry the source route in the packet header. When source node attends to send a data packet to the destination, routing agent dynamically determine appropriate route according to data traffic type and source resource.

#### 4.9.1 Simulation Topology

For the performance analysis, a simple interplanetary topology is considered where the topology consist of two source Rover nodes, which are mobile and follow random waypoint mobility model with very slow speed and more pause time. Rover mobile nodes are almost always in the converging range of fixed Landers nodes in Mars planetary network. For this reason, in the simulation topology the concept of one fixed nodes in the proximity of a mobile nodes is used. In other words, each mobile node in the network has a fixed node in the communication range throughout the simulation. Both fixed Lander nodes and Rover nodes are communicated back to Earth ground station node using two orbiter nodes which had eight communication passes for each Mars day (sol) for a fixed period of time. Buffer size of fixed and orbital nodes is defined three times greater than that of rover mobile nodes. For each traffic class, data rate of 4pks/sec with each packet of 512 bytes is selected. All the graphs are results of an

Table 4.11: Simulation Parameters

Parameter	Values
Number of Nodes	8
Number of Source	1
Packet Size	512 bytes
Traffic rate	4 pack/sec
Propagation delay	
Interplanetary Backbone	20 sec
Propagation delay	
Planetary Nodes	0.5 sec
Transport protocol	UDP
Simulation Length	1000s

average of 5 scenarios with different initial conditions for the mobility of the nodes.

#### 4.9.2 Performance of the protocol for best effort traffic

The first set of simulations is conducted for best effort traffic; the purpose is to have a basic overview of the performance of protocol under very general condition. The experiments are conducted by varying the visibility duration, the local buffer size within nodes. The common parameters of these simulations are listed in Table 4.11. To emulate the characteristics of interplanetary networks, we have modified the ns-2 simulator to manage the link up/down schedules. In this simulation, each Lander-Orbiter link experiences a series of randomly generated up/down periods. The visibility duration is the up time between two down time periods. For instance, if the down time of a link spans  $[t_1, t_2]$ ,  $[t_3, t_4]$ , and  $[t_5, t_6]$ , then  $[0, t_1]$ ,  $[t_2, t_3]$ , and  $[t_4, t_5]$  are the visibility durations of this link. In this test, we vary the total length of the visibility durations of each link from 10% to 90% of the overall simulation time randomly. Figure 4.12 4.13 shows the results. It is observed that: fewer packets could be delivered to the destinations when the percentage of visibility duration becomes small, i.e., the lack of network connectivity becomes more severe. Natural inference is that when nodes have more local storage, more bundles can be delivered to receivers. The Packet delivery time (the time interval required to complete the packet delivery to destinations.) increases when the size of local storage increases. This is because with more local storage, some packets that are dropped in the case of smaller storage could be buffered for longer time until they get a chance to be forwarded. Therefore, some of the delivered bundles may have been held in the network and experience longer queuing delays before reaching their destinations. Thus in the case of best effort traffic, routing protocol behaves like

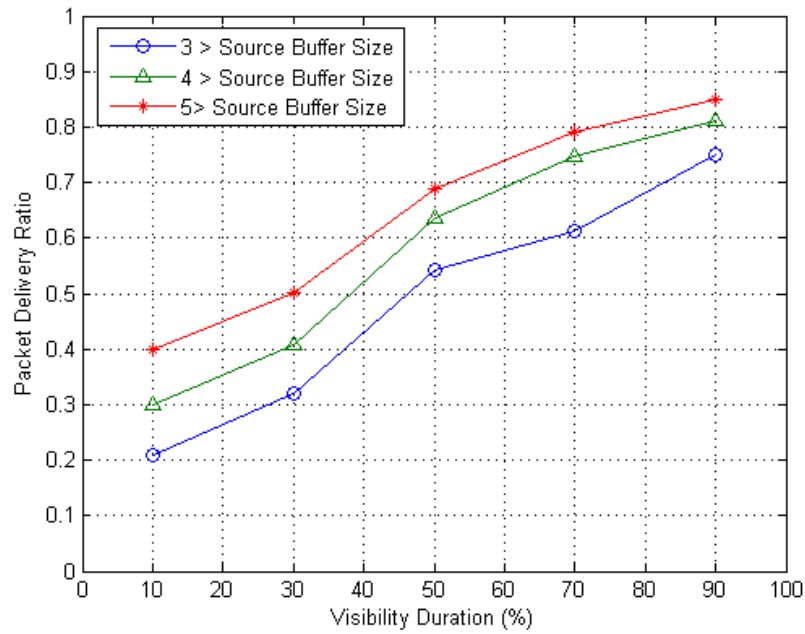


Figure 4.12: Delivery ratio with different buffer size, varying visibility duration %

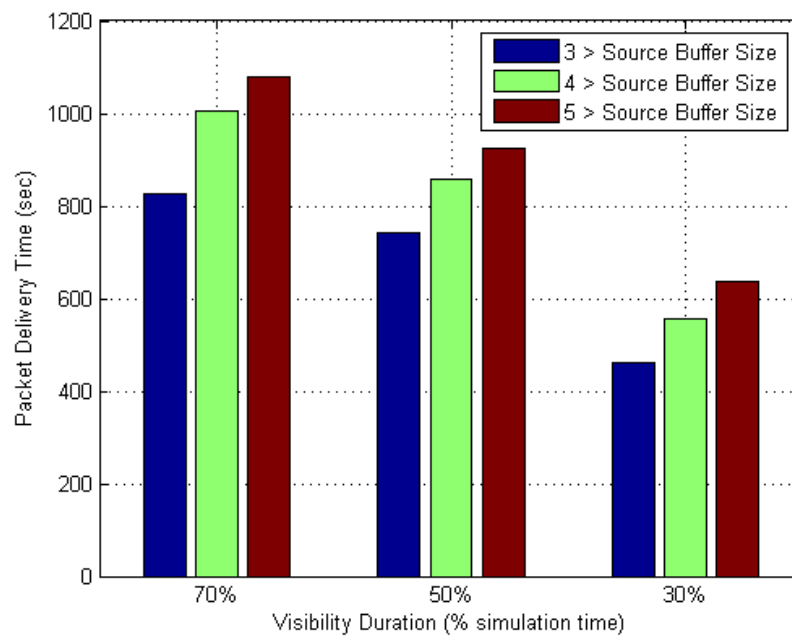


Figure 4.13: Packet Delivery time with different buffer size, varying visibility duration %

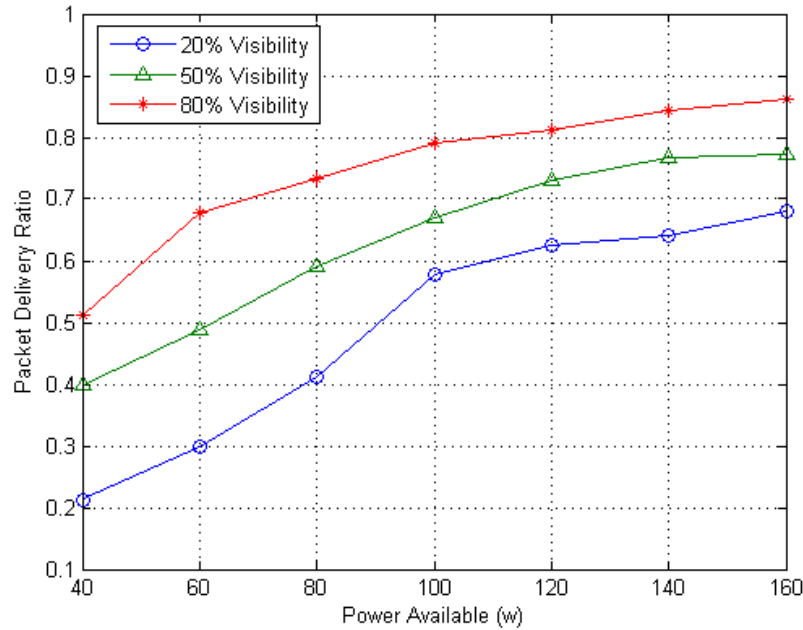


Figure 4.14: Delivery ratio with different visibility duration %, varying power level

direct delivery router, where it just forward the packets if it has enough buffer space.

### 4.9.3 Performance of the protocol for throughput sensitive traffic

It is interesting to see the impact of the power in source node over the routing decision. As mention earlier in the algorithm, routing decisions are made according to resource level in the source node. Our intuition in this experiment is that when power level in source node increase, the packet delivery efficiency increases. In this simulation, the percentage of the visibility duration is fixed in three level, worst (20%) average (50%) and good (80%) of the total simulation time and vary the power level in the source node (rover) with fixed buffer size. As shown in Figure 4.14. when power level is high (between 12hr-16hr in sol) and visibility duration percentage is high, high delivery percentage is achieved. This is because source nodes have options to choose between DTE and Orbital link. It uses DTE link to transfer packets when power level is high (no orbit pass) and when power is comparatively low its uses Orbital link to transfer packet to the destination. When visibility duration percentage is low with low power level in the source node delivery ration is less then optimal. And the migration of selection from Orbital link to DTE link is seen when power level is changed from 80 to 100(w) with visibility duration is 20%, where delivery ratio increase considerably. But protocol

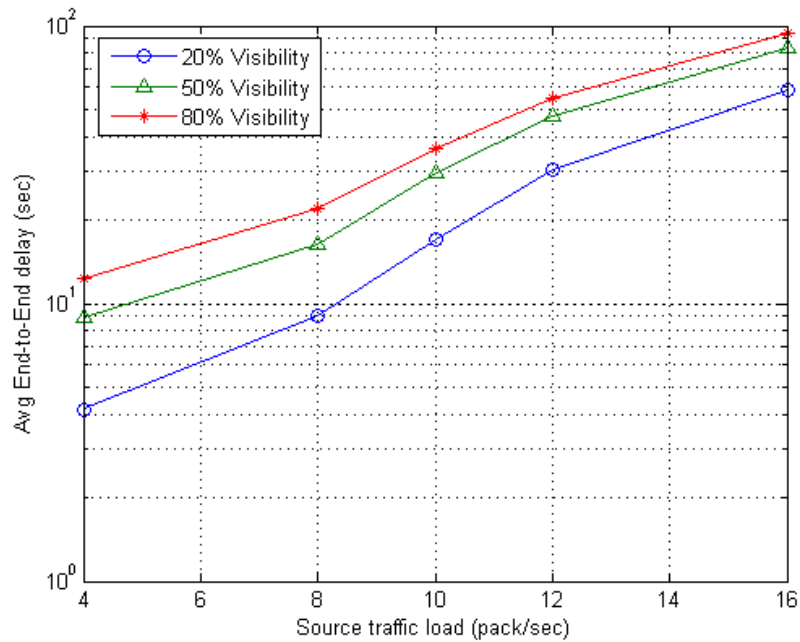
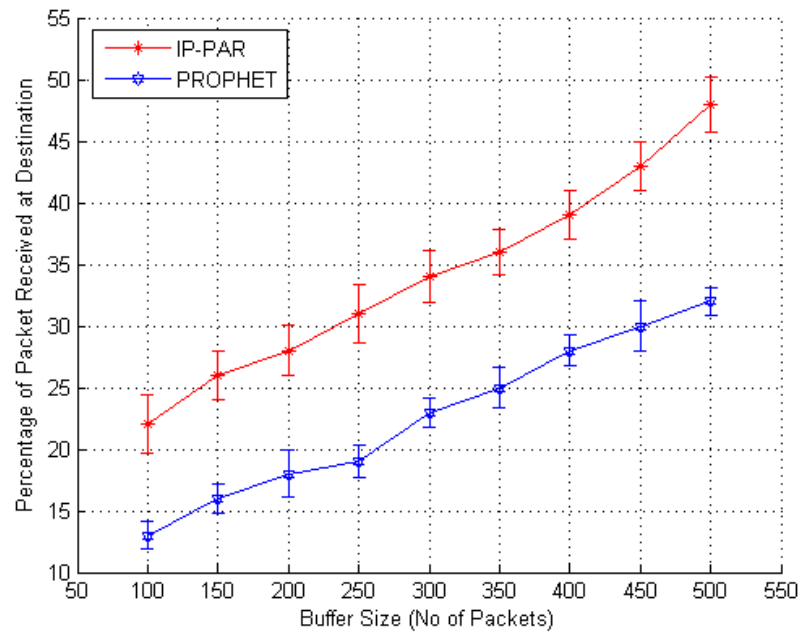


Figure 4.15: Delay, link, buffer

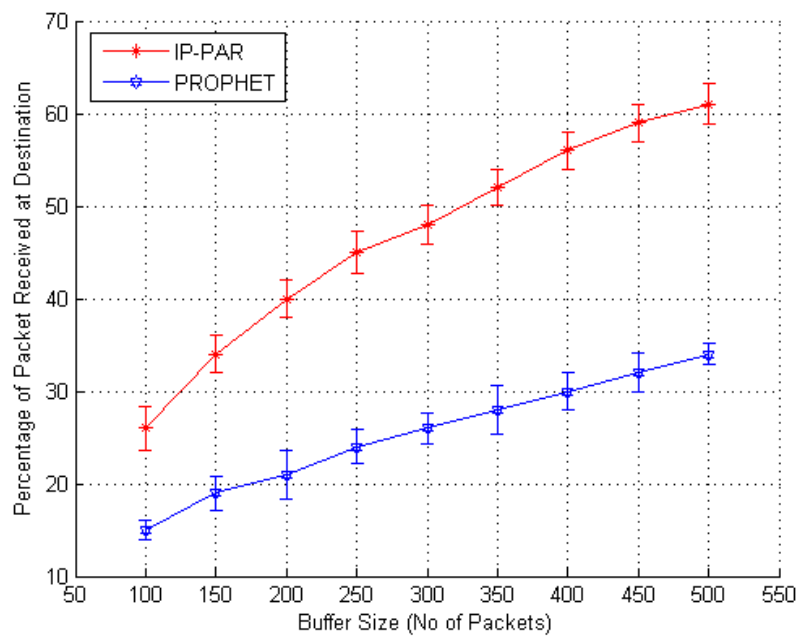
performance is good when visibility duration is average or above average in almost all power level, since protocol uses effectively the power and link available for transmission.

#### 4.9.4 Performance of the protocol for Delay sensitive traffic

Providing minimum end-to-end delay in interplanetary network is always challenging. The protocols make a simple effort to minimize the possible delay (congestion delay, and queuing delay). So it is interesting to analyze protocol effectiveness in delay sensitive traffic. In this simulation, we fix the percentage of the visibility duration in three level, worst (30%) average (50%) and good (70%) of the total simulation time and vary the traffic load with fixed power level and high buffer size. The result in Figure 4.15 shows, protocol perform less than optimal in almost all three cases. However, their performances increase in term of delivery ratio. By studying the simulation traces, we find that when visibility duration % is low, protocol tend to select fixed surface node as next hop, with the fixed high buffer size, they store long packets before the link between fixed node and orbiter is available, which increase the queuing delay, with directly increase end-to-end delay. When buffer get expired; (buffer congestion reached), they tend to select high delay DTE link, thus perform poorly. In average visibility duration % protocol use effectively the resources % and perform better than other cases.



(a) % of packet delivery vs Buffer Size with visibility duration is 70%



(b) % of packet delivery vs Buffer Size with visibility duration is 30%

Figure 4.16: Performance comparison with PROPHET protocol for Best Effort Traffic

### 4.9.5 Comparison with PROPHET

We compared the performance of our routing protocol against the probabilistic routing protocol called PROPHET [56]. The reason for selection of PROPHET is that, both protocols make decision on next hop node using the control information they have exchanged during the past encounter with other nodes. In this experiment, we fix the visibility duration to 30% and 70% and varied the buffer size in relay nodes with higher data rate. In this we have considered same topology of 8 nodes with simple best effort traffic. Our intuition is when the simulation time increases more packets are generated for a given buffer size and buffer at relay nodes are filled rapidly. This condition causes congestion at the relay nodes and this leads to packet losses which reduce the delivery ratio for higher data rate. PROPHET does not consider the buffer occupancy of the neighbor nodes, may select the next hop node which has no free buffer available or the buffer occupancy of which is very high, which cause higher data loss. Thus PROPHET observes higher data loss compared to our protocol.

Figure 7.9(a) and Figure 7.9(b) shows result of the simulation. Not surprisingly, both protocols can only deliver very few packets to the receivers when buffer is small. However our protocol perform fractionally better then PROPHET, since it use buffer resource effective by sharing information prior to the transmission, this is achieved with the cost of overhead due to information sharing. Based on our results, we can say our protocol perform optimal in the network average visibility duration, and which can tolerate overhead cost.

## 4.10 Discussion

In this work, we argue to make Mars Exploration Rovers more autonomous by embedding some intelligence in it. In this process, we have first examined the resource limitations of the communication system of Rover (Mass, Power, Energy) and characterize all possible communication links to Earth from Rover with these limitations. We showed selection of a communication link for the transmission of data back to earth is in turn selection of appropriate Antenna installed in Rover communication web. Second, we have proposed a cost base Antenna selection algorithm using which Rover autonomous select an antenna for communication according to its resource limitations. Finally, we developed the cost based algorithm into an effective routing protocol with which Rover can route all kinds of traffic autonomous according to its resource without any manual intervention from the Earth station.

We then presented Resource Aware Routing protocol, which is capable of routing different kind of traffic by using all possible network knowledge available and also taking the buffer size availability of nodes in to consideration for routing decision to achieve high delivery ratio with minimum latency. We show by using specific routing decision according to traffic type, achieves good performance, in terms of delivery ratio and delay. We also discussed buffer issues that have impact in performance, which is countered by effective buffer control policy. Finally the comparison of the proposed protocol with other probabilistic routing shows the edge in performance.

As future Mars missions depends on the ability of the Rover to operate autonomously in Mars Far-Term Communication Architecture (2.3), the proposed algorithm seems as a good approach to make Rover communicate with other entities more autonomous.



## Chapter 5

# Dynamic NAK Timer Algorithm for CCSDS File Delivery Protocol

CCSDS File delivery protocol is the main application layer protocol in interplanetary network. Many researches are going to improving the performance of file delivery protocol. This chapter elaborates the need for a dynamic timeout value for CCSDS File delivery protocol and proposed a new dynamic timer algorithm which adopts the timeout value of timers according to network conditions to achieve higher throughput by avoiding unnecessary retransmission of Meta or data PDUs.

### 5.1 Need for Dynamic timeout value

CCSDS File Delivery Protocol (CFDP) is an reliable application layer protocol design for interplanetary network, who's performance are highly challenged by the environmental constrains like limited bandwidth, high error rate and very long propagation delay. Compared with common terra and satellite communications, deep-space communication radio typically operates under meager, asymmetric bandwidth. The bandwidth capacities are asymmetric and fairly limited in the deep-space environment. The uplink channel (Earth to Destination) tends to have much lower bandwidth than the downlink channel (Destination to Earth).

Bandwidth asymmetry affects the performance of reliable CFDP, because CFDP rely on the feedback mechanism to guarantee reliable and smooth transmission. Though the bandwidth in the direction of data transmission is adequate and data can reach the receiver quickly, the feedback packet flows are congested at the bottleneck of the adverse direction and can't arrive at the sender in time, leading to the great degradation

of the whole performance. Though this constrain is addressed by CFDP, using selective Negative Acknowledgement (NAK), still an effective dynamic timeout mechanism is need to avoid unnecessary retransmission of NAK due to early or late timeout. Another constrain faced by deep space network is very long propagation delay. Under the environment of the long propagation delay, the throughput efficiency can be compromised in the form of unnecessary duplicate retransmission of an identical PDU. Unnecessary duplicate retransmission of the file-data PDU can occur if the timeout value of the NAK timer is set too small, the sender retransmits NAK before receiving data PDUs due to timer expiration, and if it set too large, there will be unnecessary retransmission of data PDUs before receiving ACK PDU from receiver. These constrain, shows the need for a dynamic timer which are able to adapt the timeout value according to the environment conditions in order to use the bandwidth effectively and to provide good throughput.

## 5.2 CCSDS File Delivery Protocol (CFDP)

CFDP is the CCSDS File Delivery Protocol [25], an international standard for automatic, reliable file transfer between interplanetary entities, built on the familiar Consultative Committee for Space Data Systems (CCSDS) [1] space data communication protocols. It is a delay tolerant protocol whose model of operations is fundamentally store-and-forward, much like e-mail that conveys files as attachments. The protocol as currently designed contains its own reliability mechanisms and does not assume an underlying retransmission capability. It presently operates point-to-point across a single link and contains and reliability mechanism that ensures that all of the pieces of the file are properly delivered across the link, with any missing pieces being automatically retransmitted.

### 5.2.1 Design Concept

In interplanetary network, due to intermitted connection, any CFDP entities pair which have file to exchange may not able to communicate at any given moment. For this reason, CFDP is built entirely on a store and forward communication model. If file transmission between Earth and Mars Entity is interrupted, CFDP entities at both ends simply store their Protocol data unit (PDUs) in their non-volatile memory until the spacecraft reemerges and transmission can resume to assure reliable file transfer. An indirect benefit of this model is that it largely prevents application from state of communication environment: Mars surface entities can record terrine observation in a

file and transmit without considering whether or not physical transmission is currently possible to Earth.

CFDP supports both "un-acknowledgement" and "acknowledgement" modes. Since powerful forward error correction coding can minimize data loss in deep space communication but cannot eliminate completely, CFDP support acknowledged modes of operation in which data loss is automatically detected and retransmission of the lost data is automatically requested. However, the long propagation delays and high error rate limit the application of the retransmission strategies commonly used in terrestrial protocols. For this reason, CFDP's retransmission model is one of concurrent transmission

CFDP is designed for re-use without modification in any number of communication environments as well. No specific link-layer protocol is mandated in the Recommendation. Instead, an abstract underlying "unit data transfer" or UT layer service is assumed to be available for CFDP's use. The specific requirements imposed on the UT layer are minimal, enabling CFDP to be run on top of such widely varying services as UDP/IP on the Internet. Each CFDP PDU is simply encapsulated in a single UT "service data unit" (e.g., packet or datagram) for transmission, and the details of conveying the PDUs from one CFDP entity to another are left to the UT service. This design function was helpful in defining simulation protocol topology, where in our topology acknowledgement mode reliable CFDP is simulated over UDP/IP.

### 5.2.2 Protocol Operation

In the CFDP, the file transfer is called a "transaction," and the sender assigns a transaction ID for each file-transfer operation. The transaction ID, along with the source ID and other information, is contained in the header of each Protocol Data Unit (PDU). Table 5.1 defines the abbreviations used for various PDU types; the protocol definition in reference [25] defines the meaning and format of these PDUs in detail. The sender informs the receiver of the start of the file transfer by transmitting the meta-data PDU, which contains information such as the source and destination IDs, the file name, the file size, etc. Like most PDUs in the CFDP, there is no ACK for the meta-data PDU, and the sender is allowed to transmit file-data PDUs (PDUs carrying the actual content of the file) after transmitting the meta-data PDU. In other words, there is no handshaking for initiating a "transaction." The receiver detects the failure in delivering a file-data PDU or the meta-data PDU by noticing missing elements in the sequence of PDUs correctly received. Each file-data PDU has a field that specifies the starting byte

Table 5.1: Abbreviations used for various PDU types

Abbreviations	PDU Type
M	metadata
FD(n)	file data segment
NAK	retransmission request
EOF	end of file (sender to receiver)
FIN	finished (receiver to sender)
ACK	acknowledgment
PRMPT	prompt

number and ending byte number of the file data carried by the PDU, so the receiver can detect missing PDUs by observing the ending byte numbers and the starting byte numbers of the correctly received PDUs. If the meta-data PDU is lost in the first trial, the receiver will detect that the meta-data PDU is missing, because the new transaction ID in the header of that received PDU will indicate that the new transaction has begun. The receiver reacts to the missing PDU by sending NAK messages. Each NAK message contains the list of PDUs requested by the receiver for retransmission. Upon receiving a NAK, the sender retransmits the PDUs requested. When the sender runs out of the file-data PDUs to send, the sender sends an EOF PDU, thus initiating the closure of the file transfer.

After receiving the EOF PDU, the receiver acknowledges it with an ACK (EOF) and waits until the meta-data PDU and all of the file-data PDUs are received before it initiates the closure of the transaction. All data are eventually received because of the NAK mechanisms, and the receiver can notice the reception of all data from the file-size information contained in the meta-data PDU and the EOF PDU. Then, the receiver sends a FIN PDU. After receiving the FIN PDU, the sender acknowledges it with an ACK (FIN) and closes the transaction. When the ACK (FIN) is successfully delivered back to the receiver, the receiver also closes the transaction, at which point, the transaction is closed at both entities. For EOF and FIN PDUs, there are ACKs and retransmission timer mechanisms, so their exchange is reliable. According to the CFDP, the receiver and sender must both transmit an ACK message in response to each EOF/FIN PDU, even after closing the transaction, in order to prevent possible anomalies in closing the transaction.

The quality of service offered by the protocol is selectable, according to mission requirements and transmission capability, and ranges from an unacknowledged option, whereby a file is transmitted with no attempt at completeness should errors occur (errors will be detected and data discarded), to a fully acknowledged option providing error

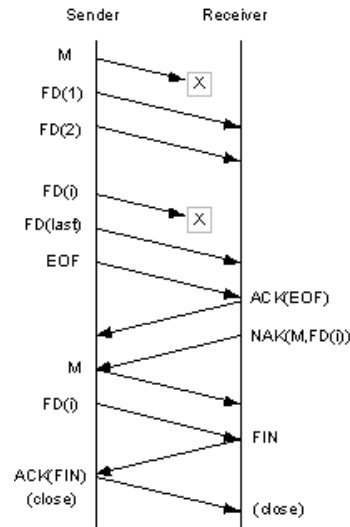


Figure 5.1: Protocol Transmission Sequence for Deferred NAK Mode [Image Courtesy [25]]

recovery through retransmission. The acknowledged sub-options share a common acknowledgment mechanism but use different strategies in making retransmission requests to optimize for different scenarios. Depending upon mission requirements and transmission capability four user selectable options associated with the issuance of NAKs: Deferred, Immediate, Prompted, and Asynchronous NAK Mode.

- In the Immediate NAK mode [17], each discontinuity in the data detected at the receiving entity results in the immediate transmission of a NAK to the sending entity. The Immediate NAK mode is useful, for example, where the communicating entities are tightly coupled; it makes no attempt to control the number of NAK messages it uses, in return for maximizing completeness of the received portion of a file as the transfer progresses.
- In the Prompted NAK mode, the sending entity transmits a Prompt (NAK) message to the receiving entity telling it to send its NAK. When the receiving entity receives the Prompt (NAK), it sends any outstanding NAK. In response to a Prompt (NAK) when no data is missing, a CFDP NAK may be empty (that is, request the retransmission of no data). The EOF is treated as an inherent prompt and results in the receiving entity's sending a NAK if any data is missing.
- In the Asynchronous NAK mode the receiving entity issues a NAK (if any data is missing) in response to some outside event; that is, the receiving entity is triggered by something outside of the CFDP to send any necessary NAK. Such an external

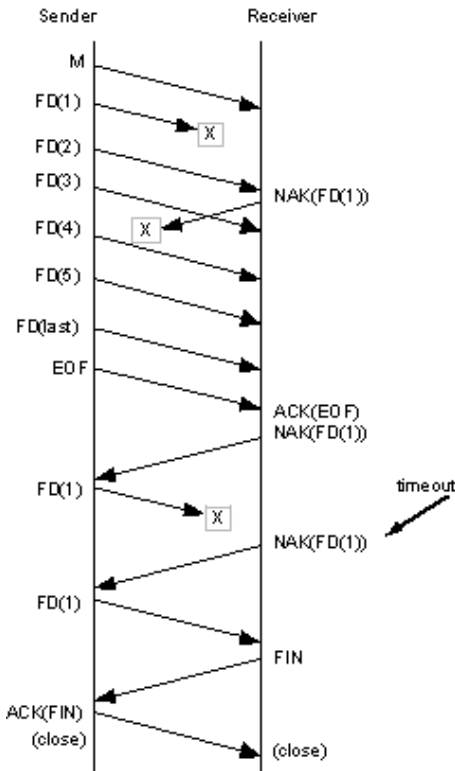


Figure 5.2: Time-out Triggered NAK Retransmission [Image Courtesy [25]]

event might for instance be the impending loss of the space-to-ground link.

- In the Deferred NAK mode, the receiving entity saves all information about missing data until the EOF is received. It then issues a NAK to request the missing data. An example is shown in Figure 5.1. The deferred NAK mode may be appropriate where communicating entities are very loosely coupled, such as when interplanetary distances introduce very long light time delays.

### Timer

Several timers are used in the reliable service processes. In each case in which a time-out capability is required, a timer is started upon issuance of the item. Upon receipt of the required response, the timer is disabled. If the required response is not received before the timer expires, the item is reissued. A count of the number of retransmissions is kept. If the preset limit of retransmissions is exceeded, a fault is declared. In reliable service, within the file copying process timers are invoked for the EOF, the EOF-triggered NAK, and finished (FIN) transmissions. The operation of the NAK time-out is illustrated in Figure 5.2. A NAK timer is started upon the issuance of the

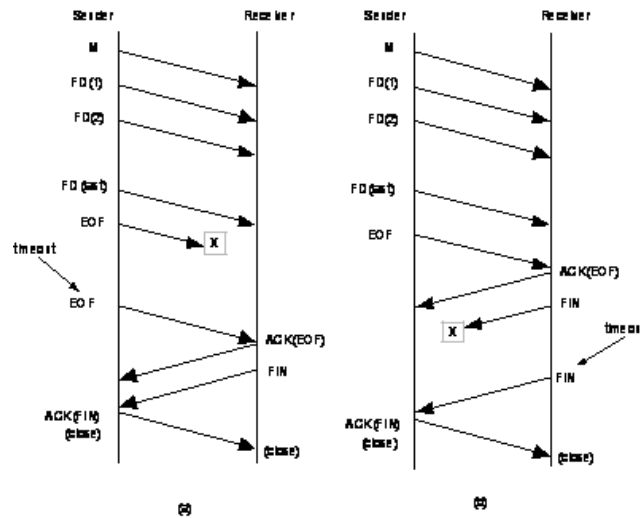


Figure 5.3: Time-out Triggered EOF and Finished Retransmissions [Image Courtesy [25]]

EOF-triggered NAK (which requests (re)transmission of all file data not yet received). Note that previous individual NAKs are not acknowledged. When the timer expires, the receiving entity again determines whether or not any of the transactions file data or metadata has yet to be received. If any file data gaps or missing metadata remain, normally a NAK is issued and the timer is reset. The operation of the end-of-file and finished time-outs is shown in Figure 5.3, parts (a) and (b), respectively. Both timers are implementation specific it is not required to be the same on successive iterations of this data completeness determination cycle; in fact, it might even be dynamically altered (possibly multiple times) prior to any single expiration.

### 5.2.3 Deferred NAK Mode

In this work, main focus is given to deferred NAK mode since deferred NAK mode suite best in interplanetary Mars network scenario, where are characterized by higher intermitted connectivity with variable high propagation delay.

In the deferred NAK mode, the receiver defers issuance of NAKs until it correctly receives the EOF PDU from the sender. The receiver keeps the record of all missing PDUs until the EOF PDU is successfully delivered. After receiving the EOF PDU, the receiver issues an ACK (EOF) and issues a NAK that requests retransmission of all missing PDUs, if any. Upon receiving a NAK, the sender immediately retransmits all PDUs that the NAK requests. At the end of each transmission of a NAK, the receiver sets a NAK timer, and when the NAK timer expires, the receiver again examines

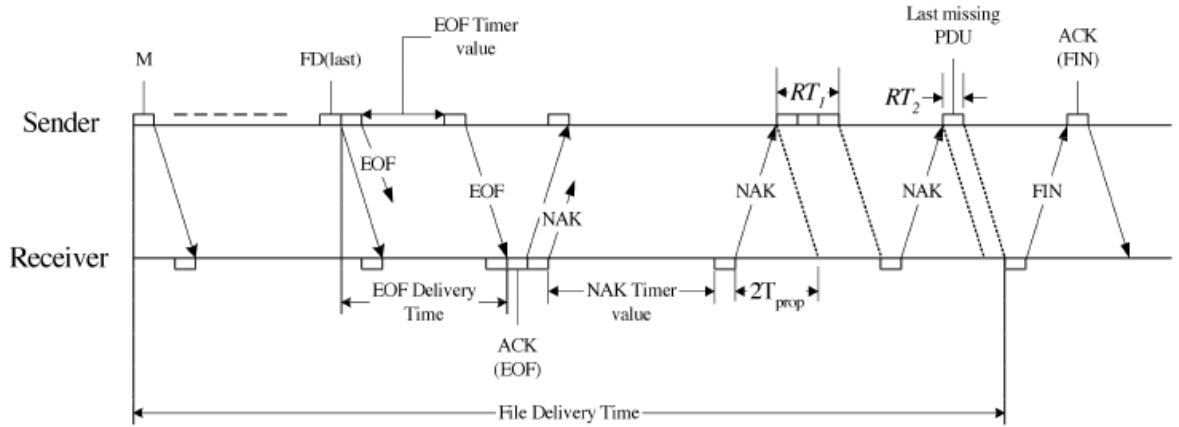


Figure 5.4: Deferred NAK mode: Where  $T_{prop}$  stands for one-way propagation delay, and  $RT_k$  stands for the duration of the  $k^{th}$  retransmission spurt. ("Transmission spurt" refers to consecutive transmission of PDUs back to back.). [Image Courtesy [53]]

the record of missing PDUs. If missing PDUs still remain, the receiver issues another NAK and again starts a NAK timer. This process continues until the receiver receives all necessary PDUs that contain the whole file content and the meta-data PDU. After receiving all necessary PDUs, the receiver issues a FIN PDU, and upon receiving the FIN PDU, the sender issues an ACK (FIN) and closes the transaction. The delivery of the FIN PDU is guaranteed in the same way as the EOF PDU. The receiver closes the transaction when the ACK (FIN) is successfully delivered to the receiver. Figure 5.4 shows the operation of the deferred NAK mode. The performance metric file delivery time (defined as time measured from beginning of the transmission till the first instant when all file data, meta data and EOF PDU have been successfully received by the receiver), mainly depended on both EOF and NAK timer value, hence by tuning the timer value according to network condition performance can be improved.

### 5.3 Related Works

Previous work on performance analysis of CFDP includes Daniel C. Lee and Wonseok Baek [53], Wang et al [97][98][99] and Tomaso de Cola et al [31]. The minimum expected file-delivery time of the CFDP deferred NAK mode is derived in [53]. The paper also discusses the ARQ timer-setting rule that minimizes the expected file-delivery time under the constraint that the throughput efficiency is maximized. With regard to performance measures, this paper is mainly concerned with the time taken to transfer



a file (file-delivery time) and throughput efficiency.

Compare CFDP/TCP with the commonly used TCP and space communication protocol standards (SCPS) protocol stacks to see which one is more effective over a LEO-satellite link and over a GEO-satellite link are presented in [97][98]. The experimental results show that CFDP does not have performance advantage over other protocols over a less lossy channel with a BER less than or equal to  $10^6$ . However, along with the increase of BER, CFDP shows significant performance advantage. The experimental results also show that CFDP/TCP in the deferred NAK mode is much more effective than other two stacks in the simulated point-to-point space communication links, especially when operated with a long link delay, a high BER and channel-rate asymmetry. It is found that CFDP/TCP is less sensitive, and FTP/TCP and SCPS-FP/SCPS-TP are highly sensitive, to the increase of link delay.

In [31] author propose a CFDP extension, based on the implementation of erasure coding schemes, within the CFDP itself, in order to assure high reliability to the data communication even in presence of very critical conditions, such as hard shadowing, deep-fading periods and intermittent links. Different encoding techniques are considered and various channel conditions, in terms of Bit Error Ratio and bandwidth values, are tested.

## 5.4 Dynamic Timer Algorithm

In this section, we illustrate the proposed dynamic timer algorithm for deferred Mode CFDP to improve the delivery latency by minimizing the file delivery time. We have defined set of rules to tune the timeout value for both sender and receiver according to the environment condition. In acknowledged transmission mode, CFDP entity has two timer namely EOF and NAK timer to ensure the retransmission of lost EOF and PDUs. As total file delivery time is given by the sum of time taken until the reception of EOF ACK and time taken for reception of all lost PDU by the receiver, both timer value have significant impact in the delivery latency.

### 5.4.1 Preliminaries

The main aim of this algorithm to set timeout value dynamically to reduce unnecessary retransmission of PDUs due to early or late timeout thereby, minimizing the file delivery time. For simplicity, we make the following assumption.

- Meta data PDU and all file data PDUs are assumed to have identical lengths,

Table 5.2: Notation used

Symbol	Description
$T_{prop}$	One-way propagation delay
$T_{ack(eof)}$	Transmission time of ack of end of file PDU
$T_{PDU}$	Transmission time of meta data or file data PDU
$RT_k$	retransmission time of meta data or file data PDUs for $k^{th}$ retransmission windows

identical transmission times, and identical probabilities of failed delivery (PDU error or loss). The Meta data PDU is usually shorter than a data PDU and thus has lower probability of failed delivery. However, the length of the Meta data is so small in comparison with the length of the file data that the assumption of equal lengths has relatively a negligible effect on the total file delivery time.

- All NAKs have identical lengths and identical probabilities of failed delivery. (The length of a NAK depends on the number of PDUs that it requests. However, the differences are small, and the lengths of NAKs are all small, so this assumption should not significantly affect the performance measure.)
- Probability of loss of EOF PDUs are assumed to be low, as the uplink bandwidth is very high when compared with downlink bandwidth, and EOF PDUs is much shorter, compared with file data PDU.
- we also assume negligible processing time.

For the convenience of mathematical derivation, we have defined  $T_{eof}$  and  $T_{nak}$  as the timer interval set for both EOF and NAK Timer as per the Protocol specification [25], which gives freedom on this value to implementers. Note the Table 5.2 for the notations used in the analysis Due to the effect of the aforementioned large bandwidth-propagation delay product, we ignore the relatively insignificant transmission time of individual control messages and individual PDUs and only count latency in terms of propagation delay

#### 5.4.2 EOF and NAK Timer Algorithm

In this section we address the dynamic timer algorithm. As can be deduced from the protocol description in the introduction (illustrated in Figure 5.4), the ex-

pected file delivery time depends upon certain parameter values that any implementer can freely choose, for example, the time-out value of the EOF timer and the time-out value of the NAK timer. In general the timeout value for both EOF and NAK timer are set according to the recommendations of the CFDP implementation as two times propagation delay ( $2T_{prop}$ ) plus the transmission time of ACK(EOF) in case of EOF timer and retransmission time of require PDUs for the  $k^{th}$  retransmission windows in case of NAK timer, which are defined in the following equations.

$$T_{timeout}^{eof} = 2 * T_{prop} + T_{ack(eof)} \quad (5.1)$$

$$T_{nak}^k = 2 * T_{prop} + RT_k \quad (5.2)$$

### Static EOF Timer

In CFDP deferred Mode, sending entity sends a metadata PDUs followed by one or more data PDUs and an EOF PDU. After sending EOF PDU sending entity starts an EOF timer and waits for EOF (ACK) from the receiving entity. Since we assumed that EOF PDU have very low loss probability and size of EOF (ACK) PDU is small, we proposed a static timeout value of EOF Timer set according to the equation 7.3. EOF timer mechanism is simple like any protocol scheduler mechanism with following rules:

1. When EOF PDU is send, sending entity start the timer with the timeout value set according to the equation 7.3
2. when timer value expires, sending entity re-sent the EOF PDU and re-starts the timer with same value.
3. when the sending entity receives EOF ACK PDU from the receiving entity, it disables the EOF timer and starts the re-transmission phase if any.

### NAK Timer Algorithm

In this section, we have elaborated the dynamic NAK timer algorithm, which set the NAK timeout value of the receiving entity. We propose that a timer be set upon every generation of a NAK. This algorithm use counter parameter introduced in CFDP implementation [25], where counter is defined as expiration limit of the NAK timer. Counters are implementation specify. In the algorithm the counter threshold value is fixed using simulation in way that any increase in counter value beyond the threshold has

negligible impact in the performance. To make the timer dynamic  $\beta$  factor is introduced in to the general equation 7.4 as:

$$T_{nak} = \beta * T_{prop} + RT_k \quad (5.3)$$

Due to the geometry of the nodes in the space and their orientation, we have very long and varying propagation delay, which affect the setting of timeout value of NAK timer. In this algorithm, ' $\beta$ ' value is use to dynamically control the timeout value setting by varying propagation delay. In the general implementation of the CFDP, when a counter reaches its threshold value it's invoke user specific fault procedure in retransmission phase. Expiration of counter is attained when receiver do not receive any further PDU from the sender even after some defined limit of counter value. This situation may occur either due to intermitted or no connection during that passage of time or drop of PDU due to bandwidth congestion. We have given the notation  $\mu$  for the counter threshold value. We have used a fixed counter value instead dynamic since we have observed from the simulation, that increase in counter value beyond certain threshold does not have impact in file delivery time.

#### NAK Timer rule

When receiving entity receive a EOF PDU, it send a EOF(ACK) for the requested EOF PDU and check for any gap in the received packet sequence, if it finds any, it send NAK PDU requesting all missing data PDUs and enable the NAK timer and initialize NAK counter.

1. At the generation of first NAK, timeout value of NAK Timer  $T_{NAK}^1$  is set according the equation 7.5 with the  $\beta$  value set as 2, meaning two time the propagation delay plus transmission time of the requested PDUs (equation 7.4).
2. At the generation of subsequent NAKs before the expire of the counter, the following rules are applied:
  - If there is no active timer (timeout value = 0), then a new NAK timeout is again set as two time propagation delay plus the transmission time of the requested PDUs
  - If there is an active timer and receiving entity receive an EOF PDU (if EOF PDU is lost), then receive compare the remaining timeout value with two time propagation delay, if remaining time is greater, then receiving entity start

the new NAK timer and set the value to be remaining timeout value plus the transmission time of the requested PDUs, else if the remaining timeout value is less the two times propagation delay, it start a new timer with value set according to equation 7.4 and stop the old timer.

3. At the generation of subsequent NAKs, after expire of the counter value (due to intermitted connection or multiplexed transactions), the following rules are applied:
  - When the counter reach the threshold ( $\mu$ ) with succession transmission of NAK without any reception in PDU from the sender; then receiver change the ' $\beta$ ' value according to the following equation 7.6. And start, the new NAK timer by setting the timeout value according to the equation 7.5 with newly determined  $\beta_{new}$  value. And re-initialize the counter.

$$\beta_{new} = \beta_{old} + \frac{\mu}{\beta} \quad (5.4)$$

In real operations, the sender may not be able to start retransmission of the PDUs requested by the NAK. For example, if the sender is performing multiple outgoing transactions concurrently (multiplexed transactions), the sender may have to delay the requested retransmissions in a particular transaction, because of previously queued outbound data belonging to another transaction that must be transmitted before the newly requested PDUs are retransmitted. For this reason, algorithm vary the propagation delay by a fraction of  $\beta$  when the counter expire, so that receiver wait bit more than usual before sending any unnecessary NAK PDUs.

4. When receive data PDUs from the sender, receiver do the following:
  - When receiver receives the requested missing data PDUs, from the sender, it compare (equation 7.7) the remaining timeout value with the general timeout value (2 times propagation delay plus time required to for requested received PUDs),

$$T_{rem}^k \geq T_{nak} \quad (5.5)$$

if the remaining timeout value is greater than compared timeout value; the receiver reset back the timeout value to general timeout value assuming the network connection is good as before. If the remaining timeout value is less than equal to general value it uses the same existing timeout setting.

The proposed NAK timer setting, which relies on counter and  $\beta$  value, almost remove the ambiguity in how to determine the time-out value of the NAK timer. As previously explained, NAK timer is set upon the first successful reception of EOF. However, an improper setting of the NAK timer value can cause unnecessary duplicate retransmissions. Without proper consideration of potential retransmission activities at the sender, it is hard to determine the smallest time-out value of the NAK timer that prevents any unnecessary duplicate retransmission. The proposed setting rule use "counter" mechanism so that the receiver, upon expire of counter, may have a approximate estimation of sender's potential retransmission activities (like: multiplexed transactions or network state). And timer algorithm use varying propagation delay to counter early timeout thereby reduces unnecessary retransmission of NAK PDUs in bottleneck return links and data PDUs in the forward link. The following sections show the improvement obtained in term of File delivery time using the proposed algorithm with the help of simulation.

## 5.5 Implementation and Validation

As the first step before implementing the proposed dynamic timer algorithm, we have implemented the CFDP protocol itself in Network simulator (NS2) [4]. NS2 is an open source simulated with well documented simulation structure and we have good knowledge in NS2, for these reasons we have preferred NS2 over other simulators. CFDP is an application layer protocol which may or may not use underling transport protocol according to its implementation. Since the structure NS2 abide by the full protocol stack simulation (i.e. Simulation script should provide detail of all protocol from application to physical layer), we have implemented reliable CFDP to work over unreliable UDP protocol, where reliability mechanism is implemented in CFDP itself. CFDP protocol is implemented using C++ language, where application layer CFDP protocol class inherits NS2 application class, thus CFDP sits in application layer of the NS2 architecture. All simulation scenarios are developed using TCL script language as per the specification of NS2 architecture.

### 5.5.1 Validation

As the article "Do Not Trust All Simulation Studies of Telecommunication Networks" [63] suggest that creditability of the simulator is an important factor in any performance evaluation studies and one of the solutions to this crisis is to use a valid

simulation model. To this extend we have validated our simulation model. To validate our simulation model we have used two validation technique; animation validation and comparison to other models. We have used Network Animator (NAM) [4] a companion animator of NS2 to verify that the setup behaves as intended regarding basic operation as shown in Figure 5.4. For the second validation technique, we have uses analytical model developed in [53] and compared simulation file delivery time achieved over the given scenario with the expected file delivery time calculated using analytical model.

### 5.5.2 Analytical Model and its assumptions

In our validation phase, we have used the analytical model developed by Daniel et al [53], where the mathematical expression derived in for the expected file-delivery time in deferred NAK mode discuss how to compute it numerically. For computation, author considered a the single-hop file-transfer operation for a file size of 1MB, with the transmission rate of 20kb/s in both direction and propagation delay of 1 astronomical unit (note that 1 astronomical unit is equal to 480 sec). The expected file-delivery time depends upon several variables, e.g., file size, PDU size, the propagation delay, etc. Numerical value for expected file-delivery time is generated by varying number of PDUs. Probability of PDU error in the forward link ( $P_{ef}$ ) is fixed to be  $10^{-3}$ . And it is assumed that probability of error in EOF PDU and probability of error in NAK PDU in the return link are two orders of magnitude less than  $P_{ef}$  because of the small sizes of the NAK, EOF PDUs. Here we validate of implemented CCSDS CFDP protocol code without the proposed dynamic timer algorithm for its functioning. To validate our simulation model, we used same topology as the analytical model. We have plotted file delivery time vs. Number of PDUs.

#### Animation Validation

From the NAM graphical output, we can see the basic behavior of the protocol, where EOF packet trigger receiver sends NAK packets, thereby starting the retransmission mechanism. Since we have uniform error model for simulation, we can see uniform drop in PDUs. Thus we can validate the intended behavior CFDP protocol.

#### Model validation using Comparison

We have plotted result of simulation for file delivery time over number of PDUs sent, which is shown in the figure 7.10. From the graph in the figure 7.10 we can see

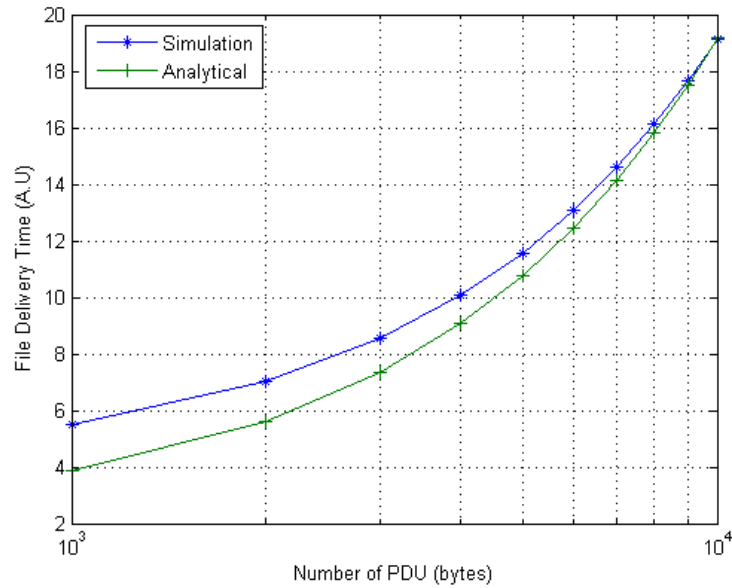


Figure 5.5: Validation of Simulation Model

that, there is a linear increase in file delivery time when more number of bytes is transfer. And result of simulation model seems almost fit with that of analytical model, for the reason we can argue that code and functioning of the simulation model is valid. Though the analytical model is valid with the simulation model, we tend to use the simulation model rather than analytical model, since it difficult to express the expected file delivery time topology other than single-hop and introducing dynamic timeout value for NAK time is simple in the case of simulation model. Thus after validation of CFDP protocol with the static timer, the proposed dynamic timer algorithm is implemented into the protocol.

### Implementation Protocol Behavior

To understand the implementation behavior of CFDP, we have simulated the same scenario with simple uniform error model, where we have calculated the file delivery time for various bit error rate (BER). The PDU error rate parameter  $P_{er}$  of CFDP is derived from BER using the expression  $P_{er} = 1 - (1 - BER)^N$ , where N is the number of bytes. File delivery time is plotted for three BER value  $10^{-3}$ ,  $10^{-3.5}$ ,  $10^{-4}$  (hard intermitted environmental conditions) The resulting graph is shown in Figure 5.6. As expected the file delivery time increase with increase in BER due to the retransmission of lost PDUs to complete a file transmission.



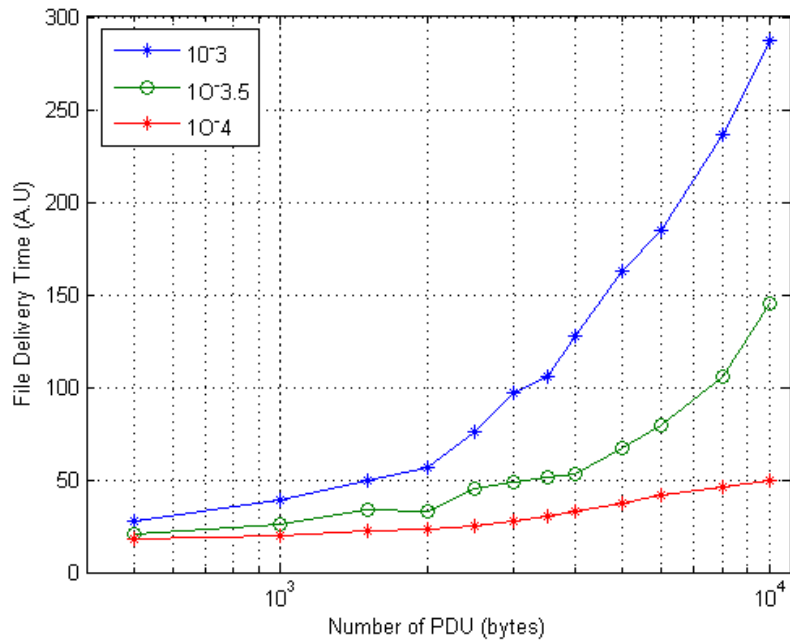


Figure 5.6: File delivery Time over various BER in single hope scenario

### 5.5.3 Counter Threshold

The timeout setting rules proposed for NAK timer is depended on counter threshold value  $\mu$ . In order to understand impact of the counter and find out optimal threshold value, the same point-to-point simulation scenario is used, which have been used in validation phase. The counter threshold value is defined for BER =  $10^{-3}$  (worst environment condition - high error rate) assuming it would be effective for lower error rate as well. The simulation is conducted with different sets of counter threshold value like 3;5;7; and result is presented in Figure 5.7. The counter threshold value is considered from 3 because below 3 (2or1), timeout setting rules almost behave like general static timer mechanism. And it is seen from the graph in Figure 5.7, that after certain threshold value counter does not have any effect on file delivery time. And it is also seen that when  $\mu = 5$  file delivery time is comparatively less than when  $\mu = 3$  and also after which counter does not have any impact; so the threshold value for the counter is fixed 5 ( $\mu = 5$ ), when BER =  $10^{-3}$ .

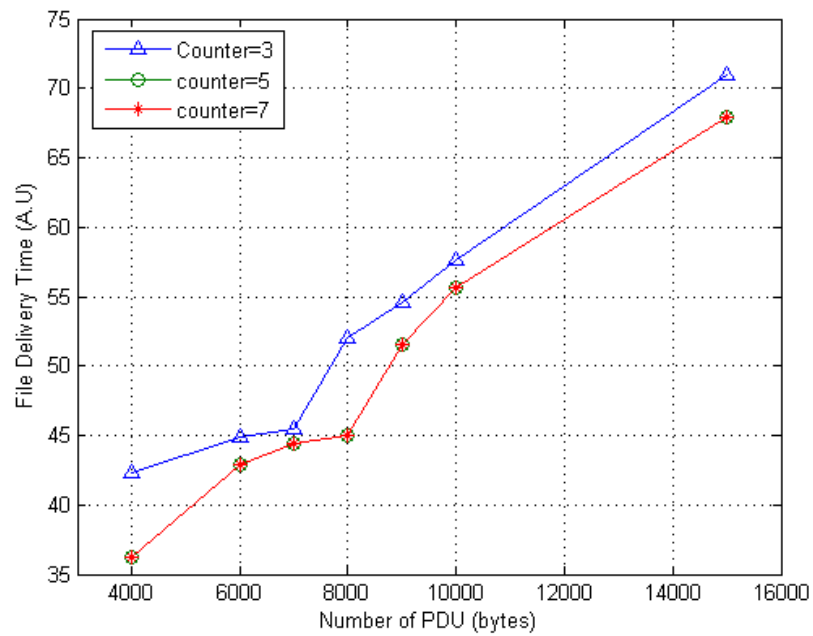


Figure 5.7: Counter Threshold

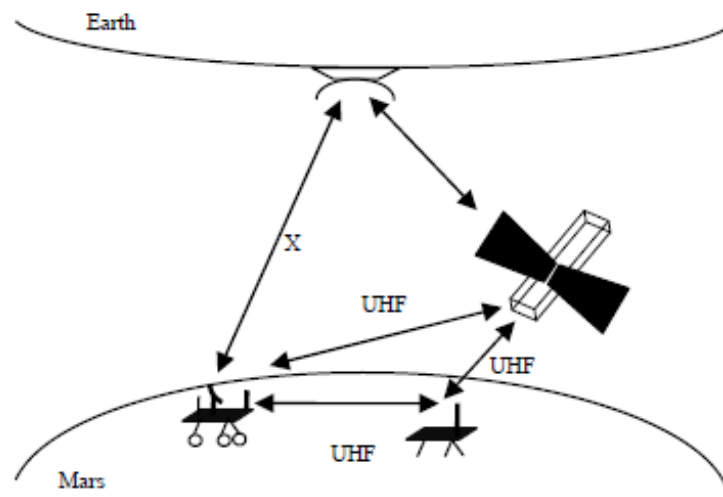


Figure 5.8: Simulation Topology

Table 5.3: Simulation Parameters

Parameter	Orbiter-Rover Link	Orbiter Ground station Link
Link Delay	12ms	1AU ( 1.AU = 480s)
Data Rate	1024Kbps	258Kbps
Bit error rate		
Clear Sky		$10^8$ to $10^7$
Hard link - intermittence		$10^6$ to $10^4$
PDU Size		1KB
File Size		1MB
Link Availability		75%
Transport Protocol		UDP

Table 5.4: BER and the corresponding PDU error rate.

BER	PDU error rate
$10^{-3}$ (bad weather)	0.9997
$10^{-4}$ (bad weather)	0.5507
$10^{-5}$ (good weather)	0.0769
$10^{-6}$ (good weather)	0.008
$10^{-7}$ (good weather)	$7.99968 \times 10^{-4}$

## 5.6 Performance Analysis

In this section, we present the simulation results for dynamic timer algorithm under various conditions. In order to evaluate the performance improved of the dynamic timer algorithm over simple static timer rules. The simulation is done with each BER and link delay for both symmetric and asymmetric channels.

### 5.6.1 Simulation Topology

For all simulation, we have used a simple interplanetary network topology as shown in the Figure 5.8, where we have a Rover node, Lander Node. Both are connected to orbital node which is connected to Ground station node. In the simulation, we used a simple point-to-point link between nodes and space like parameter are emulated using a link model which assign appropriate link delay and error model for each link. And also the orbit passes are emulated using Up/Down functions of NS2. The simulation parameters are shown in Table 6.6 for the links. BER and corresponding PER are listed in the Table 5.4.

Here, we attempt to have comparative evaluation of CFDP performance with and without our dynamic timer algorithm. For each combination of the above parameters, we repeat the simulation with 10 different seeds and we present their average outcome with 95% confidence interval where the confidence interval is calculate using the following equation

$$CI = \bar{X} \pm \alpha * \frac{\sigma}{\sqrt{N}} \quad (5.6)$$

Where,  $\bar{X}$  is mean of the sample,  $N$  is sample size,  $\sigma$  is the standard deviation of the sample and  $\alpha$  is interval co-efficient which is chosen to be  $\alpha = 1.96$  for 95% confidence interval using normal distribution.

### 5.6.2 Metric

In our performance analysis, we have considered average file delivery time as main performance metric. Since the timer algorithm were proposed to reduce delivery latency by minimizing the retransmission of packets under bandwidth constrain. Another metric which we have considered in our analysis is throughput efficiency which is given by the following formula

$$Throughput = (1 - P_{loss}) * \frac{TransferSize}{TransferTime} * \frac{1}{Bandwidth}$$

where  $P_{loss}$  is loss probability

### 5.6.3 Performance on symmetric channel

Figure 5.9 shows the average file transfer time of 1Mbyte file of the CFDP protocol in comparison with and without dynamic timer algorithm, over symmetrical channel (20 kb/s: 20 kb/s) with respect to the BERs and used 75% up time for orbital links. The proposed dynamic timer algorithm (DTA) do not show any signification performance improvement when BER is low and show only slight improvement when BER increase. Since there is no bottleneck in return link due to bandwidth (symmetric), and the link up time is relatively high, the proposed algorithm does not have much impact on performance. The proposed algorithm is less sensitive in symmetric channel.

### 5.6.4 Performance on asymmetric channel

Figure 5.10 shows the average file transfer time of 1Mbyte file of the CFDP protocol in comparison with and without dynamic timer algorithm, over asymmetrical

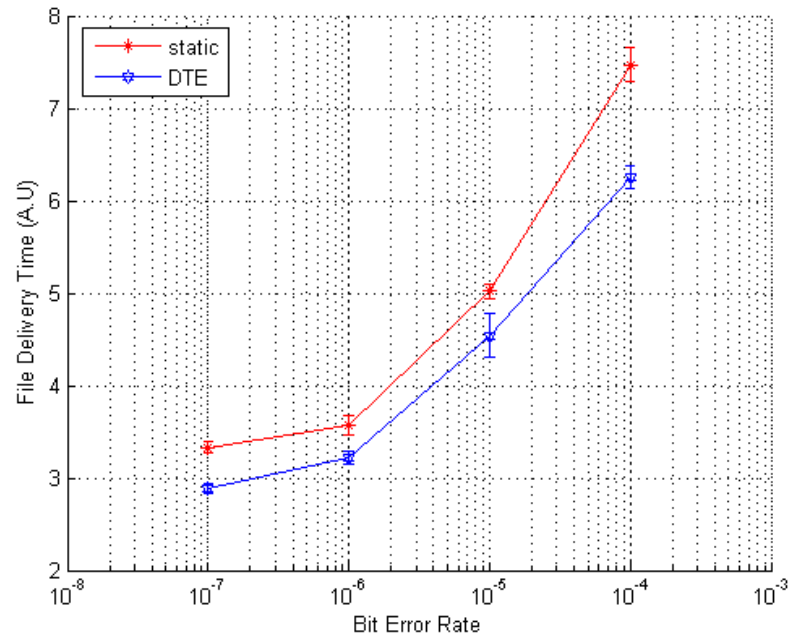


Figure 5.9: File Delivery Time over symmetric channel

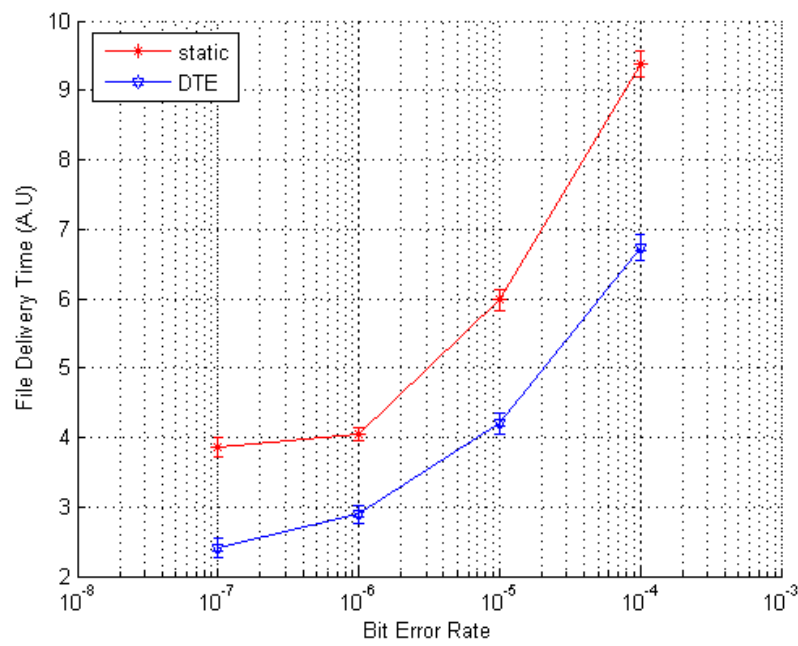


Figure 5.10: File Delivery Time over asymmetric channel

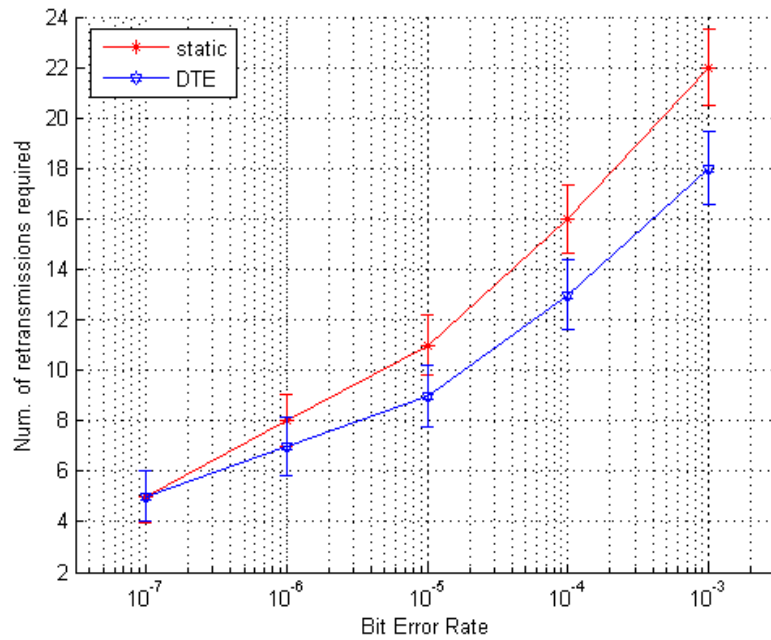


Figure 5.11: Number of Retransmission required to complete 90% of file transfer for 10Mb file

channel (128 Kb/s: 20 Kb/s) with respect to the BERs and 75% uptime for orbital links. From the graph, it is observed that, there is a significant improvement in file delivery time with proposed DTA, due to the reason that DTA reduce number of retransmission in both forward and return link with its dynamic timeout values, where by reducing unnecessary time for transmission of additional redundant packet and improving file delivery time.

Considering that the protocols show significant performance difference almost at all BER, let us directly compare their average number of retransmission and throughput.

### 5.6.5 Number of Retransmission Required

In the following experiments the number of retransmission required to complete the 90% of file transfers for 10Mb file is evaluated by varying the BER of the channel. Intuitively, we can expect as the channel noise level increase, the number of retransmission required also increase and it is shown in the Figure 5.11. This simulation is conducted with 60% uplink duration. When the BER is quite low, Number of retransmission required for both static and dynamic timer are almost same, since probability of loss is quite less with low BER and link break. As the BER changes from good weather

condition to worst, we can see a significant difference in retransmission requirement between static and dynamic. The dynamic timer algorithm, varies the required timeout value to counter link break there by minimize the possible retransmission due to early or late time out, for this reason we have few retransmission requirement in the case dynamic timer algorithm when compared with the static timeout value.

### 5.6.6 Throughput performance in Intermitted connectivity

In order to understand impact of our algorithm over intermitted connectivity, we have considered a asymmetric network with BER equals to clear sky scenario for 1MB file transfer and calculated the effective throughput by varying the block size (number of PDUs) for two different link availability period: 50% of the simulation time and 30% of the simulation time. Here PDU size are changed according to the number of PDU required to complete 1MB file size (for example if Number of PDU transferred is 50 then PDU size is set a 20KB), and result is show in Figure 5.12. The throughput performance of Reliable-CFDP/UDP is totally unaffected or minimally affected by the channel BER, propagation delay and link-rate asymmetry because of the UDP protocol's unreliable transmission mechanism. In comparison, the transmission behavior of the UDP-based CFDP connection is quite different in the presence of intermittent link connectivity. For this reason, throughput performance over link connectivity is analyzed.

It can be infer from the graph that the throughput performance of the proposed dynamic time algorithm is more optimal than static timer algorithm, one explanation is being, that in UDP based CFDP, the UDP receiver will stop receiving the delivered packets when a link break is introduced in the data channel since they have no knowledge about the network which reduce throughput. In the case of proposed algorithm where, the link break is almost countered by the increase in timeout value (varying propagation delay), UDP receiver don't not break the connection and if broken, it is reestablishes without any time spent on interactions between two communicating entities. Thus good throughput performance is achieved even in low link connectivity scenario. A too large PDU size (Low number of PDUs) penalizes the transfer because the time required for the retransmission is too long. For this reason we see a low throughput performance, when the number of PDU is less (meaning big PDU size). The best performance is obtained for PDU size is small. This behavior is more evident for low link availability period (30%), the drawbacks introduced by PDU big size are outstanding.

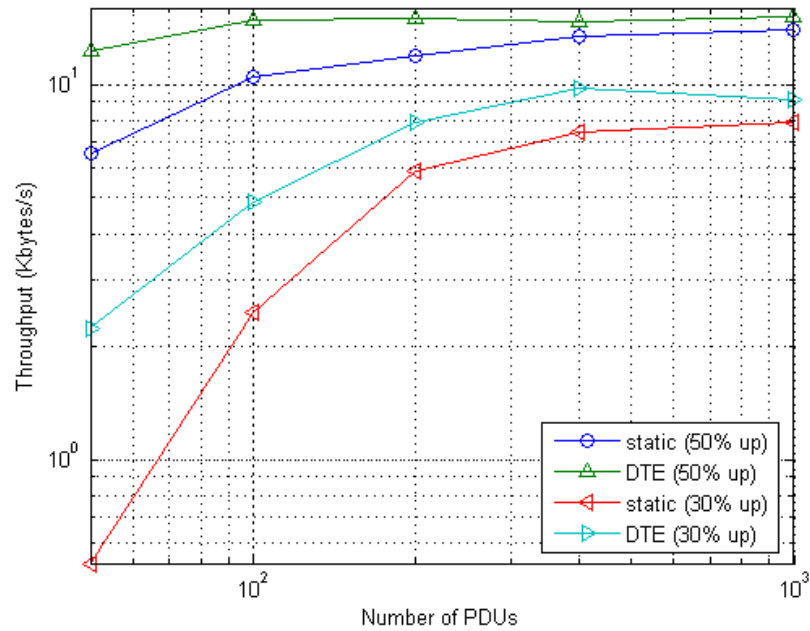


Figure 5.12: Throughput Performance over Number of PDUs for 50% & 30% Link availability period

### 5.6.7 Number of File

In order to demonstrate the improvement in the file transfer capacity of the proposed algorithm, we done carried out this simulation experiment, where the sender PUDs to destination until the end of simulation time duration (1500s) with 60% uplink duration with channel error rate of  $10^{-4}$ . And number of completed file (1MB) at the destination is calculated for various the PDU size. Figure 5.13 shows the simulation result of the experiment. As expected the dynamic timer algorithm can able complete transfer of more file compared with static timer implementation. As the number of retransmission required to complete the file increases with increase in channel error rate, static timer file completion percentage is comparative less then dynamic timer algorithm. As size of PDU increase, increases the retransmission time, hence static timer algorithm perform poorly when packet size is too high. On the other hand, unnecessary retransmission is reduce with the introduction of dynamic timeout setup, thus we can see a more number of files are transfer for given time duration in case of DTE. Thus we can conclude that the proposed dynamic timer algorithm is effective in highly intermitted asymmetric link.



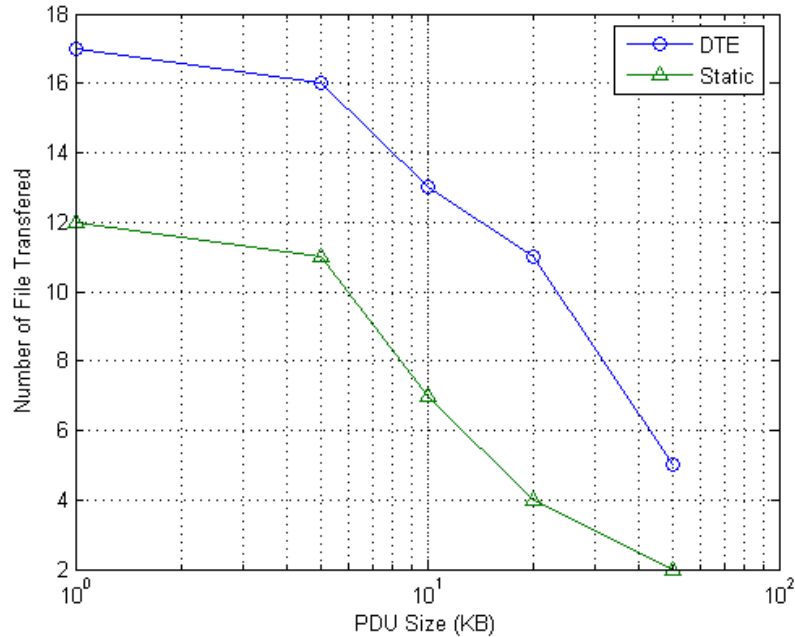


Figure 5.13: Number of File completely transfer where File Size = 1MB

### 5.6.8 Impact of Buffer Size

Intuitively, we expect that a larger file size will create a very high buffer requirement because a single PDU error will require Orbiter (MTO) to hold the entire file in storage. Through the buffer storage delay is higher in the case of proposed dynamic timer algorithm, where the node stays more time in retransmission phase when channel error is high (due to increase in timeout value). In the case of static time algorithm, node storage is utilization is proportional to number of retransmission requirements, for this reason buffer requirement for both static and dynamic timer algorithm have negligible difference. Since there is no application buffer size definition in NS2, we are unable to provide exclusive sibilating results on buffer size requirement. And changing the link buffer in simulation dose not have any effect in the performance.

### 5.6.9 Would Immediate NAK Mode Make a Difference?

The natural question, given what we learn about Deferred NAK, is whether Immediate NAK will make a significant difference. The answer is no. The latency advantage of Immediate NAK is only significant when one is dealing with a low bandwidth-propagation delay product link, such as the UHF relay link between a Mars craft and a relay orbiter. Immediate NAK mode allows a receiver to generate retransmission re-

quests before receiving the EOF message. However, upon reception of an EOF, the protocol will work identically to Deferred NAK mode.

In timer equation, we actually ignored the file transmission time since it is small compared to the propagation delay. This means that the assumption derived for Deferred NAK is equally applicable to Immediate NAK over a long delay link, since the transmission time, compared to the overall latency, is really in the noise. Therefore it is clear that Immediate NAK will not provide any noticeable latency reduction over the deep space link; it is a special mode best applied for short-range communications.

## 5.7 Discussion

In this work, we have studied the most commonly use deep space application protocol CCSDS File delivery protocols for its functioning and limitation over Interplanetary network. We have analyzed various research work, conduction to evaluated the performance of CFDP over LEO, GEO and interplanetary network and concluded that timer implementation in CFDP have great impact in performance of the protocol, and defined the need for new dynamic timer algorithm.

We have then, selected CFDP deferred NAK mode of transmission as subject to study, since it suits best for varying delay interplanetary network. Then we proposed a new dynamic NAK time algorithm, which uses implementation specific "counter" parameter to set its timer value dynamically according to the network conditions. We have also implemented CFDP protocol in NS2 with our dynamic NAK timer algorithm to evaluate its performance over default static timer implementing. We show in the performance result, the proposed dynamic timer algorithm achieves good performance in term of delivery ratio and delay over asymmetric intermitted connections.

As CFDP standards, allows users to implement their own timer and counter mechanism (user specific), our approach seems to provide a good timeout definition which achieves good delivery performance over the interplanetary networks.

## Chapter 6

# QoS based Protocol Selection in Interplanetary Network

This chapter speaks about QoS framework for interplanetary network. Framework helps the interplanetary applications to choose optimum underlying protocols, suitably measured in terms of ability to achieve the application QoS requirements with the environmental constraints.

### 6.1 Motivation

Recent developments in the design of deep space mission have given rise to numerous interplanetary applications, other than simple data transfer from distant nodes. And the main focus of researchers is to provide communication infrastructure in a challenging environment. To that extent, CCSDS and DTNRG have developed protocol stacks which perform optimally at the given environmental conditions. They do, however, possess some limitations. In addition, neither of the CCSDS or DTN protocol stacks contains autonomy or flexibility. Interplanetary network applications are designed in such a way that they can work under any lower layer protocols. To use effectively the protocol resources, CCSDS deployed protocol selection based on application layer knowledge. They proposed three transport protocols to respond to varying Quality of Services. While the idea of multiple protocols in attempting to satisfy QoS levels is considered effective, the protocols deployed are not.

For this reason, we have intended here to combine both CCSDS and DTN protocols to provide more options in protocol selection to satisfy application QoS. And also make the selection process dynamic by taking into account the environmental

condition during selection. To this extend, we have proposed a QoS framework which act as middleware reside between application and transport protocols, where the application layer select its transport protocol which has ability to achieve its QoS requirements.

## 6.2 QoS mapping in Interplanetary Network

Interplanetary networks applications are designed in such a way that it can work under any lower layer protocols seamlessly. In general each protocol layer is considered as black box with specific input and output data. QoS mapping perform as a translator of data coming from different layers. There are two sorts of mapping, vertical and horizontal used for transforming information between layers and system components. In our approach we have considered vertical QoS mapping, since mapping modules main job is to translate the application QoS requirements into underlying protocols QoS metric, by which application selects optimum protocol suitably measured in terms of ability to achieve the applications QoS requirement with the environmental constrains.

QoS Mapping rules are the basic elements in application mapping module. A rule transforms one or set of source QoS information into on or set of target QoS information coming or going into same or different layers. In general QoS mapping rules are classified in two categories namely:

1. Function based mapping rule: are basically mathematical formula and often created using interpolation methods
2. Table based mapping rule: are characterized by a lookup table, often defined by users or developers, but which can also be built using experimental tests. A lookup table contains a limited number of entries by which a set of output values can be obtained from the input values.

Both function based and table based mapping rules have their own limitations. Since building and validation of mapping function is usually big challenge in function based approach which make mapping rules limited. The limitation of table-based mapping comes from the application of such rules: while function-based rules can be used for a wide range of applications or environments (i.e. packet rate calculation can be used in all the TCP/IP applications), table-based rules are only valid within specific working conditions. However, in our approach we have just used the abstract principal of table based mapping rule for the given application traffic, where application QoS metric are

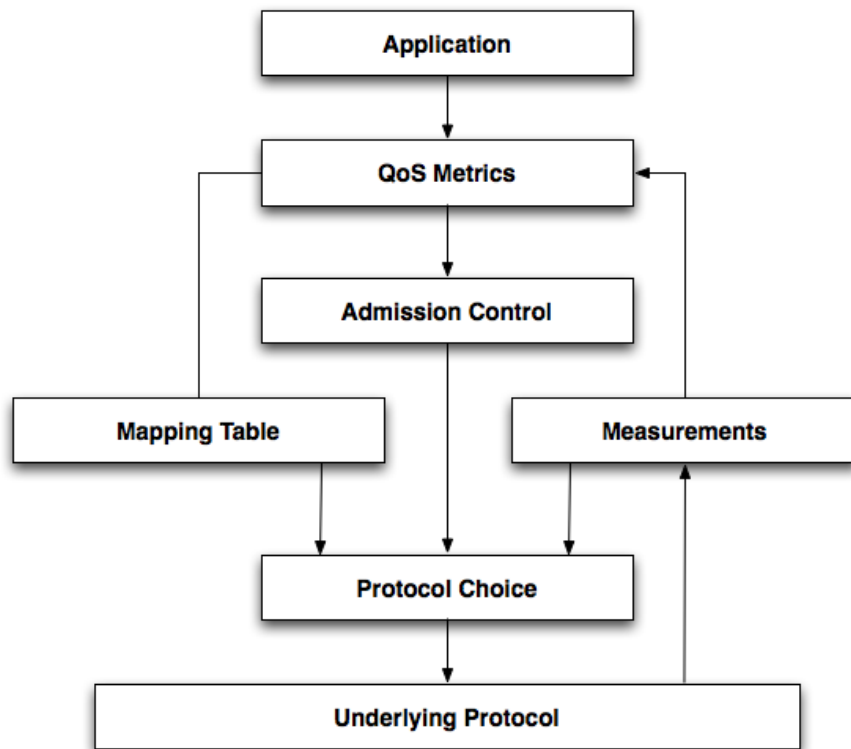


Figure 6.1: QoS Framework to select transport protocol

synthesized and a protocol chose are made from the protocol pool according to mapping rules from the table. In our approach we have considered only two layers of the protocol stack, the application and transport layer. Further depth in the protocol layers are considered as vision for future work.

### 6.3 QoS framework to select protocol stack for communication

To provide communication autonomy to the Mars surface nodes, we have proposed a QoS framework, as depicted in Figure 7.12. QoS framework is integrated between application layer and transport layer. Working in a top-down approach, where the protocol choice are made according to combined information of application QoS metric with dynamic environment measurement data. Choosing the underling protocol in real time, maximizes the opportunity that required QoS is achieved under given network conditions. The proposed framework, with the combination of QoS metric and measurement modules is able to construct the Mapping rule to select the underlying protocol

Table 6.1: QoS Metric for different interplanetary application

Application	QoS Metric	Description
Telemetry	Delay ( $D_{max}$ ) and Throughput ( $B_{min}$ )	This is the primary application which report health and status of spacecraft. An important aspect of telemetry systems is its delivery characteristics, which are either "stream-oriented" or periodically delivered.
Scientific data	Best Effort	Scientific data delivery is one of the prime application interplanetary network, which are in general not particularly time sensitive, have high tolerance of poor timing estimation and error coping abilities.
Multimedia	Delay ( $D_{max}$ ) and Throughput (constant)	This type of application is required to deliver great volumes of audio and visual information about the local environment to Earth. They are time sensitive applications, which require high data volume.
Command & Control	Delay ( $D_{max}$ ) and Low loss rate	Command and control refers to the closed-loop control of remote systems. These types of applications have low tolerance of poor timing estimation and error coping abilities.

for the transmission of data with the promising QoS requirements. Upon reception of a transmission request from the application layer, the framework use the application layer QoS metric and dynamic measured environmental information to perform the admission control (whether to start the transmission) and then with the table based mapping rule select the underlying protocol for the communication. The framework processes through four states during the communication, which include measurement, admission control, evaluation of mapping rule and protocol selection or abort.

### 6.3.1 Application Layer QoS Metrics

Application Layer QoS Metric are specified in terms of maximum acceptable delay ( $D_{max}$ ), minimum throughput ( $B_{min}$ ), maximum acceptable bit error rate ( $E_{max}$ ), data volume ( $V_{min}$ ) and best effort (without any QoS). These application layer QoS represent the type of traffic (class of service) injected by the source application. Typical interplanetary application is characterized according to the QoS metric depicted in the Table 6.1. The application layer QoS metric are populated once at the beginning of a communication. It contains information about the application performance requirements (maximum acceptable delay, minimum allowable throughput).

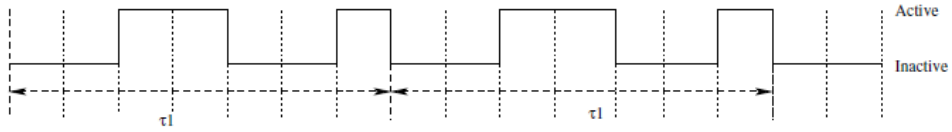


Figure 6.2: Periodicity of Link

Table 6.2: BER for different atmospheric conditions

Atmospheric conditions	BER value
Almost clear sky	$10^{-8} - 10^{-7}$
Hard link intermittence	$10^{-6} - 10^{-4}$
Deep fading periods	$10^{-3} - 10^{-2}$

### 6.3.2 Measurement Module

In the framework, measurement module process three measurements, propagation delay, periodicity of links and link error rate. The propagation delay and link availability period comes from ephemeris. Astronomical data regarding the motion of planets and satellites has been recorded and tabulated as ephemeris tables by NASA [8]. The ephemeris tables of [8] provide two major distance measurements of planets. The first, the geocentric distance, is the distance of the planet being observed from the earth, in AU. The other is its distance from the sun in AU ( $1 \text{ AU} = 1.496 \times 10^8 \text{ km}$ , the average distance between the sun and the Earth). With the help of this distance measurement we obtain propagation delay. A link in interplanetary network can be considered as t-graph switches between active and inactive states over time. Since the movement of planets nodes is periodic, the link shows periodicity. Consider a link changing state as shown in Figure 6.2. It is observed that the state of the link repeats itself with a period of  $\tau_1$ , which is obtained from ephemeris table. Link error rate changes depending upon the atmospheric conditions like clear sky; rain; wind; etc. A predefined error rate are associated for each atmospheric conditions which is furnished in the table 6.2.

### 6.3.3 Admission Control

Admission control is an important part in any QoS framework, especially for dynamic network like interplanetary network. Admission control rejects new flow when sufficient resources are not available. Admission control works upon environmental measurements. In our proposed framework, we have two set of admission rules, one for delay sensitive application class and another for delay insensitive application class.

1. When delay sensitive application traffic is requested for transmission by the application layer, admission control module compare propagation delay with maximum acceptable latency. If propagation delay is greater than maximum acceptable latency which is unable to overcome then admission control abort the flow.

Pseudo code:

```
If (Propagation Delay > Maximum Allowable Latency )
    Unable to overcome the transmission delay: abort the flow;
Else
    Allow the flow;
```

2. When delay insensitive traffic is requested for transmission by the application layer, admission control module check buffer capacity of the node and allow the traffic only if available buffer in the node is greater than transmission data volume.

Pseudo code:

```
If (Available Buffer > Transmission Volume)
    Allow the flow;
Else
    will create buffer congestion: abort the flow;
```

#### 6.3.4 Mapping Table

In the proposed framework, mapping rules are table based, that is predefined list of protocol are accumulated in protocol pool before the beginning of transmission like Application layer QoS Metric. A mapping rule is an association of application requirement and protocols offer. Protocols are mapped with their abilities in providing required QoS in the given environment. This mapping is defined based on QoS metric, measured information and protocols ability. To evaluate ability of the underlying protocol in predefine protocol pool, we have analyzed the performance ability of transport protocols which are proposed for interplanetary network based on specific performance criteria.

#### Transport Layer QoS Approach

Proposals from the CCSDS and DTNRG are successful standalone protocol stacks. They do, however, possess limitations. Transmission Control Protocol (TCP) is the most widely used reliable, in-sequence transport mechanism for the network. It



is a stream-oriented transport and features end-to-end flow control and congestion control. On the other hand User Datagram Protocol (UDP) provides a low-complexity, best-effort transport. It is often used in cases where reliability and sequenced delivery are not required. UDP is also unaffected by one-way or asymmetric links. The CCSDS SCPS-Transport Protocol (SCPS-TP) is designed to interoperate with TCP/User Datagram Protocol (UDP). It combines both TCP- and UDP-like services: reliable-stream, partially reliable-stream, and best-effort. Another TCP based interplanetary transport protocol is TP-Planet. Its approach to connection establishment is effective; it sends the initial data transmission with the connection request, maximizing throughput while ensuring reliability. TP-Planet is limited where bandwidth is limited.

Delay-Tolerant Networking (DTN) bundling protocol provides an overlay that allows communication across different networks via a store-and-forward mechanism. It is designed to operate in an environment where a complete end-to-end path does not exist due to sparse topology or frequent and unpredictable link disruptions. While ensuring reliability, bundling is ineffective when network resources are limited. Store-and-forward techniques are suitable only when transmissions are of lengths suitable for the buffers. The store-and-forward approach is also unsuitable for multimedia applications. Licklider Transport Protocol (LTP-T) is another DTN based transport protocol. LTP-T enables flow and congestion control on hop-by-hop basis to provide reliability. As it operate with same store and forward principal like bundle protocol, it is delay insensitive.

### **Rules**

In the proposed framework, many to one mapping pattern is used, that is many Application QoS metric to map to one transport protocol in the protocol pool. For example, if a telemetry application is considered for transmission. This application has strict boundaries on transmission latencies. In addition, it has a low tolerance of data loss, then mapping rule select reliable connection oriented protocol like TP-Planet for transmission. Mapping table of interplanetary applications with their QoS requirements to their corresponding choice of the protocols are shown in the Table 7.1.

#### **6.3.5 Protocol Choice**

Protocol choice is final stage in the proposed framework, where application layer selects the most appropriate underlying protocol for the given application class under current environmental conditions. When admission control module allow the

Table 6.3: Mapping Table of Interplanetary applications

Application	QoS Metric	Protocol
Telemetry	Delay ( $D_{\max}$ ) Throughput ( $B_{\min}$ )	TP-Planet / LTP-T
Scientific Data	Best Effort	Bundle Protocol / SCPS-TP
Multimedia	Delay ( $D_{\max}$ ) Throughput (constant)	RCP-Planet
Command and control	Delay ( $D_{\max}$ ) High delivery Rate	LTP-T / SCPS-TP (TCP based)

traffic flow, the protocol choice module pick the transport protocol for the corresponding application class from the mapping table and evaluate the selected protocol with the current environmental state before validating the choice. In the framework, it has been taken into account some possible network condition of the system in order to make optimal decision. We have defined set of selection rules for all four application class for possible network conditions, where the protocol is selected according to their ability and its (protocol) parameters are reconfigure according to the environmental condition before transmission of data.

#### Telemetry Application Class

As telemetry application are time sensitive and require good reliability it is mapped to TCP based space protocol like TP-Planet. Before selecting the mapped protocol for transmission, it is evaluated with the environmental information. TP-Planet relies on conversational traffic, hence they implicitly expect continues bidirectional connectivity between source and sink and also its performance is below optimal in error conducive environments. As its is know that overhead incurred by TP-Planet is high when the file size small, since significant portion of the connection period is spent in the Initial State, where the number of probing segments transmitted is high compared to that of data packets. As the file size increases, the overhead decreases. This is because the overhead in the Steady State is due to the transmission of small sized (40 bytes) probing segments and hence is much lower compared to Initial State. Therefore, as the file size increases, the time spent in Steady State also increases, which in turn decreases the overall overhead in the connection period. With respect to BER TP-Planet performance is very optimal at high error rate. On the other hand goodput of LTP performance decrease with increase in file size and also provide good performance even in high error rate, thanks to its store and forward technique. Taking this limitation

into account, before selecting the protocol for transmission, module evaluates both BER and file size before selecting the protocol.

Pseudo code:

```

if (application class == telemetry) {
    TP-Planet == true;
    LTP == true;
}
if (BER < appMaxBER) && (File Size > 10MB)
    Protocol =TP-Planet;
else
    Protocol = LTP-T;

```

If both BER is less than acceptable application requirement and file size is more than 10MB then module choice TP-Plant for transmission. . If either one of the condition is failed, then module choice next available protocol (LTP-T) from the table for transmission.

### **Scientific Data Application Class**

In time insensitive traffic like scientific data transfer, admission control module block the flow only when source buffer space is full or almost full. On the other hand, protocol choice module, make the selection decision from the mapping table between bundle protocol and SCPS-TP. Bundle protocol provide reliability service using custody transfer mechanism, where node which take the custody of the incoming bundle (packet) store the bundle is buffer until it get a positive acknowledgement or until expire time of the bundle itself. For this reason, nodes with higher buffer storage are considered for custody transfer to avoid storage congestion. Considering buffer factor in to account, protocol module selects protocol according to buffer availability at neighboring node in the path. On the other hand, overcome various space channel problems and provide reliable data transport in space communications. And SCPS-TP performance mostly limited in high error link and not much affected by buffer size. For this reason, protocol module evaluates buffer size to select the suitable protocol.

Pseudo code:

```

if (application class == Scientific Data) {
    BP == true; SCPS-TP == true;
    calculate(buffer availability);
}
If (buffer availability > data volume)
    Protocol = BP;
else
    Protocol = SCPS-TP;

```

### Command and Control Application Class

Bundle protocol (BP) and LTP-T provide good reliable data transmission in interplanetary network. Though both protocol used the notion of custody transfer for reliable service and cope with data loss in high error environment. Accepting of custody of all receiving block leads to storage exhaustion, which also lead to storage congestion. Though there is far, little to choose between LTP and the BP in terms of performance, standard deviation of goodput for larger file size is less compared with Bundle protocol, here LTP-P is selected in first place for this application in the mapping table instead of BP. So to avoid buffer congestion buffer space availability of the neighboring nodes are check before selection of LTP-T for transmission.

Pseudo code:

```
if (application class == Command and Control) {
    if (reliability service == true){
        LTP == true;
        SCPS-TP=true; }
    If (buffer availability > data volume)
        Protocol = LTP-T
    else
        Protocol = SCPS-TP(TCP based)
}
```

### Multimedia Application Class

In case of multimedia applications (audio or video), research shows that RCP-Planet [36] provide reasonable performance, hence the application is mapped with this protocol.

#### 6.3.6 Autonomic Behavior

As defined by IBM, a system must possess four properties to be truly autonomic [49]: properties includes, self-configuration, self-optimization, self-healing, self-protection. According to the definition, the proposed framework is not fully autonomic. It self-configure in selecting transport protocol (and can be extended to other underlying layers) to match application QoS over environmental constrains. Measurement allows self-optimization, by using past responses and information to set the environmental information for mapping when the same scenario re-occurs. where else self-healing and self-protection is not included in the framework which make it not fully autonomic.

Table 6.4: Configuration Parameters

Application Parameters	Environmental Parameters
Transmission volume (bytes)	One way propagation delay (sec)
Maximum Bit error rate (bits)	Possible next node
Maximum latency (sec)	Storage availability (packets)
Transfer waiting time (sec)	Error rate (pks)

## 6.4 Implementation and Analysis

We have implemented the proposed QoS framework in Network simulator (NS2). We have used "TCL" script language to develop this QoS framework, which act a middleware between application layer and transport layer. Thus the developed script read a predefine configuration file of the nodes, evaluate the environmental parameters and write a protocol scenario for the simulation. The configuration file contains the application parameter (Table 6.4) like transmission volume, maximum allowable BER and maximum allowable latency for the given application. Initially we have implemented a semi-autonomous protocol selection. That is environmental parameters of each node are predefined before the start of the transmission. For example; buffer availability value for each node is predefined in the configuration file and while selecting protocol for scientific data and command and control application, it uses this value for evaluation of the chosen protocol.

Simple TCL procedures have been developed for each module in QoS framework. Initially for simplicity, we have assumed that the variation in propagation delay is very less, hence neglected and measurement module is predefined in a configuration file before the simulation, thus more static than dynamic. When an application request is received by the framework procedure, mapping module used the appConf file to map the corresponding protocols using the mapping table, and send this information to admission control module, which uses source envConf file to determine, whether to admit or abort the transmission flow, if the result is admit, it send the collated information to protocol choice module, where protocol choice module use the envConf files of next possible nodes and construct protocol scenario for the simulation and finally simulation module ; simulate the given constructed scenario for performance evaluation.

### 6.4.1 Demonstration

To demonstrate the working of implemented framework, we have considered a simple scenario of three nodes (Rover, orbiter, Earth Station), where rover wanted to

send telemetry data to ground station. The Application QoS requirement is defined in appConf file shown below.

```
appConf File:
Transmission Volume = 100MB
File Size = 10MB
Maximum Latency = 0.2 sec
Maximum BER = 10E-4

rover envConf File:
Buffer Size = 1.2MB
Possible Next Hope = orbiter
Propagation delay = 0.06 seconds
Clear Sky (BER = 10E-6)
```

When the application layer sends a transmission request for the telemetry application class, the admission control module of the framework evaluate the environmental constrain to decide whether to allow the transmission flow or not. Since telemetry class is a time sensitive application class, it compare propagation delay with maximum allowable latency, in this case, propagation delay is less the allowable latency and control module allow the flow and pass the information to protocol choice module.

```
Pseudo code:
Dmax = 0.2 (appConf)
Propagation Delay = 0.06 (envConf)
While ((Dmax > Propagation Delay) || (0.2 > 0.06))
    Flow == true;
```

Protocol choice module maps the suitable transport protocol using mapping table information, namely; TP-Planet and LTP. And it evaluated the selected protocol using the environmental information before selecting for transmission. Here, it compares the BER which application can allow with environmental BER and select TP-Planet as transport protocol.

```
Pseudo code:
Max BER = 10E-4 (appConf)
BER = 10E-6 (clear sky : envConf)
if (application == telemetry ){
TP-Planet = true;
LTP-T = true; }
If (TP-Planet == true || LTP-T == true) {
    if (Max BER > BER ) && (File size > 10MB>)
        Protocol = TP-Planet
    else
        Protocol = LTP-T
}
return(Protocol)
```

When QoS framework returns a value for transport protocol, TCL middleware construct the scenario script for simulation.

```

instproc create-traffic {id src dst start Protocol} {
    $self instvar ns_
    set dSACK = 15
    set pro_($id) [new Agent/Protocol]
    set sink_($id) [new Agent/ProtocolSink]
    $ns_ attach-agent $src $pro_($id)
    $ns_ attach-agent $dst $sink_($id)
    $ns_ connect $pro_($id) $sink_($id)
    set app_($id) [new Application/telemetry]
    $app_($id) attach-agent $pro_($id)
    $ns_ at $start "$app_($id) start"
}

```

Thus, TCL middleware (including QoS framework) write the simulation scenario after selecting the optimum transport protocol, suitably measured in term of ability to achieve the application QoS requirements with the environmental constrain.

## 6.5 Performance Analysis

The performance of framework is measured in term of its processing delay in select a transport protocol in the given scenario. Since a series of evaluation asses protocol suitability, given the QoS requirements of the application and operation capabilities of the protocol, there is countable amount of additional processing delay introduced in the system. In order to investigate the performance of QoS framework, we conducted extensive simulation experiments. Telemetry protocol choice and their performance with and without framework parameter re-configuration is evaluation in section 6.5.1. Effect of Buffer size and packet size over the throughput in scientific data application protocols is evaluated in section 6.5.2. Effect of re-configuration in the protocol setting for throughput performance over the multimedia application protocol is investigated in section 6.5.3. The processing delay incurred by the framework is evaluated in section 6.5.4.

### 6.5.1 Telemetry Application Scenario

Throughput efficacy is one of important Application QoS required by the telemetry traffic. In this section, we evaluate the throughput efficacy of both the protocol mapped for telemetry application. Both in TP-Planet and LTP, there are few

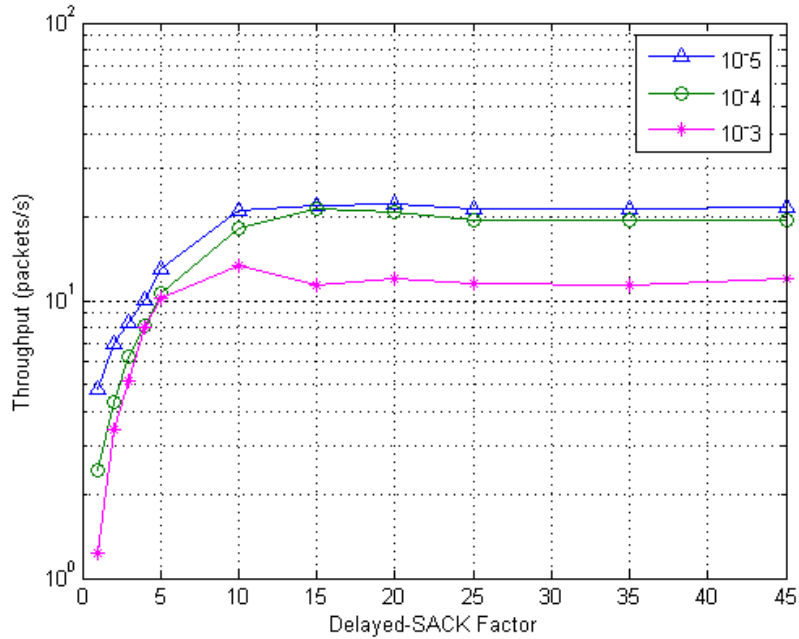


Figure 6.3: Throughput for changing delayed SACK factor with different PER and RTT=120s

protocol parameters which affect the performance of the protocol. In case of TP-Planet the parameters like sliding windows size, delayed SACK factor influence the performance of the protocol. In case of LTP-T, checkpoint ratio, number of red part influences the performance. For this reason, Protocol choice module in the framework tunes these parameters to achieve good performance.

### Throughput Performance - TP-Planet

As the interplanetary networks are characterized by extremely high propagation delays, link errors, asymmetrical bandwidth, TP-Planet address these challenges by deploying SACK congestion control by delaying the SACKs with certain delayed SACK factor  $d$ . TCP based transport protocols performances are affected largely by bandwidth asymmetry. This simulation experiment is performance to show the effect of bandwidth asymmetry on the performance and the improvement achieved by delayed SACK. For this experiment we used a simple interplanetary network topology as shown in the Figure 5.8. Here RTT is set as 120s ( $RTT_{min}$  in very long range), 3 set of packet error rate is used  $p \in [10^{-3}, 10^{-4}, 10^{-5}]$  and the simulation time is set to be 1200s. Data rate is set as 50KB/s. We have calculated the throughput by varying the delayed SACK factor for an asymmetric link of ratio 1000:1 (100 and 0.1 KB/s) for all three set of



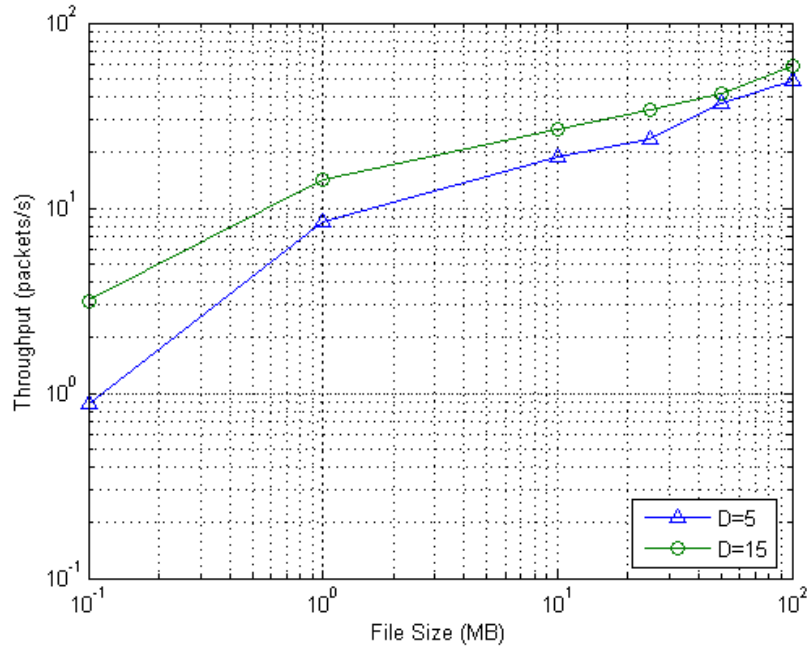


Figure 6.4: Throughput for changing file size with delayed SACK factor  $d = 5$  and  $15$  with framework re-configuration

packet error rate.

Figure 6.3 show the simulation results. As shown in Figure, an increase in the delayed SACK factor also leads to an increase in the throughput. However, throughput decreases for all cases with either too high or too small delayed SACK factor. This is because if it is too small, reverse channel is congested. On the other hand, if it is too large, the source receives less number of SACKs than it expects which leads to performance degradation. From the graph we can infer that, we can achieve good throughput at specific delay factor; like when  $d=15$  for a packet error rate of  $10^{-4}$ . Thus protocol choice module, when TP-Planet is chosen as transport protocol for transmission it set the delay factor according the BER defined with help of simulation.

In the following experiment, we have evaluated the throughput performance of the protocol with and without framework setting according to the environmental condition. In the experiment we varied the file size with fixed delay factor ( $d=5$  default setting of TP-Planet) and dynamically reconfigured delay factor by the framework according to condition. Figure 6.4, shows the performance improvement archived when the delay factor is set dynamically over fixed scenario. Thus we can argue that framework select protocol to their ability to provide required QoS.

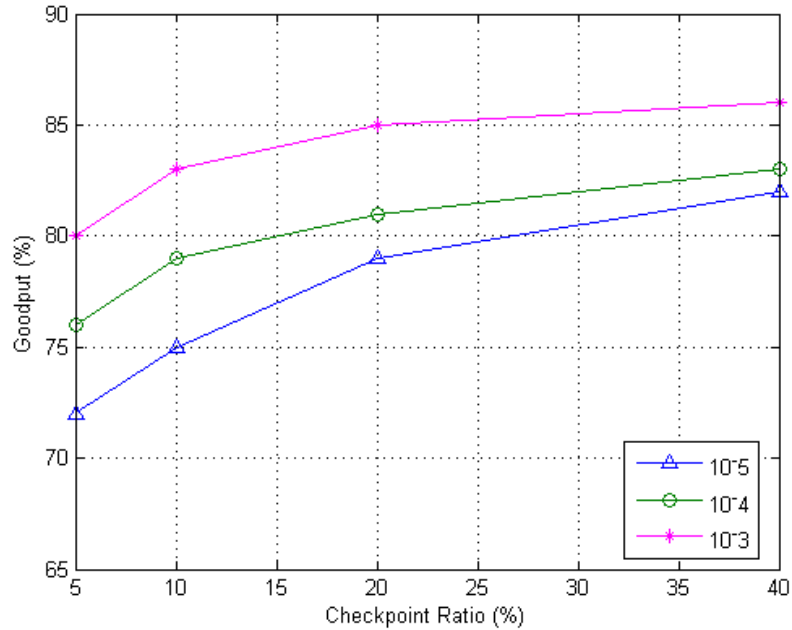


Figure 6.5: Goodput percentage as a function of checkpoint ratio for different loss rate

### Goodput Performance - LTP-T

LTP-T basically behaves like a sequence of independent LTP session. LTP operation is divided mainly into red part (reliable transmission) and green part (unreliable transmission) It starts with red part segment transmission and one may include any number (application dependent) of checkpoints. A special red part segment, the EORP (End Of Red Part), must be a checkpoint. On the reception of a checkpoint, report segments, containing report claims, are sent to the transmitter, which are acknowledged by report acknowledgment segments. At the reception of an EOB (End of Block), the entire data is delivered to application layer. The checkpoint ratio (percentage of number of red block) have deep impact the goodput performance of the protocol since goodput is defined as the ratio between the number of red segments to the application has to send and the number of red segments actually transmitted.

In order to evaluate the optimal checkpoint ratio, we have done simulation experiment by varying the checkpoint ratio and calculated the goodput for different packet error rate so that protocol choice module set the optimal checkpoint ratio dynamically. From the Figure 6.5, we can see that after certain checkpoint ratio (CP=20 %) goodput is almost constant for all packet error scenario, there is impact of checkpoint ratio on the goodput. The protocol choice module adapts this checkpoint ratio threshold for each

Table 6.5: Goodput and End to End Delay for Error rate of  $10^{-3}$ 

	CR=20 (framework)	CP=10 (default)
Goodput %	73.36	80.79
End-to-End delay	0.63 AU	0.84 AU

packet error rate. So when LTP-T is selected as the transport protocol choice module set the checkpoint ratio according to BER of given scenario, by this way it improve the performance ability of the chosen protocol. In the following experiment we have shown the performance improvement achieved using QoS framework with dynamically configured checkpoint ratio according to the packet error rate. Results shown in table 6.5.

### 6.5.2 Scientific Data Application Scenario

In scientific data application traffic, Bundle protocol is predominantly bundle protocol is chosen as transport protocol. Orbital nodes and earth ground station nodes generally have high buffer space compared to Mars surface nodes (Rover/Lander), for this reason bundle protocol is selected mostly. Another protocol in the pool for this traffic is SCPS-TP, where SCPS-TP is developed with a number of extensions and modifications to TCP to cope up with interplanetary challenges. Asymmetry channels are predominate challenge in interplanetary network, where packet size affect the throughput performance of TCP based transport protocol. For this reason, in the following experiment we have intended to find the optimal packet size for SCPS-TP over varying BER channels. To this extend, we have conducted simulation experiments for a simple 3 node topology (rover, orbiter, earth station) with different packet size at BER  $10^{-4}$ ,  $10^{-5}$ ,  $10^{-6}$  over asymmetric channel. Experimental parameter is shown in the table 6.6. From the figure 6.6 we can see, when bit error rate is less ( $10^{-6}$ ), the minimum

Table 6.6: Simulation Parameters

Experimental factors	Settings/values
Experimental file size (byte)	1 000 000
RTT (ms)	250
Channel rates	115 200 bit/s : 2400 bit/s (ASYM)
BERs	$10^{-4}$ , $10^{-5}$ , $10^{-6}$
Packet size (byte)	250, 500, 750, 1000, 1250, 1500, 1750, 2000
Send buffer size (byte)	65 536
Receive buffer size (byte)	65 536

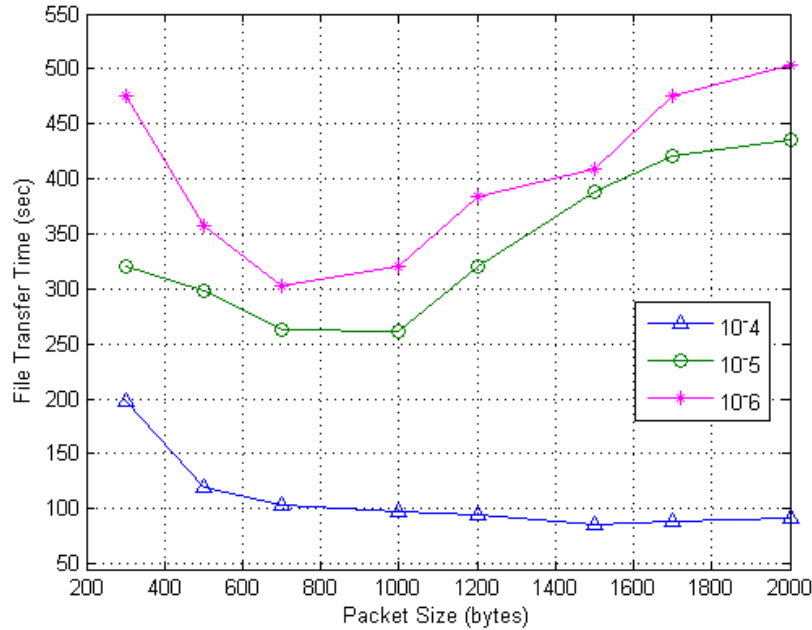


Figure 6.6: File transfer time of SCPS-TP as the function of packet size for different error rate

file transfer time is shown in the range of 1600-1800bytes; with no significant difference seen between two average points. Either 1600 or 1700 bytes can be considered as the optimal size. In the case when  $BER = 10^{-5}$  the smallest file transfer time is seen when the packet size of 1000 bytes, since lower the packet size, time take to transfer large file size is high to the propagation delay and when packet size is too high due to retransmission it has high file delivery time. These reasoning is true for lower BER ( $10^{-4}$ ), where small file delivery time is reached when packet size is in the range 600-800 bytes. Thus with the help of this experimental evaluation, the protocol module can fix the optimal packet size according to the BER for the given scenario if SCPS-TP is selected as protocol for transmission. There by improve throughput performance of the protocols.

Bundle protocol selected in this kind of traffic offer both reliable and unreliable transport using custody transfer service. As custody transfer need more buffer space for its service, in this experiment we have evaluated the buffer requirement for the given load for both with and without custody transfer. For this simulation we have considered the same topology of 3 nodes with link availability period of 80%. We calculated the buffer space required for various traffic load with bundle size of 100bytes (packet) with maximum transfer unit is set to be 1500bytes.

From the graph 6.7, we can see that, as traffic load increase, buffer requirement

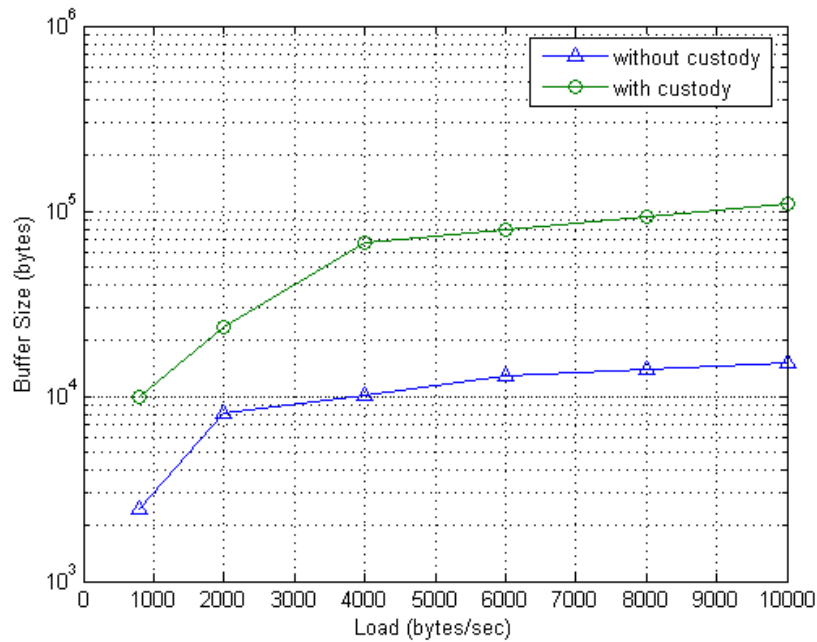


Figure 6.7: Minimum buffer size at different load with link availability period of 80%

for custody transfer service increase almost linearly compared to requirement without custody transfer. Thus protocol choice module check the buffer availability and compare that with requirement for given traffic load before selecting the protocol for transmission, by which it can attain high data delivery performance.

### 6.5.3 Multimedia Scenario

Existing rate control protocol like RAP, RCS, TFRC, cannot solve the rate control problem in InterPlaNetary Internet which is characterized by extremely long propagation delays, high link errors, asymmetrical bandwidth, and blackouts. Thus IPN research group has proposed a new rate control algorithm protocol, RCP-Planet, to address all the above challenges. RCP-Planet consists of two novel algorithms, i.e. Begin State and Operational State. Bandwidth asymmetry problem is addressed by FEC block-level ACKs. Moreover, RCP-Planet incorporates Blackout State into the protocol to improve performance in blackout conditions. RCP-Planet had novel rate-probing scheme, where it send extra redundant packets NIL, to probe the rate. The number of NIL packets  $L$  is a design parameter for the rate-probing mechanism. If  $L$  is too small, the observed rate might not be accurate enough to capture the available bandwidth. On the other hand, if  $L$  becomes larger, the overhead also becomes higher.

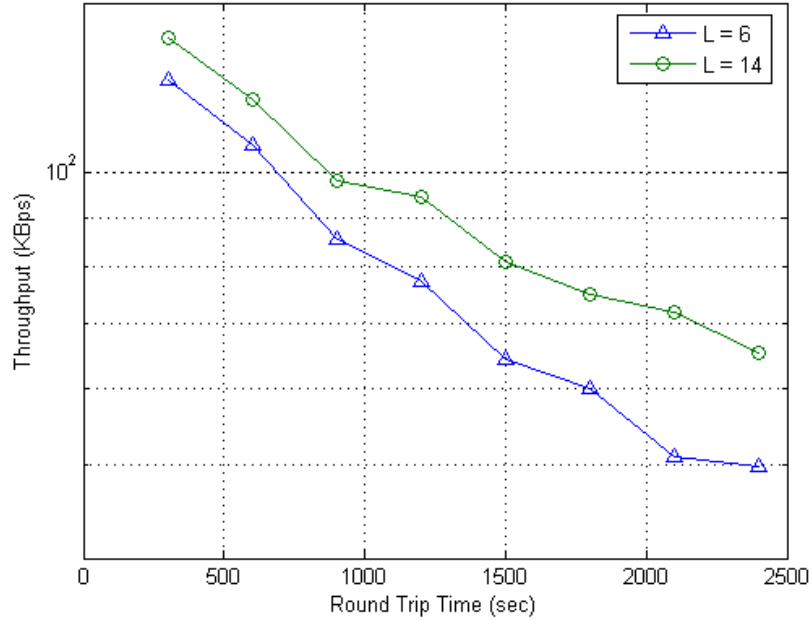


Figure 6.8: Throughput performance for different RTT with length of NIL packet is set as  $L=14$  and  $L=6$

Experimenters conducted by the author of the protocol [36] shows that when  $L = 14$  protocol achieve good performance results in all packet error rate. For this reason, protocol choice module set by default the NIL packet length to be 14, so that ability of the protocol to achieve higher performance is increased.

To show the performance of the framework when RCP-Planet is chosen, we conducted experiment using simulation to calculate the throughput performance achieve by protocol for different range of RTT. The throughput performance of protocol are shown in Figure 7.13 over a 10 Mb/s InterPlaNetary backbone link. The RTT value ranges from 300 to 2400 s including the RTT range for Mars-Earth communication, i.e. 8.540 min based on the orbital location of the planets. And we can see with default configuration of  $L$  ( $L=14$ ) protocol achieve high throughput for almost all RTT values. Thanks to the novel rate-based probing algorithm.

#### 6.5.4 Processing delay

In order to evaluate the processing delay introduced by the framework, we have considered command and control traffic application, where application cannot tolerate delay. For this experiment we have taken the same 3 node topology of interplanetary network, where ground station sends a command to the Rover to check its power sys-

Table 6.7: Process delay introduced by the TCL Middleware

	Processing Delay
without Framework	$3 \pm 0.96$ ms
with Framework	$230 \pm 18$ ms

tem for maintenance. The framework upon reception of transmission request from the application layer, check the type of application class and its QoS requirement from QoS Metric table and select the corresponding protocols choice from the mapping table and evaluate the selected protocol with the environmental condition for their suitability to obtain the required QoS. The selection process undertaken by the framework introduces further delay in addition to its transmission delays. In this experiment we calculate the process delay introduced by the framework over the selection process.

Table 7.2 below furnished the results of the experiment conduction to calculate the process delay introduced by the TCL middleware (delay introduced in construction of the TCL script for the simulation) as we can see from the results a considerable amount of delay is added into the system.

## 6.6 Discussion

In this work, we have proposed a QoS Framework, an autonomous middleware which resides alongside any protocol stack between application and transport layers. The framework help the application layer to select the most appropriate protocol for each transmission by comparing its requirements against environment constraints, and configures it to maximize performance. Its overall objective is to helps the interplanetary applications to choose optimum underling protocols, suitably measured in term of ability to achieve the application QoS requirements with the environmental constrains depending on the transmission scenario.

Positive results are produced in a range of scenarios from an implementation of the framework in NS2. As observed in the performance analysis the framework shows a positive impact in performance when the framework identifies environment constraints in relation to application requirements in advance of transmission beginning. In a limited selection of scenarios, the framework introduces overheads to the transmission without parallel improvements in performance.

However it is left in the future perspective, to make the QoS framework completely dynamic (autonomous) where, when the environment is dynamic and the framework takes intermediary action to autonomously accommodate changes

## Chapter 7

# Conclusion

Interplanetary overlay network and Delay tolerant network have emerged in the past couple of years as a possible means for extending the current Internet architecture to support various Mars missions. These networks are mainly characterized by the fact that connectivity between entities suffers from disruptions. From the link perspective, links could exist only intermittently, have highly asymmetric data rates, present large propagation delays or suffer from high error rates. At network scale, the heterogeneity of links and of communication stacks leads Internet-like networking solutions to fail. CCSDS and DTN are conceived to address these issues. This thesis represents our contributions to interplanetary network performance issues. We addressed the following issues:

1. Routing and Transport issues in Intermitted connected network
2. Resource Management issue in resource limited nodes in Mars surface
3. QoS issues in interplanetary applications
4. Interoperability and autonomous issues in interplanetary network nodes

In Sec. [7.1](#) we describe what we accomplished and the results we obtained. In Sec. [7.2](#), we present directions for future work to continue and extend the research described here.

### 7.1 Contributions

In this thesis, we made several contributions to interplanetary network.



First, we have analyzed the performance and ability of the various proposed protocols for interplanetary network. In particular the performance of CCSDS and DTN protocol and their limitations in the proposed reference Mars surface network scenarios. And characterized their performance metric and tabulated their limitations as the function of power, buffer and connectivity.

First in Chapter 3, we analyzed the performance of different routing protocols over resource constrained reference network, and came to a conclusion that there is a need for effective buffer management policy, since nodes in the reference scenarios (Rover/Landers) are resource limited. Thus, we have proposed a simple buffer management policy for these, resource limited nodes for both real time and best effort in particular image data traffic. We have then evaluated the proposed buffer management policy and identified it achieve good performance over the constrained resource limited nodes. As performance of the buffer management policy mostly depends upon the routing protocol, we shift our attention to a resource-aware routing protocol which adapts the proposed buffer management policy.

In Chapter 4, we have proposed an antenna selection algorithm for Mars Exploration Rover (MER), where MER can autonomously select which antenna to use between X-band Direct-To-Earth (DTE) and UHF for communication according to its constrains and limitation to effectively use their resources and thereby increase data return. The proposed algorithms make Rover more autonomous in terms of communication system. The proposed algorithm, defines a cost function for each possible path based on three parameters namely power required by the rover to transmit in that path, average delay, and error rate. A weighting is given to each parameter in the function based upon application Quality of Service (QoS) requirement. The algorithm has low computational complexity and provides optimum performance. The proposed antenna selection algorithm act as a base for the design of new source routing protocol called "resource-aware routing". Where the protocol uses knowledge about the connectivity and resource consumption of the nodes to make an effective routing decision to route all kind of traffics in interplanetary network, where communicating entities are of different natures. We present encouraging results which make Rover more autonomous to route the application specific data according to its resource availability in all most all extreme conditions with the help of effective buffer policy. As, future Mars missions depends on the ability of the Rover to operate autonomously in Mars Far-Term Communication Architecture, the proposed algorithm seems as a good approach to make Rover communication to with other entities autonomous.

In Chapter 5 we have proposed and implemented a dynamic timer algorithm for CCSDS File delivery protocol. The Consultative Committee for Space Data Systems (CCSDS) File Delivery Protocol (CFDP) is a new international standard developed by CCSDS to meet the growing space communication need for effective transfer of information in a wide variety of mission configurations of inter-planetary space links. The performance of CFDP in the deferred NAK mode has been evaluated in a theoretical manner. The delivery latency mainly depended on timer used in both sender and receiver. Thus an effective dynamic timer algorithm is proposed which set the timeout value according to present network condition there by reducing unnecessary retransmission of data PDUs. The performance improvements justify the need for dynamic timer algorithm in CFDP over asymmetric bandwidth connection in high error environmental conditions. As CFDP standards, allows users to implement their own timer and counter mechanism (user specific), our approach seems to provide a good timeout definition which achieves good delivery performance over the interplanetary networks.

Finally in Chapter 6, we have proposed a QoS framework which acts as a middleware reside between application and transport protocols, where the application layer selects the transport protocol which has ability to achieve its QoS requirements. Recent developments in the design of deep space mission have give rise to numerous interplanetary applications, other than simple data transfer from a distant nodes. And the main focus of researchers is to provide communication infrastructure in a challenging environment. To that extend, CCSDS and DTNRG have developed protocol stack which perform optimal at the given environmental condition. They do however posse some limitations. In addition, neither of the CCSDS or DTN protocol stacks contains autonomy or flexibility. Interplanetary networks applications are designed in such a way that it can work under any lower layer protocols. To use effectively the protocol resource, CCSDS deployed protocol selection based on application layer knowledge. CCSDS proposed three transport protocols to respond to varying QoS. While the idea of multiple protocols in attempting to satisfy QoS levels is considered effective, the protocols deployed are not. Thus the proposed QoS Framework makes application layer use the cross-layer technique to select the most appropriate underlying protocol for the given network condition to achieve the required QoS. The performance improvements obtained with the help of QoS framework over selection of suitable protocol and configuring according to the network condition are promising. And the application of cross-layer approach make application layer more effective in delivering the proposed QoS.

## 7.2 Perspectives

In this section, we present some of the research perspectives which we envision as a follow up to the thesis. We describe here, the future work that could extend the contributions of this thesis.

Future work that might follow our work on resource-aware routing in Chapter 4 includes software development of the protocol for emulation study, i.e created a Rover like node with resource limitation in real time and evaluating the performance of the protocol with definition of permanent buffer space and power source. Presently routing algorithm is simple unicast protocol, it would be interesting to extend the protocol to Hop-By-Hop multicast routing on a unicast infrastructure.

In chapter 5 a dynamic timer algorithm is proposed. And the performance of the proposed algorithm is analyzed over unreliable UDP protocol for simple data traffic for file transmission, as future work it would be interesting to investigate, other data traffic class like audio/video broadcasting and medical/meteorological images transfer, where they are characterized by different constraints on the maximum probability of data loss. And also it would be interesting to investigate the performance improvement of protocol with combination dynamic timer algorithm and erasure coding schemes in presence of very critical conditions, such as hard shadowing, deep fading periods and intermittent links.

Finally, in Chapter 6, where we have proposed a QoS framework which satisfying 2 of 4 properties of autonomic, namely self-optimization and self-configuration, and it is left to future work to make the framework completely autonomic by completing other properties like self-healing and self-protection. Presently Framework implementation in the simulation is semi-dynamic, in other words decision of transport protocols are made with help of predefined protocol and environmental parameters. It would be interesting to make the framework fully dynamic, where the feedback is given to the middleware directly from the system itself.

# Résumé - Contributions aux Études de Performances des Réseaux Interplanétaires

## Chapter I: Introduction

Mieux connaître l'espace a été, depuis l'antiquité, une fascination continue pour les humains. Depuis quelques décennies, l'humanité est capable d'envoyer des gens et des machines vers l'espace pour acquérir des connaissances précieuses sur l'histoire de notre univers, l'astrophysique, etc. Compte tenu des délais inévitablement impliqués dans les communications sur de telles distances, les observateurs ne peuvent pas collecter simplement des données et de les ramener à la terre quand ils reviennent. La transmission des résultats des mesures dès qu'elles sont acquises permet d'exploiter ces données rapidement et éventuellement de reconfigurer les équipements en fonction des informations. Les réseaux spatiaux sont devenus une réalité, avec les réseaux interplanétaires.

La Figure 7.1. donne une vision de ce qu'un réseau interplanétaire (Interplanetary Network IPN) pourrait être et de la complexité de leur architecture (les communications de surface ne sont pas représentées de plus). Des projets d'exploration spatiale de plus en plus complexes, impliquant plusieurs nœuds de communication tels que des satellites, robots ou "aerorobots" (ie capteurs atmosphériques). La plupart de ces nœuds peuvent communiquer directement avec la terre. Toutefois, cela provoque des questions concernant à la fois la bande passante et la consommation d'énergie, entre autres. Par exemple, en 2003/2004, les demandes de communications de et vers Mars étaient au moins 3 fois plus élevées que ce que permettaient les installations disponibles, affaiblissant considérablement les transferts des données et l'intérêt des différentes missions

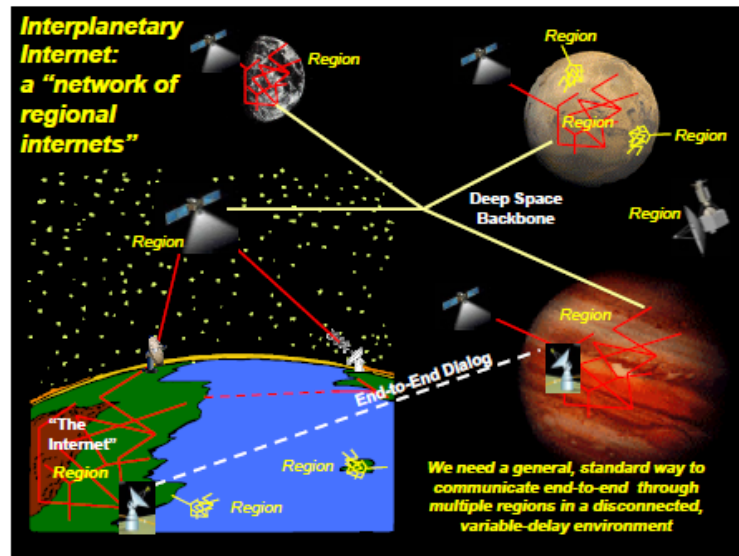


Figure 7.1: Interplanetary Internet - Future Mars Mission [Image Courtesy: NASA [19] ]

concernées. En outre, ces réseaux mettent en œuvre des liaisons inter-robot (réseau de capteurs) ou robot-communications par satellite, ce qui signifie que ces IPNs, par nature, ne correspondent plus à de simples liens interplanétaires point-à-point, mais constituent des réseaux complexes formé de nœuds hétérogènes. Ces réseaux soulèvent des problèmes difficiles, étant donné que la connectivité entre les nœuds est habituellement transitoire en raison des mouvements orbitaux et rotations de la planète. En conséquence, la communication avec certaines parties du réseau ne pourrait être possible que pendant des périodes précises. Pour cette raison, les missions spatiales présentent des besoins de communication assez différentes de ce que permettent les réseaux de télécommunication classiques, de sorte qu'on peut songer à développer des piles de protocoles spécifiques pour répondre à ces exigences.

## Contexte

L'Internet interplanétaire est un réseau informatique conçu dans l'espace, composé d'un ensemble de nœuds de réseau qui peuvent communiquer entre eux. La communication serait fortement retardée par les grandes distances interplanétaires, donc l'IPN a besoin d'un nouvel ensemble de protocoles et de technologies tolérantes aux retards et aux erreurs. Alors que l'Internet tel que nous le connaissons a tendance à être un "réseau de réseaux" à haut trafic, où les délais et les erreurs restent négligeables, et construit sur une solide connexion filaire, l'Internet interplanétaire est un "réseau des Internets"

de type Store-and-forward, souvent déconnecté, avec a un squelette sans fil entachées d'erreur de transmissions sur les liens et de retards pouvant atteindre des dizaines de minutes, voire quelques heures, même quand une connexion existe. Les technologies de communication spatiales n'ont cessé d'évoluer à partir d'architectures point-à-point couteuses, vers la réutilisation de la technologie sur les missions successives, et vers le développement de protocoles normalisés convenus par les agences spatiales de nombreux pays. Cette dernière phase à commencé en 1982 grâce aux efforts du Comité consultatif pour les systèmes de données spatiales (CCSDS) [1], CCSDS, un organe composé des principales agences spatiales du monde.

### **Pourquoi ne pas utiliser les protocoles terrestres dans les IPN**

Un des objectifs principaux de ces organismes de normalisation a été d'adapter les protocoles terrestres bien développés (TCP / IP [90], FTP [76]) dans les réseaux de communication " Deep Space ". L'Internet comme nous l'utilisons a été conçu dans le contexte où des entités (hôtes et routeurs) sont pour la plupart reliés par des liaisons filaires. Les protocoles et les applications qui ont été développés il y a trente ans sont encore en usage aujourd'hui. Ils s'appuient sur des hypothèses que nous décrivons ici:

- Connectivité de bout-en-bout: ils supposent que, entre deux nœuds quelconques qui peuvent communiquer les uns avec les autres il y a une liaison de bout en bout continue et bidirectionnelle.
- Des délais d'aller-retour: la connectivité de bout en bout, fondée principalement sur des liens câblés, conduit à supposer des durée de trajets allant de quelques millisecondes à une seconde.
- Débits symétriques: les débits de données sont supposées être à peu symétriques. Même si les réseaux d'accès comme l'ADSL sont asymétriques par nature, le niveau d'asymétrie est approprié au regard des besoins des utilisateurs. La liaison montante, qui a la plus faible capacité, a été suffisamment provisionnée pour la plupart des applications.
- Faible taux d'erreurs: des erreurs de transmission peuvent survenir pour diverses raisons comme la défaillance du lien ou la congestion, mais sont considérées comme étant inhabituelles et se présentent à un taux faible.

Un réseau interplanétaire diffère radicalement de l'Internet terrestre par un certain nombre des caractéristiques qu'on retrouvera plus ou moins mélangées:

- Coupures intermittentes de connectivité: la connectivité peut souffrir de perturbations menant au partitionnement du réseau, pour un grand nombre de raisons comme les problèmes de mobilité, les questions de la radio, et les questions de la batterie.
- Questions Délai: Des liens pourraient avoir un retard de propagation très élevé ou très variable rendant impossible le fonctionnement de protocoles comme le TCP traditionnel.
- Besoins de transfert des données asymétrique: Les liens peuvent être soumis à des débits de données très asymétriques voire simplement unidirectionnels.
- Taux d'erreur élevés: Quelques liens peuvent présenter des taux d'erreur élevés. Ils pourraient nécessiter un niveau élevé de correction et un grand nombre de retransmissions, conduisant à la création de goulets d'étranglement sévères.

Les défis posés par la communication dans un réseau interplanétaire, mettent à mal les performances des protocoles existants, et ouvrent les portes à la conception d'architectures de protocole plus efficaces. De façon plus détaillée, un effort particulier a été fait par les organismes de normalisation tels que le Comité consultatif pour les systèmes de données spatiales (CCSDS) et le groupe de travail CCSDS Delay Tolerant Networking [3] au sein de l'Internet Research Task Force (IRTF [4]). Le CCSDS a développé une pile de protocole complète (SCPS [7]), comme une alternative à la suite TCP / IP, en spécifiant les couches de protocole, en particulier approprié pour l'espace lointain. Le DTNRG a mis au point une architecture de réseau overlay nommé Delay Tolerant Architecture (DTN [94]), travaillant au-dessus de la couche de transport et capable de tolérer des interruptions et de longs délais sur les liens, grâce aux fonctionnalités offertes par le protocole Bundle [82].

### Les problèmes de recherche ouverts

En dépit des efforts pertinents déployés par la communauté scientifique, certains domaines de recherche ne sont encore que partiellement explorés. Plus spécialement, une certaine attention doit être portée aux problèmes de performances et aux couches de transport.

Dans ce travail, nous mettons l'accent sur des scénarios des réseaux interplanétaires (comme le réseau de surface de Mars), dans lequel les nœuds sont limités en

ressources et la connectivité réseau est d'intermittente. Dans ce contexte, nombre de problèmes de réseau doivent être abordés:

1. Transport: Comme nous le verrons dans le chapitre suivant, il est largement admis que les nœuds ont besoin de communiquer dans un mode " store-and-forward " et qu'ils ont alors besoin d'avoir des capacités de stockage pour transporter des données pour le compte des applications clientes. Mais un grand nombre de questions restent ouvertes dans ce domaine. Par exemple, quels types d'acquittements, de mécanisme de temporisation, de contrôle de flux et d'évitement de congestion faut-il utiliser?
2. Gestion des Ressources: Comme les appareils peuvent avoir une capacité limitée en termes d'énergie, de puissance de calcul et de stockage, quel type de politiques et stratégies doivent être définies pour le contrôle d'admission et de gestion de mémoire tampon?
3. Routage: dans un environnement où les nœuds ne communiquent que par intermittence, un chemin d'accès complet peut ne jamais exister physiquement entre une source et une destination. Le routage devient un problème difficile. Des stratégies de routage qui prennent en compte les propriétés d'un tel environnement sont donc nécessaires pour assurer la livraison efficace des messages.
4. Interopérabilité: Les nœuds du réseau interplanétaire peuvent avoir à communiquer via des réseaux internet classiques et à interagir avec des applications traditionnelles. Ainsi, l'interopérabilité, avec des solutions de sécurité, sont nécessaires sur les dispositifs planétaires et leurs protocoles.

Dans ce travail de thèse, nous nous sommes concentrés sur les problèmes de performance en essayant de comprendre les propriétés des nœuds interplanétaires et de l'environnement réseau qui pourraient permettre des solutions efficaces aux problèmes de routage et de décisions de transfert. Nous avons proposé quelques solutions pour le routage, le transport et les protocoles d'application dans le but d'améliorer leur performance. Les questions que nous abordons sont les suivantes: Quel type de gestion du buffer sont nécessaires compte tenu des ressources limitées du Rover?, Est-il possible d'avoir une sélection efficace de route fondée sur l'information environnementale?, Est-il possible pour l'application spatiale de sélectionner son protocole de transport conformément à sa qualité de service ?



Cette thèse est structurée comme suit. Tout d'abord, le chapitre 2 propose un état de l'art sur les réseaux interplanétaires, en donnant un aperçu des architectures, des applications et des protocoles qui ont été étudiés et déployés. Puis, les chapitres 3 à 6 présentent nos contributions dans l'étude de performance de ces réseaux interplanétaires. Enfin, le chapitre 7 conclut ce travail en résumant les contributions et en discutant des orientations futures qui pourraient le prolonger.

## Chapter II: Etat De l'Art

Ce chapitre donne un aperçu des travaux des recherches relatifs aux réseaux interplanétaires, à l'architecture, aux protocoles et leur normalisation. Les besoins de communication pour les futures missions de la National Aeronautics and Space Administration (NASA) ont révélé la nécessité de diverses activités de recherche visant à traiter les questions liées aux taux d'erreurs, aux retards, et aux liens intermittents dans les réseaux interplanétaires. Cela a conduit les planificateurs de ces missions, les concepteurs de systèmes, ingénieurs et scientifiques de la communauté spatiale à travailler ensemble à l'élaboration d'une architecture de communication réaliste. Ainsi, nous présentons d'abord les architectures à court et long termes dans les réseaux interplanétaire et leurs défis de communication en environnement d'espace lointain.

L'objet de ces recherches est principalement de répondre à l'interrogation sur la réutilisation des protocoles Internet existants dans l'espace lointain, de sorte à minimiser les coûts et les risques. Donc, nous discutons ici pourquoi TCP ne convient pas dans ces types de réseaux et la nécessité de nouveaux standards de protocoles. Enfin, nous présentons les différentes normes proposées comme alternatives et leurs limites dans un environnement difficile.

### Architectures réseau

Les développements en technologies spatiales permettent la réalisation des missions lointaines comme l'exploration scientifique de Mars. La vision de l'exploration spatiale à venir inclut des missions vers l'espace lointain qui nécessitent la communication entre les planètes, leurs lunes, les satellites, les astéroïdes, les sondes spatiales robotisées, et les véhicules habités. Cette vision s'inscrit dans la conception et le développement de la prochaine génération des réseaux de l'espace lointain, qui devrait offrir un service de type Internet en espace lointain, définie comme réseaux Interplanétaires [5]. Et la vision est de transformer en profondeur les réseaux de l'espace en un réseau interplanétaire

[101].

En [19] [73], une décomposition différente de l'architecture interplanétaire a été définie. Cependant, l'architecture est définie en fonction de plusieurs types de sous-réseaux appelé éléments architecturaux. Ces éléments architecturaux sont définis par la NASA en tant que:

1. Réseau fédérateur Mars-Terre: Cet élément d'architecture fournit les liaisons de données longue distance, directement entre les véhicules sur Mars [18] et la Terre ainsi que les éléments d'infrastructure basés sur Terre.
2. Véhicule Mars réseaux de proximité: Cet élément architectural fournit les liaisons de données entre la surface de Mars et les véhicules aériens et les engins spatiaux en orbite autour de Mars.
3. Réseaux locaux (LAN) à bord des véhicules de Mars: cet élément architectural relie entre eux les différents modules du véhicule grâce à un réseau interne composé d'un ou plusieurs types de bus série ou parallèle interconnectés.

### Architecture de Communication

Comme l'intérêt pour l'exploration de Mars croît et que les nouveaux véhicules sont déployés avec des outils de communications meilleurs et plus nombreux, l'architecture devra se doter de meilleures capacités [15]. Les sections suivantes décrivent les scénarios potentiels à court terme (jusqu'en 2010), à moyen-terme (jusqu'en 2020) et long terme (après 2020) pour le réseau de Mars [73].

### Défis des communications dans l'environnement spatial

Les énormes distances entre les entités appelées à communiquer, et la contrainte relativiste qui restreint la transmission du signal à la vitesse de la lumière, sont la cause de grands délais de propagation du signal. Il faut environ 4 minutes à la lumière pour parvenir de la Terre à Mars, lorsque ces planètes sont les plus rapprochées [32]. Ce retard peut dépasser les 20 minutes, lorsque la Terre et Mars sont en opposition.

Les caractéristiques uniques des entités communicantes et des liens dans le scénario spatial rendent l'utilisation directe de la pile de protocoles terrestres impossible [32] [13]. Cela est dû aux contraintes suivantes qui sont inhérentes à cet environnement:

1. Fort délai de propagation du signal

2. Fort taux d'erreur
3. Bande passante asymétrique et restreinte
4. Connectivité intermittente, évanouissements

### Caractéristiques de TCP sur un réseau fédérateur Mars-Terre

Une grande partie de l'étude décrite ici a ses racines dans un projet de recherche de la NASA pour développer un réseau interplanétaire, ou Internet interplanétaire (IPN). L'objectif principal des différentes agences spatiales est d'appliquer les protocoles terrestres bien connus dans le réseau de l'espace lointain. Pour cette raison, une étude a été faite sur le plus célèbre des protocoles terrestres, Transmission Control Protocol (TCP [90]) dans le réseau interplanétaire. L'idée de base est d'essayer de rendre les communications de données entre la Terre et les vaisseaux spatiaux (très) éloignés pratiquement aussi simples qu'elles sont entre deux personnes de différents côtés du monde. Il se trouve que, avant qu'un noeud du réseau puisse envoyer des données d'application utilisant le protocole TCP, un dialogue dit three-way handshake est nécessaire qui consomme 1,5 trajet d'aller-retour (RTT). Il ya aussi une temporisation générique, de deux minutes mise en œuvre dans la plupart des piles TCP: si aucune donnée n'est envoyée ou reçue pendant deux minutes, il y a rupture de la connexion. A partir de là, on voit bien qu'aucune communication ne sera possible dès que le vaisseau sera éloigné de plus d'une minute (à la vitesse de la lumière). Dans le cas de Mars, au plus près de la Terre le RTT est approximativement de 8 minutes, et peut atteindre 40 minutes dans le pire cas. On ne peut donc pas utiliser TCP sous ses diverses variantes terrestres pour les communications entre la Terre et Mars.

### Normes de Protocoles dans les missions en espace lointain

Depuis le début de l'ère spatiale, la NASA ainsi que des agences spatiales d'autres pays ont concentré leurs efforts vers la conception et la standardisation des protocoles nécessaires à l'exploration spatiale. Le Comité consultatif des systèmes de données spatiales CCSDS représente une telle tentative, qui est un succès. Le CCSDS File Delivery Protocole [26] est un protocole de transfert, flexible et efficace pour des missions très diverses de transfert de données. L'une des recommandations finales du CCSDS se présente sous la forme du Space Communications Protocol Standards (SCPS)[7]. Il s'agit d'une suite de protocoles conçus pour permettre la communication sur des environnements difficiles.

### Le CCSDS File Delivery Protocol (CFDP)

Le CFDP est un protocole de transfert, analogue au classique FTP, adapté au transfert de fichiers entre entités séparées par des distances interplanétaires, sur les liens spatiaux qui ont des bandes passantes asymétriques.

### Space Communications Protocol Standards (SCPS)

1. SCPS-FP [84]: Un protocole de traitement de fichier qui télécharge des commandes de satellites et des logiciels, et la collecte de données de télémesure, de façon optimale.
2. SCPS-TP [87]: Un protocole sous-jacent de retransmission de contrôle, qui fournit de façon optimale un transfert de bout en bout fiable pour la livraison de commande des engins spatiaux et des messages de télémétrie entre des ordinateurs qui communiquent sur un réseau contenant un ou plusieurs chemins de transmission potentiellement peu fiables.
3. SCPS-SP [86]: Un protocole de protection de données fournissant de bout-en-bout la sécurité et l'intégrité de chaque échange de messages.
4. SCPS-NP [85]: un protocole de mise en réseau évolutif prenant en charge aussi bien des liaisons avec ou sans connexion et acheminant ces messages sur les liens spatiaux.

### Delay Tolerant Networking Architecture (DTN)

DTN [50] [38] a été conçu comme un dérivé du projet IPN [5]. L'architecture DTN a pour but de fournir des communications interopérables entre un large éventail de réseaux qui peuvent présenter des caractéristiques de performance exceptionnellement pauvres et disparates. La conception englobe la notion de message de commutation avec stockage réseau et retransmission, la résolution différée de noms, le routage tolérant le cloisonnement réseau, pour créer des systèmes mieux adaptés aux opérations dans des environnements plus difficiles que la plupart des autres architectures de réseaux existants, en particulier aujourd'hui, TCP / IP Internet.

L'architecture DTN [94] définit une couche de superposition (overlay) orientée message de bout en bout, appelée le "Bundle Layer". Cette couche peut s'exécuter sur une grande variété de protocoles de transport et de niveau bas. Les dispositifs de mise en œuvre de la couche d'ensemble sont appelés nœuds DTN. La couche fournit un

mécanisme de stockage persistant pour lutter contre l'interruption de réseau. Il comprend un transfert de nœud-en-nœud (hop-by-hop) responsable de la livraison fiable avec un mécanisme facultatif d'acquittement. Les applications adaptées au DTN peuvent envoyer des messages appelés Application Data Units. Ces messages sont transformés en unités de données de protocole bundle appelées Bundles. Ces messages virtuels de commutation, utilisant un mécanisme Store-and-Forward fonctionnent d'une façon assez similaire au modèle de service postal.

### **Bundle Protocole (BP)**

Le protocole Bundle [82] prévoit des mécanismes efficaces pour communication dans/ ou au travers d'environnements à fortes contraintes environnementales, telles que la connectivité intermittente, les délais variables et longs, les forts taux d'erreur. Pour fournir ce service, BP forme un réseau overlay store-and-forward, inséré entre l'application et la couche de transport dans la pile de protocoles.

### **Licklider Transmission Protocol (LTP)**

LTP est un protocole de transmission point à point pour communications tolérant les délais mais fiable. Il est conçu pour servir de protocole de couche de convergence pour des tronçons d'un chemin de bout en bout dans un réseau tolérant les délais. Il agit au dessus de la couche liaison. LTP-T est une nouvelle proposition qui étend LTP en tant que protocole de transport. LTP-T est conçu pour un environnement multi-hop, donc la fiabilité et les questions liées deviennent plus complexes comparées à LTP.

### **Conclusion**

Dans ce chapitre, nous avons identifié ce qui est exactement le réseau interplanétaire et exploré diverses propositions d'architecture de communication pour réseau interplanétaire. Dans un premier temps, nous avons analysé diverses limitations posées par le milieu interplanétaire qui affectent directement les performances des protocoles de communication, TCP en particulier. Deuxièmement, nous avons étudié les limitations de diverses variantes du protocole TCP sur le réseau interplanétaire et le besoin de piles de protocoles nouveaux. Troisièmement, nous avons exploré diverses normes de protocole proposées par les organismes de normalisation spatiales comme le CCSDS et leur difficulté à fournir la qualité de service requise. Enfin, nous avons exploré la famille

de protocole appelé "Delay Tolerant Network, récemment mise au point et qui a été développée comme alternative.

De cette étude exploratoire, nous avons identifié les différentes limitations de performance posées par ces protocoles et défini un énoncé de problème excitants pour notre thèse, espérant fournir une solution efficace et des recommandations pour les protocoles sélectionnés pour atteindre de bonnes performances, dans les chapitres suivants.

## Chapter III: Performance Des Protocoles DTN Sur Les Réseaux Interplanétaires

Dans ce chapitre, nous passons en revue bon nombre des principes de l'architecture DTN [50], mettant en évidence les caractéristiques des protocoles DTN sur les réseaux planétaires et leurs limites. Dans un premier temps, nous analysons les performances de ces protocoles de routage qui semblent adaptables pour le réseau de l'espace lointain, dans le but d'étudier l'impact des ressources limitées (mémoire tampon) sur les performances des protocoles.

### Scénario de référence

Afin d'étudier les performances de ce protocole, nous avons défini un scénario de référence. Le scénario considéré dans ce travail comme l'illustre la figure 7.2. est composé de deux parties principales: la planète et les régions dorsales. De façon plus détaillée, chaque région planétaire comprend des noeuds planétaires (cercles blancs) qui travaillent à la fois comme source de trafic et noeuds de destination. La région dorsale est composée de noeuds interplanétaires (cercles noirs), siégeant en tant que relais de noeuds, reliés les uns aux autres grâce à une topologie en étoile. Enfin, les noeuds de passerelle spécialisées (noeuds en gris), responsables de la transmission de données, connectent les régions de la planète. La figure 7.2 rapporte le cas de deux région planétaires et une région dorsale. La région planétaire est composée de deux noeuds planétaires qui peuvent être soit Lander ou Rover, qui travaillent comme des sources de trafic et de destination. Les Landers sont des stations de base fixes et les rovers sont mobiles avec des déplacements aléatoires. Les noeuds planétaires sont reliés au réseau squelette par des passerelles qui sont les sondes orbitales, qui gravitent autour de Mars dans un mouvement orbital défini. La partie épine dorsale est composée de noeuds qui sont des stations sur Terre ou sur ses satellites. Les stations au sol travaillent à la fois comme source et destination de trafic. La figure 2 montre le noeud 1 des stations terrestres

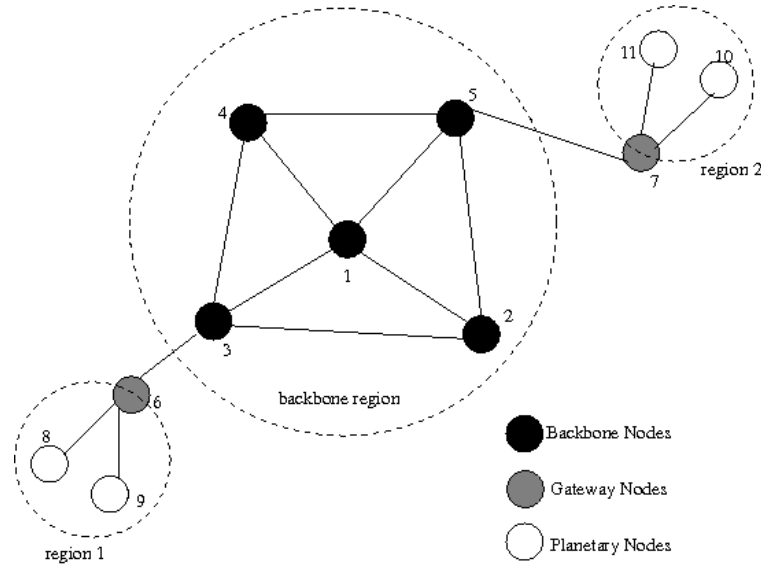


Figure 7.2: Scénario de référence

de la Terre; 2-5 nœuds sont des satellites relais de terre dans la région de la colonne vertébrale, 6,7 nœuds sont les orbites de Mars relais (nœuds de passerelle); Enfin 8-11 nœuds sont des nœuds planétaire qui peuvent à la fois envoyer et recevoir des données.

### Performances des protocoles de routage DTN

Le sujet du routage dans les DTN est presque devenu un domaine de recherche indépendant. Il y a eu plus d'une douzaine de différents schémas d'acheminement proposés [74] [91] [43], quelques thèses de doctorat [54] [106], et un certain nombre de documents [42] - surtout des études de simulation qui explorent les caractéristiques particulières d'un ou plusieurs algorithmes. Ceci est perçu comme une situation saine, et a été prévu lors de l'élaboration de l'architecture DTN. Zhang [105] propose une très bonne synthèse de ces propositions. Il existe diverses études menées sur les performances de ces protocoles [55] au-dessus de réseaux Ad-hoc [60], réseaux de capteurs intermittents [104], et des réseaux spatiaux [30]. Dans notre étude, nous avons l'intention d'analyser la performance des protocoles de quelques cas bien documentés sur le réseau interplanétaire (scénario de référence) par rapport à plusieurs contraintes comme l'énergie et des ressources de mémoire tampon.

Dans cette étude nous avons analysé les performances des protocoles, à savoir le simple Epidemic Routing [95] qui a été le premier proposé pour le routage DTN; PROPHET [56], un protocole de routage bien documenté, et le Minimum Estimated Ex-

pected Delay [91] qui utilise les informations de topologie pour les décisions de routage.

### Comparaisons de performance des protocoles ” DTN bundle ”

Dans cette section, nous présentons l’analyse des performances des protocoles de routage sélectionnés pour DTN sur le scénario de référence (figure 2). Toutes les simulations sont effectuées avec le Network Simulator NS2 version 2.29. Comme les noeuds du réseau interplanétaire ont des capacités de stockage limitées, toutes les analyses de performance visent à évaluer l’impact de la taille du buffer des noeuds sur le rendement du protocole. Bien que divers travaux de recherche [55] [74] aient déjà été faits sur différentes topologies de réseau, comme les réseaux ad hoc intermittents, les réseaux de capteurs mobiles et les réseaux satellites, notre intention ici de comprendre le fonctionnement des protocoles DTN sur les ressources limitées des noeuds planétaires, ce qui pourra servir à de futurs réglages des paramètres du protocole en vue d’en améliorer les performances (par exemple, la gestion mémoire, les décisions basées sur les ressources disponibles, etc.).

Puisque les protocoles de routage DTN fonctionnent sur le principe du ” Store-and-Forward ”, nous avons considéré le taux de livraison de paquets comme mesure de performance dans notre analyse. Ce taux est défini comme le quotient du nombre de paquets reçus par le noeud de destination, par le nombre de paquets de données transmis par le noeud source. La métrique est évaluée en fonction de trois paramètres différents à savoir la taille du tampon, la taille des paquets et la disponibilité du lien.

#### Performance sur liens intermittents

Étant donné la rotation de sondes, la circulation des Rover et des satellites de la Terre, les liens de communication sont le plus souvent intermittents. Il est donc intéressant d’analyser les performances du protocole de routage sur lien intermittents. Dans cette simulation, nous avons imité la nature intermittente en programmant le début et la fin des passages des Rover dans le NS2. Nous avons calculé le taux de délivrance des fichiers pour différents temps de disponibilité des liens, pour les trois protocoles. Le graphique de la figure 7.3, montre que les trois protocoles ont des performances limitées lorsque les temps d’inaccessibilité sont plus élevés, MEED et PROPHET se comportant relativement mieux que EPIDEMIC. Lorsque les temps d’inaccessibilité augmentent, davantage de paquets sont engendrés et les tampons à la source et dans les noeuds du voisinage sont saturés. Cette situation provoque la congestion au noeud source et aux



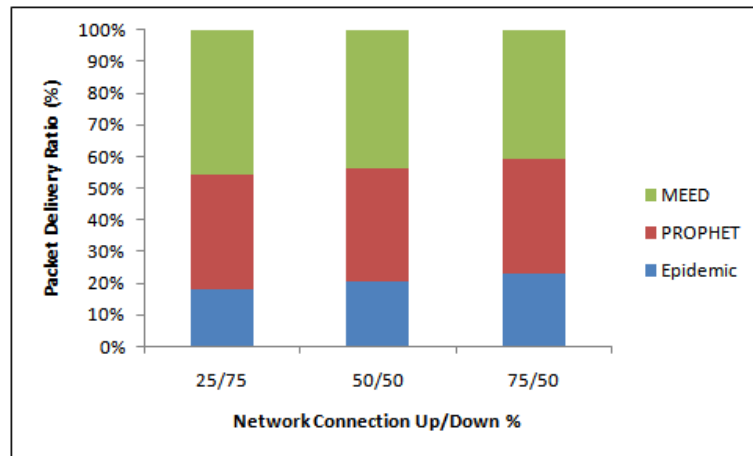


Figure 7.3: Taux de livraison des paquets avec liaisons intermittents

nœuds intermédiaires, ce qui conduit à des pertes de paquets qui réduisent le taux de livraison. Puisque chacun de ces Protocoles prennent en compte les tampons pour les décisions de routage pour éviter les encombrements de mémoire tampon. Pour cette raison, leurs performances sont optimales lorsque les tailles des tampons sont limitées.

L'analyse des performances du protocole sur réseau à liaisons intermittentes montre la nécessité d'une politique efficace de gestion des tampons pour éviter les encombrements de mémoire dans les nœuds à capacité limitée sur ce type de réseaux. Ainsi, dans cette étude nous proposons une nouvelle politique de gestion des tampons pour améliorer le taux de livraison des paquets et leurs délais. Dans la section suivante, nous élaborons le projet de politique de gestion de la mémoire tampon.

### Tampon Politique de gestion des DTN

Les futurs véhicules spatiaux et Mars Rovers auront la possibilité d'acquérir des quantités de plus en plus importante de données scientifiques. Les instruments d'imagerie multispectrale seront déployés afin d'envoyer autant d'informations que possible vers la Terre, au bénéfice à la fois de la communauté scientifique et du grand public. Toutefois, la taille du tampon embarquée est inévitablement limitée, et une partie des données doivent être jetés pour éviter les encombrements de mémoire tampon. Pire encore, dans les communications interplanétaires, la disponibilité de liaison est très limitée, en raison des longues périodes de latence; les bande passantes sont asymétriques. En outre, l'analyse des performances réalisées sur différents protocoles de routage DTN a montré la nécessité d'une bonne politique de gestion des tampons afin d'utiliser efficace-

ment les capacités afin d'atteindre de bonnes performances avec une latence minimale.

Les chercheurs ont proposé diverses solutions pour surmonter ces problèmes de bande passante asymétrique, de limitation de mémoire tampon, par des techniques comme de bons algorithmes de traitement d'image [81], des algorithmes de compression [88] et de stockage et des politiques de transfert. Des recherches sont menées sur le traitement d'image et les algorithmes de compression et aussi, indépendamment, la gestion des tampons. Mais dans cette étude, nous avons l'intention de fournir une politique simple et efficace qui combine à la fois le traitement des images et des techniques de contrôle de la mémoire tampon. D'où la proposition d'une nouvelle politique, combinant le codage d'image et le contrôle de la mémoire, afin d'optimiser la transmission de données en minimisant les pertes de données par débordement de tampon.

Le trafic sur le réseau interplanétaire peut être classé en deux sortes: le trafic en temps réel et le trafic best effort. L'ensemble des règles de contrôle proposées pour ces deux types de trafic est développé dans la section suivante.

### **Stratégie de contrôle pour le trafic temps réel**

Pour le trafic temps réel, par simplicité une politique de rejet basée sur l'information disponible localement est présentée, dans laquelle n'importe quel noeud du réseau qui détecte une congestion peut rejeter les messages en fonction de certaines règles sans aucune connaissance globale du réseau, puisque une politique basée sur une connaissance globale est difficile à mettre en œuvre. Étant donnée cette limitation, une question clé est: Lorsque la mémoire tampon dans le noeud devient encombrée, quel message devrait être supprimé de manière à optimiser une métrique de routage spécifique?

Dans le trafic en temps réel, le message le plus ancien dans le réseau satisfaisant à l'équation 1 est rejeté, lorsque le noeud détecte une congestion. Ceci parce que dans le cas du trafic en temps réel, le plus ancien message sera presque toujours obsolète par rapport aux messages nouvellement générés, il semble approprié de le supprimer. Les informations les plus récentes dans ce type d'application ont une valeur significative par rapport aux plus âgées de l'information (e.g. l'état actuel des batteries). Pour cette raison, dans ce type de trafic, les paquets en attente les plus récemment arrivés sont la transmis d'abord quand une fenêtre de communication est disponible.

### **Stratégie de contrôle des trafics Best Effort (données-image)**

Pour le trafic de données d'image, une combinaison de codage / tamponnage

**Algorithm 7.1** Politique de rejet pour le trafic dimages

When a packet is generated the TTL value of the packet is set according to its priority index. (High TTL value is set for High priority Packets). And let  $K$  number of packets are occupied in buffer at the node  $N$  and are stored in the ordered list according to their priority index value  $i$  such that  $K = k^i, k^{i-1}, k^{i-2}, \dots, k^{i-n}, i \in [1, n]$ . And a new packet  $k^x$  arrives at the node  $N$  with priority index of  $x$

**if** { Buffer storage of node  $N$  is not FULL } **then**

Insert the packet  $k^x$  in the ordered list according to the order of its priority index value with respect to other packets priority index in the buffer.

**else if** Buffer storage of node  $N$  is FULL **then**

Compare the priority of incoming packet  $k^x$  with priority index of last packet at end of ordered list  $k^{i-n}$  and

**if** {  $x \geq i - n$  } **then**

Drop packet  $k^{i-n}$  and insert the new packet  $k^x$  according to the order of its priority index.

**else**

Drop the incoming packet  $k^x$

**end if**

**end if**

est adaptée pour maximiser la transmission des données importantes et minimiser la valeur des données perdues pour cause de débordement de tampon. Pour cela nous avons proposé une politique de priorité où les paquets se voient allouer une priorité en fonction de leur signification par l'algorithme de traitement d'images, selon l'algorithme 7.1. Les paquets sont alors ordonnancés selon leur indice de priorité dans le tampon du nœud. Lorsque le tampon est plein la politique de rejet sélectionnée est appliquée. Ici les paquets émis sont prélevés en tête de tampon.

Comme on assigne aux paquets de priorité élevée une plus forte valeur de TTL et qu'on les classe au sommet de la liste, cela augmente la probabilité de délivrance. Ainsi, nous pouvons affirmer que la politique proposée améliore l'efficacité de débit des paquets de priorité élevée compte tenu d'une largeur de bande donnée sur le réseau à disponibilité intermittente. Et, puisque le paquet avec un indice élevé de priorité suffisent à récupérer les informations d'image entière, les performances (goodput) seront améliorées même si quelques paquets de faible priorité sont perdus par congestion du tampon.

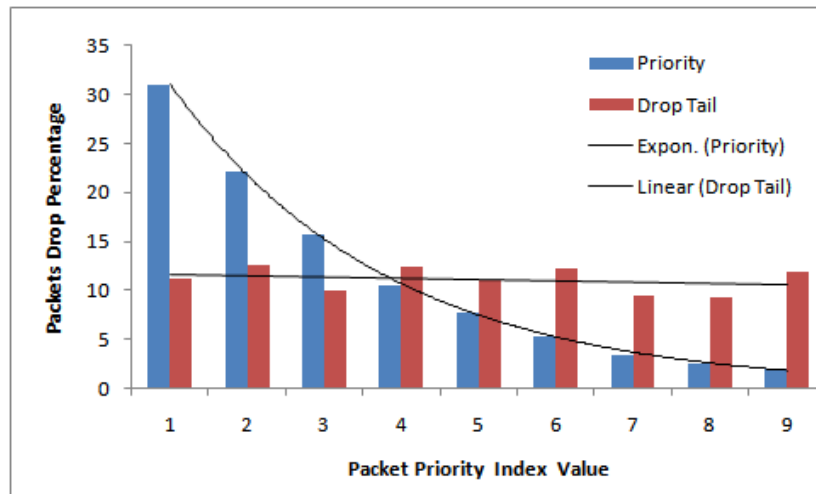


Figure 7.4: Comparaison des deux politiques ” Priority ” et ” Drop Tail ”

### Performance de la politique de gestion de mémoire tampon

Dans cette section, nous avons évalué les performances de la politique de gestion de mémoire tampon proposée pour les deux classes les trafics. L'évaluation des solutions a été réalisée grâce à NS-2. L'analyse des performances a été de deux ordres: macroscopiques et microscopiques. Dans le premier cas, l'étude analyse le rendement fourni par l'ensemble du réseau. Dans ce but deux métriques sont prises en compte: lae taux de livraison des paquets (PDR) et le Goodput. Dans le cas de l'analyse microscopique, l'attention a été accordée auxperformances fournies par le contrôleur de mémoire tampon au sein de chaque nœud.

### L'analyse microscopique

Nous n'avons considéré que la politique de rejet pour le best effort dans cette analyse. Pour comprendre l'influence de notre politique de gestion de la mémoire tampon, nous avons comparé la métrique pourcentage de perte de paquets (PDP) avec la politique Droptail (DT). Dans cette politique chaque paquet est traité de manière identique. Lorsque la file est remplie à sa capacité maximale, les paquets nouvellement arrivés sont supprimés jusqu'à ce que la file d'attente retrouve assez de place pour accepter le trafic entrant. Dans la Figure 7.4 nous pouvons voir que PDP de la politique de DT ont une distribution linéaire par rapport à l'indice de priorité des paquets, la raison étant que la politique de rejet traite tous les paquets de manière égale, avec la même probabilité de rejet. Dans notre politique, nous pouvons voir une distribution

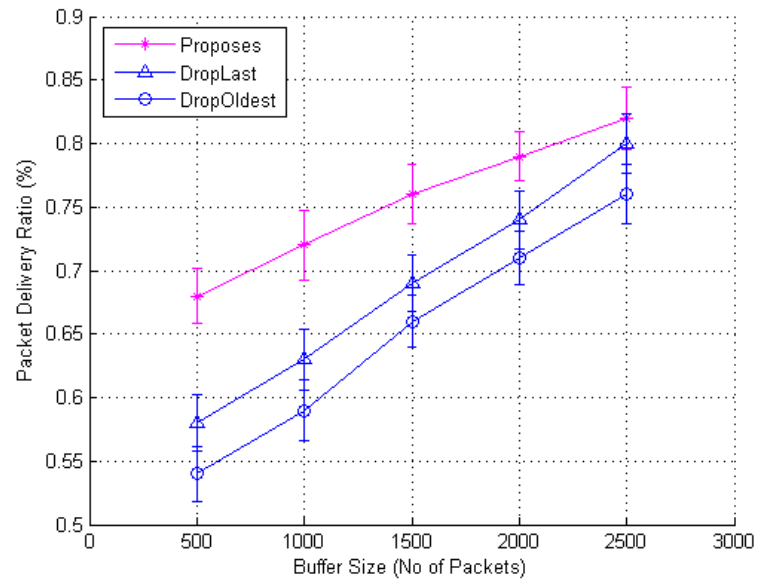
exponentielle par rapport à la valeur de priorité d'index des paquets.

### Analyse macroscopique pour le trafic temps réel

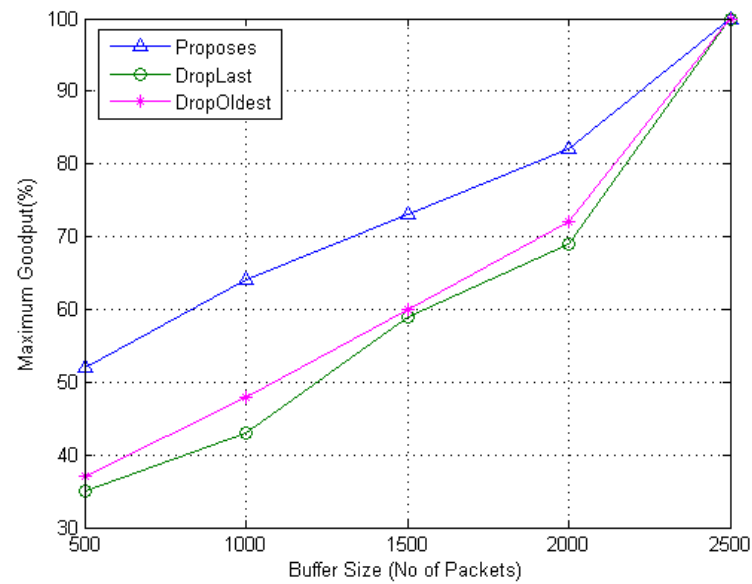
Nous avons comparé la politique de rejet proposée avec d'autres mécanismes, comme " Dtop Last (DL) et Drop Oldest (DO) pour comprendre l'efficacité de la politique proposée. La Figure 7.5(a) montre le taux de délivrance de paquets (PDR) pour toutes les politiques considérées dans la simulation. Nous pouvons voir dans le graphique que DL donne des résultats médiocres pour le trafic en temps réel, car il rejette les nouveaux paquets (information importante) et garde des paquets ayant une valeur inférieure du TTL restant, qui expirera avant d'atteindre la destination. Bien que DO et la politique proposée rejettent des paquets en fonction de leur valeur TTL, DO utilise la valeur résiduelle TTL (le plus ancien paquet dans la file d'attente), alors que la politique proposée agit en fonction de leur âge. Mais ils diffèrent dans la politique de transfert, DO envoyant les paquets en tête de tampon alors que la politique proposée vide le tampon par la fin. Les paquets en tête ont moins de temps restant que les paquet de queue et ils ont une probabilité de livraison à la destination plus faible, et donc DO aura un PDR plus mauvais que la politique proposée.

### Analyse Macroscopique pour le trafic Best Effort

La Figure 7.5(b) montre l'évolution des goodput par rapport à la taille du tampon. Quand la taille du tampon décroît les trois politiques voient leur performance se dégrader. Cependant la politique proposée se comporte mieux que les deux autres. Les données transmises par le Rover permettent de reconstruire les images grâce aux paquets de haute priorité (grâce à des algorithmes de traitement d'image), et puisque la politique proposée achemine les paquets de haute priorité avec une meilleure probabilité, elle permet d'atteindre une meilleure efficacité (goodput) comparée aux autres politiques (DO & DL). Dans ces deux cas, l'indifférence à la priorité des paquets explique les résultats médiocres. Ainsi, la dégradation du goodput est inférieure pour la politique proposée par rapport aux deux autres politique de rejet.



(a) PDR pour différentes tailles de tampon



(b) Goodput Performance

Figure 7.5: Analyse Macroscopique

## Chapter IV: Routage Base Sur Les Ressources Pour Réseau Interplanétaire

Le Chapitre 4 achève les travaux de recherche effectués dans le chapitre 3, en proposant une politique de routage basée sur les ressources disponibles pour un réseau de proximité sur Mars, en prenant en considération les limites des nœuds Rover / Landers telles que la puissance ou la mémoire. Dans un premier temps, un algorithme de sélection d'antenne pour les Rovers est proposé, qui sert de base pour la conception du protocole de routage basé sur les ressources.

### Motivation

Les futures missions d'exploration en surface de Mars intégreront des équipes humaines et des robots travaillant ensemble à des objectifs scientifiques et technologiques sur les surfaces planétaires. Par exemple, un géologue collectant des échantillons sur la surface de Mars travaillera avec un assistant robotisé pour annoter ou analyser ces échantillons. En conséquence, un système de communication souple et dynamique sera essentiel à la réussite de l'exploration de Mars. Actuellement, les nœuds de surface de Mars sont actionnées à distance depuis la Terre avec un détail de la configuration et des plans d'opération téléchargés dans les rovers. Alternativement, il serait intéressant d'intégrer une certaine intelligence dans les nœuds de surface (Rover / Landers) pour un fonctionnement autonome, de sorte que des reconfigurations ou des interventions manuelles seront très rares et limitées à ces circonstances exceptionnelles. Par ailleurs, les nœuds de la surface de Mars ont des ressources limitées (stockage, puissance). Ainsi, un protocole de routage unicast original est proposé dans ce chapitre, offrant aux nœuds des Rover / Lander nœuds suffisamment d'intelligence pour prendre les décisions de routage de façon autonome. Pour améliorer les performances, le protocole limite la décision de routage prise sur la base des ressources disponibles au nœud en question. La première étape examine les différents types d'antennes des Rover / Lander, du point de vue de leurs caractéristiques et des conditions dans lesquelles leur sélection permet d'améliorer les performances.

### Antennes de Rovers d'exploration de Mars

Les Rovers ont 2 antennes dans leur dispositif de communication: X-band Direct-to-Earth (DTE), Direct-de-Terre (DFE), conçues pour fonctionner à des débits plus faible avec une plus grande fiabilité. L'autre système est une radio UHF à plus

haut débit UHF avec antenne omni-directionnelle conçue pour communiquer avec des satellites en orbite autour de Mars. Les Rovers de Mars sont des ressources limitées : il existe 2 limitations principales:

- Limitation des ressources: les Rovers Mars sont très contraints en puissance. La puissance requise dépendra principalement du panneau solaire et de la conception du système ;
- Limitation de la communication: la sélection de la fenêtre de communication est le principal défi en matière de communication Rover.

Afin de sélectionner une antenne en fonction des ressources disponibles, un premier pas important est de comprendre les caractéristiques du lien en fonction de la puissance.

### Mars Rover Link Communication - Caractéristiques

De l'architecture générale IPN, nous avons identifié trois principales voies de communication possibles: Rover-DTE, Rover-Lander et Rover-Orbiter Link. Nous avons ensuite analysé les caractéristiques de chacun de ces liens en fonction de la puissance nécessaire pour la transmission, qui dépend de divers facteurs tels que, la bande passante de modulation, l'atténuation et la fréquence des liens. Nous avons ensuite présenté sous forme de tableaux les paramètres de la liaison en fonction de la puissance, qui agit en tant que condition fondamentale pour notre algorithme de sélection de l'antenne.

### Algorithme de sélection d'antenne basé sur le coût

Pour permettre au Rover une plus grande autonomie dans le choix d'une antenne de transmission en fonction des ressources disponibles, nous proposons dans cette section un algorithme basé sur le coût. Un ensemble de règles est défini, grâce auxquelles le Rover choisit son antenne. Fondamentalement, ces règles assignent un coût au choix de chaque antenne, et permettent au Rover de sélectionner l'antenne de coût minimum. Certaines hypothèses de base sont prises, afin de faciliter le processus de sélection. Ainsi on supposera connue l'énergie générée par le panneau solaire du Rover à l'instant donné. Egalement, la topologie du réseau est modélisée comme un graphe orienté dont les nœuds sont les points et les liens les arcs. La fonction de coût associée à chaque chemin d'accès est une fonction à valeur réelle donnée par equation

$$CostC \rightarrow R^+ \tag{7.1}$$



### Fonction de coût

La voie de communication est sélectionnée avec l'aide d'une fonction de coût simple. Elle est calculée en se fondant sur des paramètres de communication Rover, à savoir le retard de propagation, la puissance et le taux d'erreur de transmission par paquets, et est donnée par l'équation suivante

$$CostC(p) = \omega_1\delta(p) + \omega_2\psi(p) + \omega_3\xi(p) \quad (7.2)$$

où  $\omega_1, \omega_2, \omega_3$  sont la pondération de chaque paramètre, ces valeurs sont fixées en fonction de l'exigence de QoS de la demande et devraient être déterminées dans une méthode heuristique;  $\psi(p)$  est la puissance nécessaire au Rover pour transmettre les infos sur le parcours  $p$ ;  $\delta(p)$  est le retard total moyen sur le chemin  $p$ . Le retard a trois composantes: 1) le temps d'attente, 2) le délai pour la transmission, et 3) le délai de propagation. Le temps d'attente comprend le temps d'attente de disponibilité d'un noeud (temps d'attente) et le temps de vidage des messages déjà planifiés. Réduire les délais abaisse le temps que passent les messages dans le réseau, réduisant ainsi la compétition pour les ressources ; par conséquent, l'abaissement de retard améliore indirectement la probabilité de livraison des messages.  $\xi(p)$  est le taux d'erreur sur les paquets (PER), qui dépend du taux d'erreur binaire (BER) de la voie  $p$ .

### Les règles de sélection

Les pondérations pour chaque paramètre de la fonction de coût sont données en fonction des applications et de leurs exigences de QoS. L'ensemble de règles est défini ci-dessous:

- Limitation de puissance: à un moment donné, par exemple, lorsque l'énergie produite dans le Rover est faible, le  $\psi(p)$  est affecté d'un poids fort par rapport à d'autres paramètres de sorte que la trajectoire qui nécessite moins d'énergie est sélectionnés
- Contrainte d'erreur: Si les données d'application demandent un faible taux d'erreur alors on affecte  $\xi(p)$  d'un poids fort, de sorte que les chemins d'accès avec un taux d'erreur minimale seront favorisés.
- Contrainte de Retard: Si les données d'application nécessitent un faible délai de transfert, le même principe est appliqué, pour  $\delta(p)$  de sorte que le chemin d'accès avec un délai de transmission minimum est favorisé.

### Procédure de sélection

La sélection du chemin se fait en 5 étapes. Le schéma logique de l'algorithme est donné par la algorithm 7.2

---

#### Algorithm 7.2 Logic for path indexing

---

Sort costC =  $\{c(p_j) < c(p_{j+1}) < c(p_{j+2})\}$ , where  $c(p_j)$  is cost of path  $j$ ,  $j$  = index of the path, Path=0;

Let  $N$  = total packet size in bits, and  $R$  = data rate;

**while**  $j \leq 3$  **do**

**if**  $\{ \text{timeWindow}(p_j) \geq \frac{N}{R(p_j)+Mrg} \ \&\& \ N \geq \beta(p_j) \}$  **then**

    Path =  $p_j$

    return(Path)

**else**

$j = j + 1$  (increment the index)

**end if**

**end while**

return(Path)

---

### Implémentation et analyse

L'algorithme a été mis en œuvre sur Matlab avec une fonction qui renvoie le chemin d'accès sélectionné par le système de communication de la Rover, selon le résultat. Afin de comprendre l'impact de chacune des règles de l'algorithmique, une première simulation a été faite en tenant compte de la décision basée sur la contrainte de puissance. La simulation est faite pour 1 jour de Mars (sol) à partir de 6 h à 18hr. Le résultat de la simulation est présentée dans la figure 7.6. D'après le graphique comme prévu, on peut voir que Rover-link Lander est sélectionné pour une période assez longue. Mais en raison de débordement de la mémoire tampon dans les Lander, à 11hr LMST le Rover sélectionne le lien DTE (forte puissance) vers l'avant de Rover-Orbiter (qui n'est pas visible). Puis, lorsque l'orbiteur est disponible à 17hr LMST, l'algorithme recalcule la fonction de coût (un message est transmis entre Rover-Orbiter) et sélectionne Rover-Orbiter au lieu de la liaison à grande puissance DTE comme prévu. Le mécanisme de sélection d'antenne du Rover conduit intuitivement à sélectionner le nœud le plus proche pour la transmission de données comme les Lander, Orbiter ou une station sol. Cette étude prévoit de développer un nouveau protocole de routage par la source. Le routage par la source signifie que celle-ci spécifie comment les données

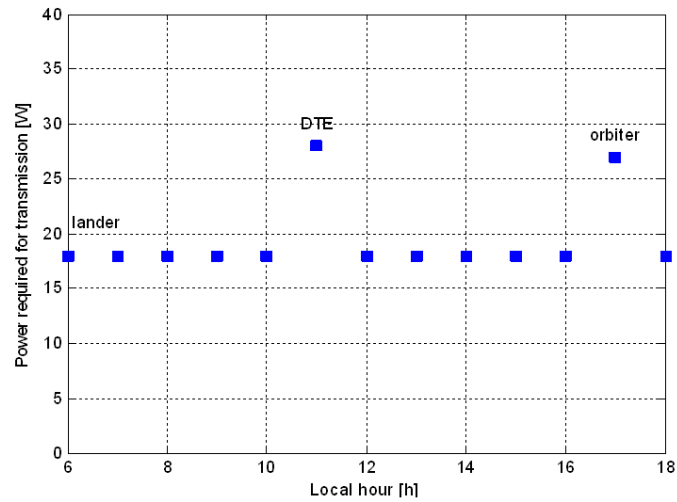


Figure 7.6: Contrainte de puissance

sont voyager à travers le réseau. Dans le scénario Mars, Rover sélectionne le prochain nœud avec lequel il transmet les données vers la Terre. Dans la section suivante nous décrivons les algorithmes de routage par la source proposés, de type "routage basé sur les ressources"

### Protocole de routage dépendant des ressources

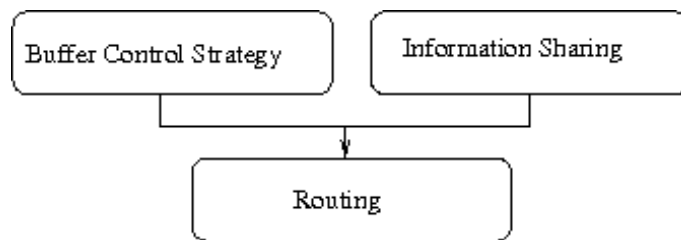


Figure 7.7: Bloc fonctionnel de protocole de routage

Un protocole de routage unicast par la source est proposé dans cette section. La figure 7.7 montre les trois blocs fonctionnels du protocole. L'objectif du protocole de routage est d'accroître les taux de livraison avec une latence minimum en utilisant toutes les connaissances disponibles concernant le réseau, et prenant également en compte la capacité disponible dans les nœuds. Chaque module du protocole de routage proposé est élaboré dans les paragraphes suivants.

### Stratégie de contrôle du tampon

Dans notre routage, nous avons utilisé la stratégie de contrôle de la mémoire tampon que nous avons développée dans le chapitre précédent, à la fois pour le trafic en temps réel et trafic best effort.

### Partage d'information et sélection d'antenne

Dans le protocole proposé peu d'informations à propos du réseau sont partagées entre les noeuds lors de leurs rencontres. Chaque fois que deux noeuds sont en contact ils partagent les informations suivantes avant tout transfert de données:

1. Espace tampon disponible: la différence entre le tampon occupé et la taille totale du tampon du noeud
2. Période de disponibilité du lien : durée prévisible pendant laquelle les deux noeuds d'extrémité du lien auront la possibilité de communiquer.
3. Epoque du prochain contact : la durée prévue après laquelle une paire de noeud se rencontreront, calculée à partir de l'historique des rencontres.

### Routage

Selon le type de données d'applications et leur exigence de qualité de service, différentes règles d'acheminement sont proposées, afin d'assurer une utilisation efficace des ressources. Comme le protocole est conçu pour le réseau de surface de Mars, le réseau se compose de 4 types différents de noeuds : noeuds de surface mobiles (Rover,  $m_i$ ), noeuds de station de base fixes (Landers  $s_i$ ), noeuds satellites riches en ressources (orbiteurs  $o_i$ ) et station au sol fixe (DSN  $g_i$ )

**Algorithme : logique de gestion du trafic nécessitant un transfert à haute fiabilité [7.3]**

**Algorithme : logique du protocole pour le trafic best effort [7.4]**

**Algorithme : logique du protocole pour trafic temps réel [7.5]**

---

**Algorithm 7.3** logique de gestion du trafic nécessitant un transfert à haute fiabilité

Let mobile node  $m_i \in M$  have application data which required to be send reliably to a ground station node ( $g_i$ )

**if** {sufficient power available  $P_{rq}^{m_i-g_j}(t) < \rho$  in  $m_i$  for DTE} **then**

    Select a node  $g_j \in G$  as next hope which is in the line of site of  $m_i$  (path 1).

**else if** power is not sufficient  $P^{m_i}(t) \leq \rho$  **then**

    Select a node  $o_j \in O$  , as the next hop which has sufficient buffer space availability  $\beta_i^j(t)$  and link availability period  $\zeta_i^j(t)$  (path 2).

**else**

    Select a node  $s_j \in S$  as the next hop which are in the communication range of the source node  $m_i$  with sufficient available buffer space availability  $\beta_i^j(t)$  and have minimum next contact time  $N_i^j(t)$  (highest likelihood to encounter orbital node in near future) (path 3).

**end if**

---

**Algorithm 7.4** logique du protocole pour le trafic best effort

**if** the traffic class is of best effort **then**

    Select a node  $s_j \in S$  as the next hop which are in the communication range of the source node  $m_i$  with sufficient available buffer space  $\beta_i^j(t)$  and have minimum  $N_i^j(t)$ .

**else if** orbit pass is available to source node **then**

    Select a node  $o_j \in O$  , as the next hop which has sufficient available buffer  $\beta_i^j(t)$  and link availability  $\zeta_i^j(t)$  period.

**end if**

---

**Algorithm 7.5** logique du protocole pour trafic temps réel

**if** the traffic class require minimum delay QoS parameter **then**

    Select a node  $o_j \in O$  , as the next hop which has sufficient available buffer  $\beta_i^j(t)$  and link availability  $\zeta_i^j(t)$  period.

**else if** no orbit pass available **then**

    Select a node  $s_j \in S$  as the next hop which are in the communication range of the source node  $m_i$  with sufficient available buffer space  $\beta_i^j(t)$  and have minimum  $N_i^j(t)$ .

**else if** sufficient power available  $P_{rq}^{m_i-g_j}(t) < \rho$  in  $m_i$  for DTE **then**

    Select a node  $g_j \in G$  as next hope which is in the line of site of  $m_i$ .

**end if**

---

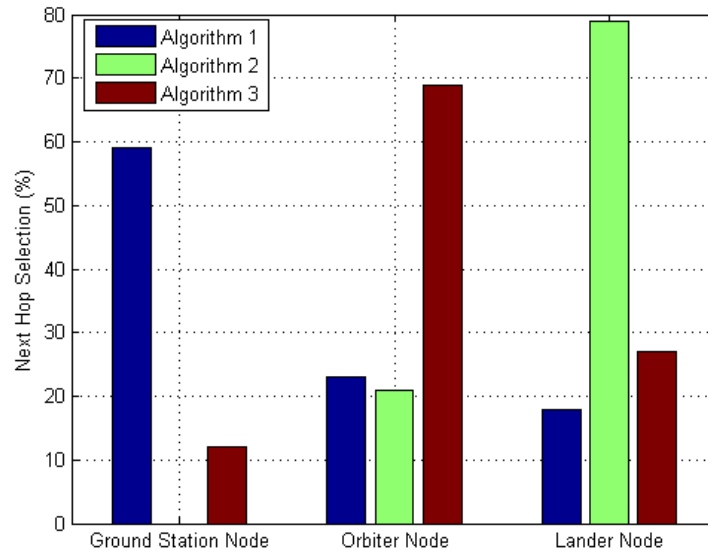


Figure 7.8: Proportion de sélection du prochain saut pour chaque algorithme

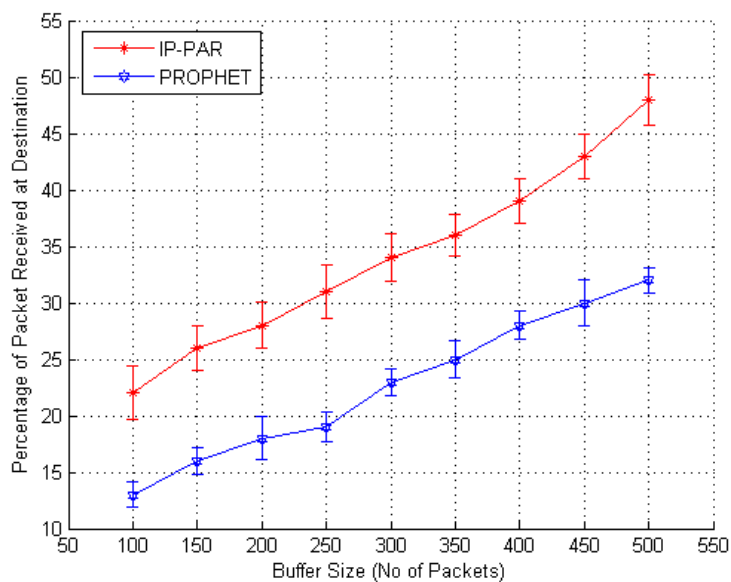
### Analyse de fonctionnalité

Pour bien comprendre l'implication de notre algorithme de routage, nous avons mis en oeuvre les fonctions de routage dans Matlab. La fonction de routage renvoie le nœud suivant sélectionné par les nœuds mobiles dans le réseau en fonction du scénario donné. Le graphique de la figure 7.8 indique le pourcentage de sélection du prochain saut pour chaque algorithme (best effort, transfert fiable, et trafic temps réel). On peut voir sur le graphique que dans la classe de trafic qui exige une plus grande fiabilité (Algorithme 7.3), le nœud source sélectionne le nœud de la station terrienne comme next-hop avec des pourcentages élevés par rapport aux 2 autres nœuds. Les nœuds orbitaux ou fixes ne sont sélectionnés que dans le cas où la puissance disponible est inférieure à la valeur seuil. Ainsi, ces 2 nœuds sont sélectionnés, soit au début de la journée (SOL) ou en fin de journée lorsque la puissance disponible est relativement faible dans les nœuds source. D'autre part pour le trafic best effort, les nœuds fixes sont majoritairement choisis comme prochain saut, à moins que le tampon ne soit plein. Et pour le trafic temps réel, le nœud orbital est choisi comme prochain saut avec un pourcentage élevé par rapport aux nœuds fixes ou aux nœuds station sol, ce qui évite les délais dans les tampons et les contraintes de puissance dans le système. Ainsi, on peut affirmer que la décision de routage est prise en fonction de contraintes de ressources.

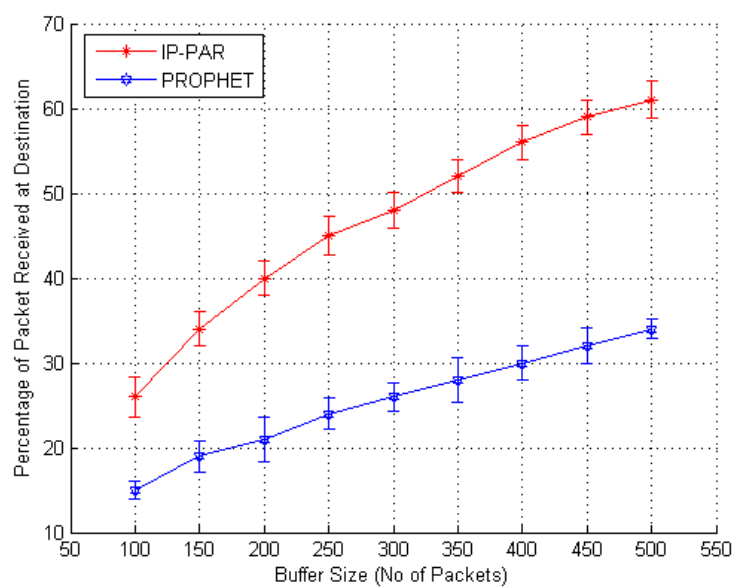
## Analyse de Performance

Le protocole est implémenté dans la version de 2.30 du Network Simulator (ns2). Le protocole implémenté est un routage source piloté par une table, où l'information de routage cohérente et mise à jour est maintenue à chaque noeud. Pour l'analyse des performances, une topologie interplanétaire simple est envisagée, qui se compose de deux noeuds source Rover, qui sont mobiles et suivent un modèle de mobilité au hasard avec vitesse très lente et le longs temps de pause. La taille des tampons des noeuds fixes et orbitaux est définie trois fois supérieure à celle de noeuds Rover mobiles. Pour chaque classe de trafic un taux de 4 paquets/sec avec paquets de 512 octets est sélectionné. Tous les graphiques sont les résultats d'une moyenne de 5 scénarios avec différentes conditions initiales pour la mobilité des noeuds. Nous avons comparé les performances de notre protocole de routage avec celles du protocole de routage probabiliste appelé PROPHET. La raison de ce choix est que les deux protocoles prennent la décision sur le prochain saut en utilisant les informations de contrôle qu'ils ont échangées au cours de leur précédente rencontre. Dans cette expérience, on fixe la durée de visibilité à 30 les noeuds relais avec des débit de données plus élevés. Notre intuition est que lorsque le temps de simulation augmente plus de paquets sont générés pour une taille de mémoire tampon et les tampons sur les noeuds de relais se remplissent rapidement. Cette condition entraîne une congestion au niveau des noeuds relais et cela conduit à des pertes de paquets qui réduisent le taux de livraison pour des débits plus élevés. PROPHET, qui ne considère pas l'occupation de la mémoire tampon des noeuds voisin, peut sélectionner un noeud qui n'a pas de tampons disponibles ou dont l'occupation de la mémoire tampon

Les graphiques de la Figure 7.9(a) 7.9(b) montrent les résultats de la simulation. Sans surprise, aucun des protocoles ne peut délivrer les paquets aux récepteurs lorsque la mémoire tampon est de faible taille. Cependant, notre protocole fonctionne relativement mieux que PROPHET [56], car il utilise plus efficacement les ressources tampons en partageant les informations préalablement à la transmission. Cet objectif est atteint au prix d'une augmentation de l'overhead en raison du partage d'informations. Sur la base de nos résultats, nous pouvons dire que nos protocoles se comportent de façon optimale dans la durée de visibilité, et qu'ils peuvent tolérer cet accroissement d'overhead.



(a) visibilité 70%



(b) visibilité 30%

Figure 7.9: Taux de livraison des paquets avec durée de visibilité 70% et 30%



## Chapter V: Algorithme De Temporisation Pour Le Ccsds File Delivery Protocol

Le protocole CCSDS de remise de dossier est le principal protocole de la couche d'application dans le réseau interplanétaire. De nombreuses recherches sont en cours pour améliorer les performances de ce protocole. Ce chapitre décrit en détail la nécessité d'un ajustement dynamique pour la temporisation du Protocole CCSDS de livraison de fichiers et propose un nouvel algorithme qui fixe la valeur de la temporisation en fonction des conditions réseau pour atteindre un débit plus élevé, en évitant la retransmission inutile de PDUs.

### Nécessité d'une temporisation dynamique

CCSDS File Delivery Protocol (CFDP) est une application de transfert fiable pour le réseau interplanétaire, dont les performances dépendent fortement des contraintes environnementales comme la bande passante limitée, les taux d'erreur élevés et les très longs retards de propagation. L'asymétrie de la bande passante affecte les performances du CFDP, parce que celui-ci se base sur le mécanisme de rétroaction pour garantir une transmission fiable et continue. Les délais de propagation, longs et fortement variables, peuvent compromettre le rendement, en forçant à d'inutiles retransmissions de PDU identiques, provoquées par des temporisations trop courtes. Ceci montre la nécessité d'un ajustement dynamique des temporisations, capables de s'ajuster aux conditions de l'environnement afin d'utiliser efficacement la bande passante et de fournir de bonnes performances.

### CCSDS File Delivery Protocol (CFDP)

CFDP est le protocole de livraison de fichier du CCSDS, une norme internationale pour un transfert de fichiers automatique et fiable entre les entités interplanétaire, construite sur les entités familières de la pile de protocoles CCSDS. C'est un protocole tolérant au délai dont le modèle de fonctionnement est fondamentalement store-and-forward, à la façon du courrier électronique qui transmet des fichiers en pièces jointes. Le protocole dans sa conception actuelle contient ses propres mécanismes de fiabilité et n'assume pas une capacité de retransmission sous-jacente. CFDP supporte les deux modes " avec-acquittement " et " sans acquittement ". Puisque de puissants codes correcteurs d'erreur sont capables de minimiser les pertes de données en matière de communication dans l'espace lointain, mais ne peuvent pas les éliminer complètement,

CFDP autorise les acquittements, dans sorte à détecter et corriger automatiquement par retransmission toute perte de données. Selon les exigences de la mission et la capacité de transmission, l'émission des NAK est classé en 4 types: différés, immédiats, invités, et asynchrones. Dans ce travail, l'accent principal est donnée en mode différé puisque le mode NAK différé convient parfaitement à un scénario réseau interplanétaire vers Mars, caractérisé par une intermittence de connectivité avec un délai de propagation élevé et variable.

### **NAK mode différé**

Dans le mode NAK différé, l'entité réceptrice enregistre toutes les informations sur les données manquantes jusqu'à réception de la fin du fichier. Elle émet ensuite un NAK pour demander les données manquantes. Le mode différé est appropriée au cas d'entités communicantes faiblement couplées, par exemple lorsque les distances interplanétaires introduisent des délais de propagation très longs.

### **Algorithme de temporisation dynamique**

Dans cette section, nous illustrons l'algorithme proposé de temporisation dynamique pour CFDP en mode différé dont le but est d'améliorer le temps de latence en minimisant le temps de livraison des fichiers. Nous avons défini un ensemble de règles pour ajuster la valeur du délai pour l'expéditeur et le récepteur en fonction de l'état du milieu. Dans le mode de transmission acquitté, l'entité CFDP entité a deux compteurs, à savoir les temporisations EDF et NAK pour assurer la retransmission des EOF et PDU perdus. Comme le temps total de livraison est la somme du temps jusqu'à la réception de l'EOF ACK et du temps pour recevoir tous les PDUs perdus, les deux compteurs ont un impact significatif sur le temps de livraison.

### **Préliminaires**

Pour simplifier, on fait l'hypothèse suivante. Tous les Meta data PDU et les fichier de données sont supposés avoir des longueurs identiques, des temps de transmission identiques, et les mêmes probabilités d'échec de distribution. Tous les NAK ont des longueurs identiques et des probabilités identiques d'échec. Les probabilités de perte d'EOF de PDU sont supposés être faibles. Le temps de traitement est supposé négligeable.

### Algorithme des temporisations EOF et NAK

Dans cette section, nous étudions l'algorithme de temporisation dynamique. Comme on peut le déduire de la description du protocole dans l'introduction, l'instant prévu pour fournir les fichiers dépend de certaines valeurs des paramètres que l'implémentateur peut choisir librement, par exemple, la valeur de temporisation EOF et le temps d'attente NAK. En général les valeurs d'attente pour les EOF et NAK sont fixées conformément aux recommandations de la mise en œuvre CFDP, à savoir deux fois le retard de propagation ( $2 * T_{prop}$ ), augmenté du temps de transmission de l'ACK (EOF) dans le cas de la temporisation EOF et de délai de retransmission du PDU demandé pour la retransmission de la k ème fenêtre pour la temporisation NAK, elles sont donc définies dans les équations suivantes.

$$T_{timeout}^{eof} = 2 * T_{prop} + T_{ack(eof)} \quad (7.3)$$

$$T_{nak}^k = 2 * T_{prop} + RT_k \quad (7.4)$$

### Temporisation EOF statique

Le mécanisme de temporisation EOF est simple, comme tout autre mécanisme du Protocole d'ordonnancement, avec des règles suivantes:

1. Lorsque EOF PDU est envoyé, l'entité émettrice démarre le compteur du temps avec la valeur du délai fixée conformément à l'équation 7.3
2. Lorsque le temporisateur expire, l'entité émettrice ré-émet le PDU EOF et recommence le compte à rebours avec la même valeur.
3. Lorsque l'entité émettrice reçoit EOF ACK PDU de l'entité réceptrice, elle désactive le compteur EOF et commence la phase de re-transmission le cas échéant.

### Temporisation NAK

Dans cette section, nous élaborons l'algorithme dynamique de la temporisation NAK, qui fixe la valeur du délai d'attente maximale du NAK de l'entité de réception. Pour rendre la temporisation dynamique, un facteur  $\beta$  est introduit dans l'équation 7.4 générale :

$$T_{nak} = \beta * T_{prop} + RT_k \quad (7.5)$$

Dans cet algorithme, le paramètre  $\beta$  est utilisée pour contrôler dynamiquement la valeur du délai d'établissement en fonction du délai de propagation. Nous avons également utilisé le compteur introduit dans la mise en œuvre CFDP, où ce compteur est défini comme la limite d'expiration de la temporisation. NAK. Nous avons utilisé une valeur de compteur fixe plutôt que dynamique puisque comme nous avons observé sur la simulation, une augmentation de la valeur du compteur au-delà d'un certain seuil n'a pas d'impact sur le délai de livraison de fichiers.

### Règles pour la temporisation NAK

Lors de la génération du premier NAK, la valeur de temporisation du timer  $T_{nak}$  Timer est fixée selon l'équation 7.5 avec la valeur  $\beta$  fixé à 2, et donc une temporisation égale à deux fois le délai de propagation plus le temps de transmission du PDU demandé (équation 7.4).

Lors de la génération des NAKs ultérieurs avant la fin du compteur, les règles suivantes sont appliquées:

- S'il n'y a pas de minuterie active (valeur du timeout = 0), alors un nouveau délai d'attente NAK égal à deux temps de propagation plus le temps de transmission du PDU demandé est utilisé
- S'il ya un timer actif et que le récepteur reçoit une PDU EOF (si EOF PDU est perdu), alors il compare la valeur du délai restant à deux temps de propagation : si le temps restant est plus grand, le récepteur démarre une nouvelle temporisation NAK et met comme valeur le temps restant augmenté du temps de transmission des PDUs demandés. Sinon, si le temps restant est plus faible que deux temps de propagation, le récepteur démarre une nouvelle temporisation, avec une valeur choisie selon l'équation 7.4, et arrête l'ancien compteur.

Lors de la génération des NAKs ultérieurs, après l'expiration de la valeur du compteur (suite à une connexion intermittente ou transactions multiplexés), les règles suivantes sont appliquées:

$$\beta_{new} = \beta_{old} + \frac{\mu}{\beta} \quad (7.6)$$

- Lorsque le compteur atteint le seuil ( $\mu$ ) avec transmission d'une succession de NAK sans aucune réception de PDU de l'expéditeur, alors le récepteur change la valeur de  $\beta$ , selon l'équation 7.6. Et il démarre la nouvelle minuterie NAK en définissant la valeur du délai selon l'équation 7.6 avec la nouvelle valeur de  $\beta_{new}$ . Et il ré-initialise le compteur.

Lorsque il reçoit les données de l'expéditeur, le destinataire mène l'action suivante:

- Lorsque le récepteur reçoit les données manquantes demandées de l'expéditeur, il compare la valeur restante de la temporisation avec la valeur normale (équation 7.7) :

$$T_{rem}^k \geq T_{nak} \quad (7.7)$$

2 fois le délai de propagation, plus le temps nécessaire pour recevoir les PUDS. Si la valeur restante est plus grande comparée à la valeur de délai d'attente, le récepteur réinitialise la valeur du timeout à la valeur standard, en supposant que la connexion réseau est comme avant. Si cette valeur est inférieure ou égale à la valeur standard, on conserve le paramètre de temporisation.

Les sections suivantes montrent l'amélioration obtenue en terme de délais de livraison de fichiers en utilisant l'algorithme proposé avec l'aide de la simulation.

### Mise en oeuvre et validation

Nous avons implémenté le protocole CFDP fiable au-dessus de connexions UDP non fiables, avec le mécanisme de fiabilité mis en oeuvre dans CFDP. CFDP est implémenté en utilisant C++, où la classe de couche d'application du protocole CFDP hérite de la classe d'application NS2, ainsi CFDP repose sur la couche application de l'architecture NS2. Tous les scénarios de simulation sont développés en utilisant le langage de scripts TCL conformément à la spécification de l'architecture NS2. Dans notre phase de validation, nous avons utilisé le modèle analytique développé par Daniel et al [53], où l'expression mathématique dérivée pour l'instant supposé de remise des données dans le mode NAK différé donne la façon de le calculer numériquement. Nous avons tracé les résultats de la simulation pour le temps de livraison de fichiers vs le nombre des PDUs envoyés : figure 7.10. D'après ce graphique nous pouvons voir que, il ya une augmentation de dans le délai de livraison de fichier lorsque un plus grand nombre d'octets est transféré. Et le résultat du modèle de simulation semble concorder avec le modèle d'analyse. Nous pouvons donc admettre que le code et le fonctionnement du modèle de simulation sont valides.

### Analyse de Performance

Dans cette section, nous présentons les résultats de simulation pour l'algorithme de temporisation dynamique sous diverses conditions. Le but est d'évaluer l'amélioration

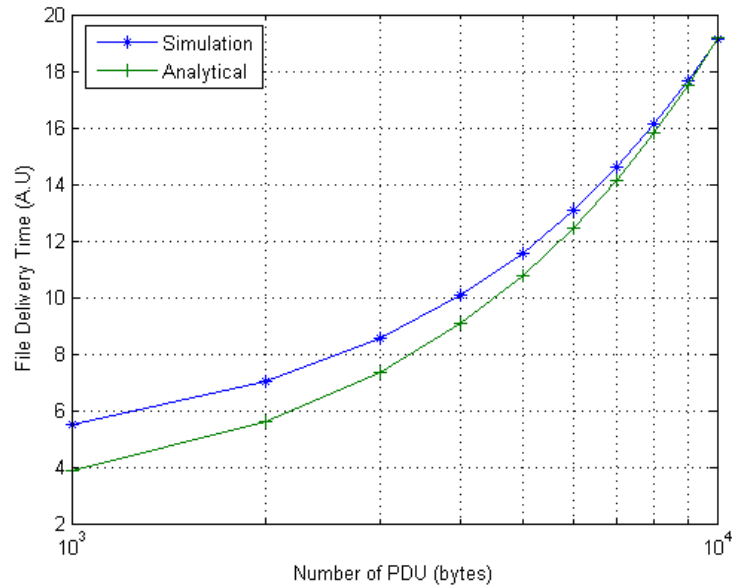
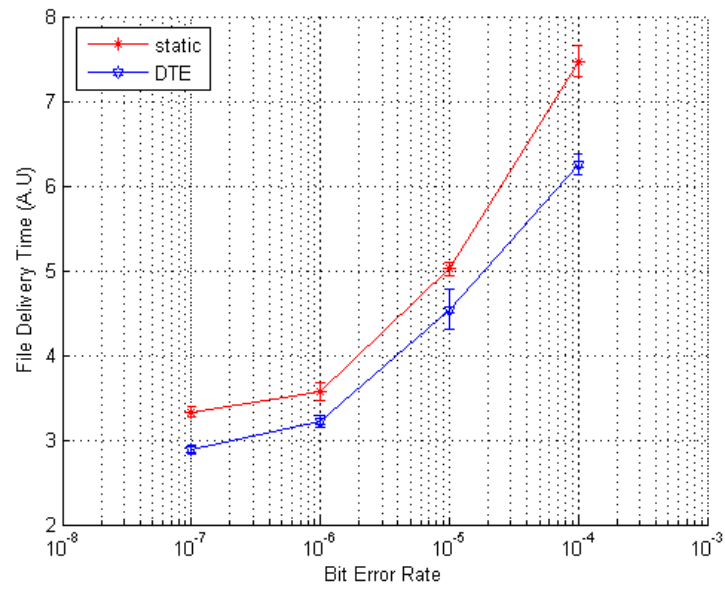


Figure 7.10: Validation du modèle de simulation

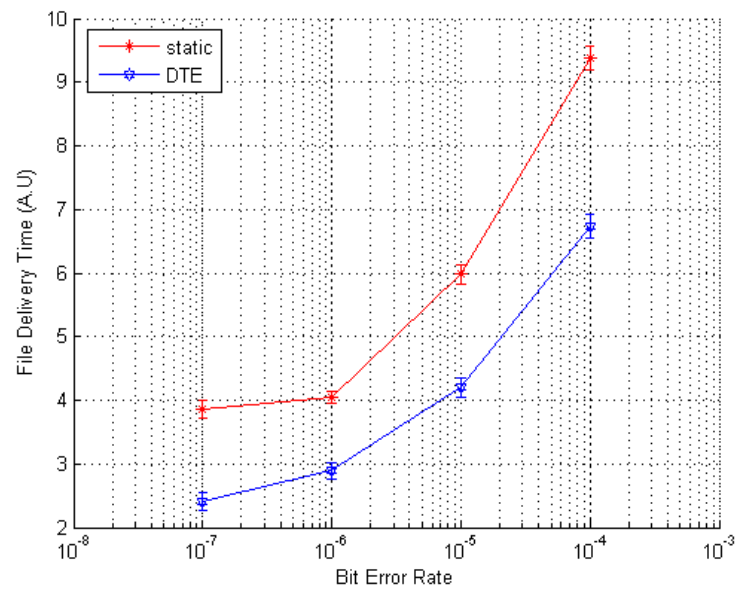
des performances de l'algorithme de par rapport aux temporisations statiques. La simulation se fait avec chaque BER et chaque délai de lien pour les canaux à la fois symétriques et asymétriques. Pour toutes les simulations, nous avons utilisé une topologie simple de réseau interplanétaire, où nous avons un noeud Rover et un noeud Lander. Les deux sont connectés au noeud orbital qui est relié à la station au sol. Dans notre analyse de la performance, nous avons considéré la moyenne de temps de livraison de fichiers comme principale métrique.

### Performance sur le canal symétrique et asymétrique

La Figure 7.11(b) montre le temps moyen de transfert d'un fichier de 1Mbyte avec le protocole CFDP en comparant les cas avec et sans timer dynamique sur le canal symétrique (20 kb / s: 20 kb / s) en fonction du BER, et pour un pourcentage de disponibilité de la liaison de 75 La Figure 7.11(a) montre le temps moyen de transfert d'un fichier de 1MByte avec le protocole CFDP en comparant les cas avec et sans timer dynamique sur le canal asymétrique (128 Kb / s: 20 Kb / s) en fonction du BER, et pour un pourcentage de disponibilité de la liaison de 75est clair qu'il y a une amélioration significative du temps de livraison du fichier avec l'algorithme DTA proposé. La raison en est que le DTA réduit le nombre de retransmissions à la fois dans dans les sens aller et retour grâce à l'ajustement dynamique de la temporisation, en réduisant le temps



(a) symétrique



(b) asymétrique

Figure 7.11: FDT sur canal asymétrique et symétrique

consommé par la transmission de paquets redondants supplémentaires, améliorant la délivrance des fichiers.

### Discussion

Dans ce travail, nous avons étudié l'application CCSDS de transfert de fichiers en espace interplanétaire lointain la plus commune, en analysant son fonctionnement et ses limitations sur les réseaux interplanétaires. Nous avons analysé différents travaux de recherche, destinés à évaluer la performance de CFDP et avons montré la nécessité d'un nouvel algorithme dynamique de fixation de la temporisation. Nous avons donc sélectionné le CFDP NAK en mode différé de la transmission en tant que sujet à étudier, car il s'adapte le mieux au réseau interplanétaire à délai variable. Puis nous avons proposé un nouvel algorithme dynamique de réglage de la temporisation, qui utilise la mise en œuvre d'un "compteur" spécifique pour fixer dynamiquement sa valeur en fonction de l'état du réseau. Nous avons également mis en œuvre le protocole CFDP dans NS2 avec notre algorithme dynamique pour évaluer ses performances par rapport à l'algorithme classique statique. Les résultats des performances montrent que l'algorithme dynamique proposé offre de bonnes performances en termes de taux de livraison et de retards sur les connexions asymétriques intermittentes. Puisque les normes CFDP permettent aux utilisateurs de mettre en œuvre leur propre mécanisme de temporisation, notre approche semble conduire à une bonne fixation de ces éléments, réalisant une bonne performance sur les réseaux interplanétaires.

## Chapter VI: QoS Based Protocol Selection In Interplanetary Network

Ce chapitre discute d'un cadre d'étude de la QoS dans un réseau interplanétaire. Un tel cadre aide les applications interplanétaires de choisir les protocoles sous-jacents de façon optimale, évalués judicieusement pour leur capacité à atteindre les exigences de qualité de service réclamées par l'application avec les contraintes environnementales.

### Motivation

Les développements récents dans la conception de missions spatiales dans l'espace lointain ont donné lieu à de nombreuses applications interplanétaires, autres que le simple transfert de données à partir de nœuds distants. Le principal objectif des



recherches est de fournir des infrastructures de communication dans un environnement difficile. Dans ce but, CCSDS et DTNRG ont développé une pile de protocoles qui offre un fonctionnement optimal compte tenu de l'environnement. Ils possèdent cependant quelques limitations. De plus, aucune des piles de protocoles, CCSDS ou DTN, ne prend en charge les critères d'autonomie et de flexibilité. Les applications pour réseaux Interplanétaires sont conçues de telle manière qu'elles puissent travailler quels que soient les protocoles des couches inférieures. Pour utiliser efficacement les ressources de protocole, CCSDS a déployé un mécanisme de sélection de protocole basé sur les connaissances de la couche application. CCSDS propose trois protocoles de transport pour répondre à divers critères de Qualité des services. Si l'idée de protocoles multiples pour tenter de satisfaire les niveaux de qualité de service est considéré comme efficace, les protocoles ne sont pas déployés.

Pour cette raison, nous avons ici tenté de combiner les deux protocoles DTN et CCSDS pour fournir davantage d'option dans la sélection de protocoles de QoS pour satisfaire la demande. Et aussi permettre un processus de sélection dynamique en prenant en compte l'état de l'environnement lors de la sélection. À cette fin, nous avons proposé un cadre de qualité de service qui agit comme intermédiaire entre l'application et le protocoles de transport, et où la couche application sélectionne le protocole de transport qui a la capacité d'atteindre ses exigences QoS.

### **Cadre de QoS pour sélectionner la pile de protocoles de communication**

Pour assurer l'autonomie de communication vers les nœuds de surface de Mars, nous avons proposé un cadre de qualité de service, comme le montre la figure 7.12. Ce cadre est intégré entre la couche application et la couche de transport. Travaillant dans une approche top-down, le choix du protocole est fait conformément aux informations combinées des métriques de QoS de l'application avec les données d'environnement mesurées dynamiquement. Le choix des protocoles sous-jacents en temps réel permet de maximiser la possibilité la possibilité d'atteindre la qualité de service requise dans des conditions réseau données. Le cadre proposé, avec la combinaison des métriques de QoS et des modules de mesure, est capable de construire la règle de mappage pour sélectionner le protocole sous-jacent pour la transmission de données avec les exigences de qualité de service prometteur. Sur réception d'une demande de transmission de la couche application, le cadre utilise la métrique de QoS de la couche application et les informations mises à jour sur l'environnement pour effectuer le contrôle d'admission (s'il y a lieu de commencer la transmission), puis avec les règles de correspondance de

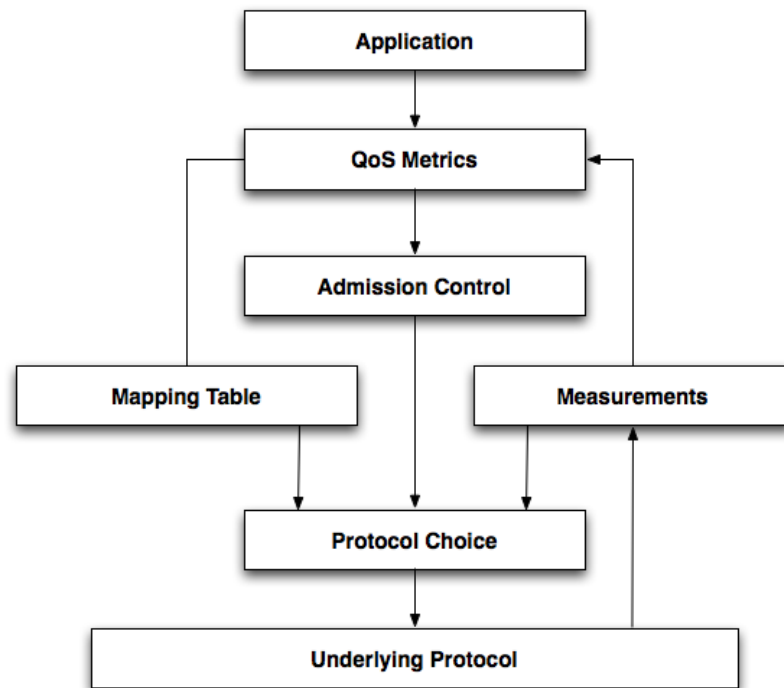


Figure 7.12: Cadre de QoS

sélectionner le protocole sous-jacent. Le cadre évolue à travers quatre états au cours de la communication, qui comprennent la mesure, de contrôle d'admission, l'évaluation des règles de correspondance et la sélection de protocoles (ou l'abandon).

### Les métriques de QoS des couches d'application

Les métriques de QoS des couches d'applications sont spécifiées en termes de délai acceptable maximum ( $D_{max}$ ), le débit minimum ( $B_{min}$ ), de taux d'erreur binaire maximum acceptable ( $E_{max}$ ), de volume de données ( $V_{min}$ ) ou best effort (sans QoS). Ces QoS de couche d'application représentent le type de trafic (classe de service) injectée par l'application source. Les métriques de QoS sont publiées une fois au début d'une communication. Elles contiennent des renseignements sur les exigences de performances des applications (délai maximum acceptable, débit minimum demandé).

### Module de mesure

Dans le cadre proposé, le module de mesure procède à trois relevés, le délai de propagation, la périodicité des liens et des taux d'erreur de lien. Le délai de propagation et la période de disponibilité du lien sont tirés d'éphémérides. Les données astronomiques

Table 7.1: Les tables de correspondance

Application	QoS Metric	Protocol
Telemetry	Delay ( $D_{\max}$ ) Throughput ( $B_{\min}$ )	TP-Planet / LTP-T
Scientific Data	Best Effort	Bundle Protocol / SCPS-TP
Multimedia	Delay ( $D_{\max}$ ) Throughput (constant)	RCP-Planet
Command and control	Delay ( $D_{\max}$ ) High delivery Rate	LTP-T / SCPS-TP (TCP based)

concernant le mouvement des planètes et des satellites ont été enregistrées et compilées sous forme de tableaux d'éphémérides par la NASA. Les taux d'erreur des liens évoluent en fonction des conditions atmosphériques comme le ciel clair, pluie, vent, etc. Des niveaux d'erreurs prédéfinis sont associés à chaque condition atmosphérique.

### Contrôle d'admission

Le contrôle d'accès est un élément important dans tout cadre de QoS, en particulier pour des réseaux en constante évolution, comme le réseau interplanétaire. Le contrôle d'accès rejette tout nouveau flot lorsque les ressources suffisantes ne sont pas disponibles. Le contrôle d'accès fonctionne sur des mesures environnementales. Dans notre cadre proposé, nous avons deux ensemble de règles d'admission, une pour les flux sensibles aux délais et l'autre pour les applications insensibles au délai.

- Lorsqu'une demande de connexion sensible au délai arrive, le module de contrôle d'admission compare le délai de propagation avec le délai maximum acceptable ; Si le délai de propagation est trop grand, le contrôle d'admission rejette le flux.
- Lorsqu'une demande non sensible au délai arrive, le module teste la capacité de stockage du nœud et accepte le flux seulement si la capacité disponible est supérieure au volume des données à transmettre.

### Les tables de correspondance

Dans le cadre proposé, les règles de correspondance sont basées sur des tables, c'est-à-dire qu'une liste prédéfinie de protocoles sont conservés dans une table avant le début de la transmission. Une règle de correspondance est une association entre une exigence d'application et une offre de protocoles. Les protocoles sont mis en correspondance

avec leurs capacités à fournir des QoS requises dans le milieu donné. Cette cartographie est basée sur les métriques de QoS, les informations mesurées et les possibilités du protocoles. Une table de correspondance (mapping table) entre les applications interplanétaires avec leurs exigences de qualité de service et les choix des protocoles est présentée dans le tableau 7.1:

### Choix du protocole

Le choix du protocole est l'étape finale dans le cadre proposé, où la couche d'application sélectionne le protocole sous-jacent le plus approprié à la classe donnée dans les conditions environnementales actuelles. Lorsque les modules de contrôle d'admission autorisent l'écoulement du trafic, le module de choix de protocole choisit le protocole de transport pour la classe d'application selon la table de mappage et évalue sa capacité à remplir son rôle compte tenu de l'environnement avant de valider le choix. On a tenu compte de certaines conditions du réseau en vue de prendre une décision optimale. Nous avons défini l'ensemble de règles de sélection pour les quatre classes d'application et pour les conditions possibles du réseau, pour choisir le protocole en fonction des capacités du réseau et les paramètres du protocole sont configurés en fonction de l'état de l'environnement avant la transmission de données.

### Comportement autonome

Tel que défini par IBM, un système doit posséder quatre propriétés pour être réellement autonome [49]: l'auto-configuration, l'auto-optimisation, l'auto-réparation, l'auto-protection. Selon cette définition, le cadre proposé n'est pas totalement autonome. Il s'auto-configue dans le choix du protocole de transport (et peut être étendu à d'autres couches sous-jacentes) pour adapter la qualité de service d'application sur l'environnement contraint. Des mesures permettent l'auto-optimisation, en utilisant des réponses passées et l'information pour définir les informations environnementales lorsque le même scénario se reproduit. L'auto-guérison et l'auto-protection ne sont pas inclus dans le cadre, ce qui ne le rend pas totalement autonome.

### Mise en œuvre

Nous avons mis en place le cadre proposé pour la QoS dans le simulateur (NS2). Nous avons utilisé le langage de script "TCL" pour élaborer ce cadre de qualité de service, qui agit comme un intergiciel (middleware) entre la couche application et la couche

de transport. Ainsi, le scénario développé lit un fichier de configuration des nœuds prédéfini, évalue les paramètres environnementaux et écrit un scénario de protocole pour la simulation. Au départ, nous avons mis en place un protocole de sélection semi-autonome. C'est-à-dire que les paramètres de l'environnement de chaque nœud sont pré-définis avant le début de la transmission. Par exemple, la disponibilité du tampon de chaque nœud est prédéfinie dans la configuration.

Des procédures TCL simples ont été élaborées pour chaque module dans le cadre de QoS. D'abord par souci de simplicité, nous avons supposé que la variation de délai de propagation était très faible, donc elle a été négligée et le module de mesure est prédéfini dans la configuration avant la simulation, qui devient donc plus statique que dynamique. Quand une demande d'application est reçue par la procédure-cadre, le module de cartographie utilise le fichier appConf pour mettre en correspondance les protocoles à l'aide de la table, et envoie ces informations vers le module de contrôle d'admission, qui utilise le fichier source envConf afin de déterminer s'il y a lieu d'admettre ou d'interrompre la transmission. Si le résultat est " admettre ", il transmet les informations recueillies au module de choix de protocole, qui utilise les fichiers envConf des nœuds possibles et construit le scénario de protocole pour la simulation et, enfin, le module de simulation.

### Analyse de performance

La performance du cadre est mesurée en termes de son délai de traitement pour sélectionner un protocole de transport dans le scénario donné. Puisqu'une série d'évaluations confirme l'aptitude du protocole, compte tenu des exigences de qualité de service de la demande et des capacités de fonctionnement du protocole, il est possible de dénombrer le temps de traitement supplémentaire introduit dans le système. Afin d'étudier la performance du cadre de la QoS, nous avons mené des expériences de simulation extensives. Le choix de protocoles de télémétrie et leurs performances avec et sans re-configuration sont évalués. L'effet de la taille du tampon et de la taille de paquets sur le débit des données dans les protocoles d'application est évalué. L'effet de la re-configuration dans le choix du protocole pour la performance de débit sur protocole multimédia est étudié. Le délai de traitement apporté par le cadre est évalué.

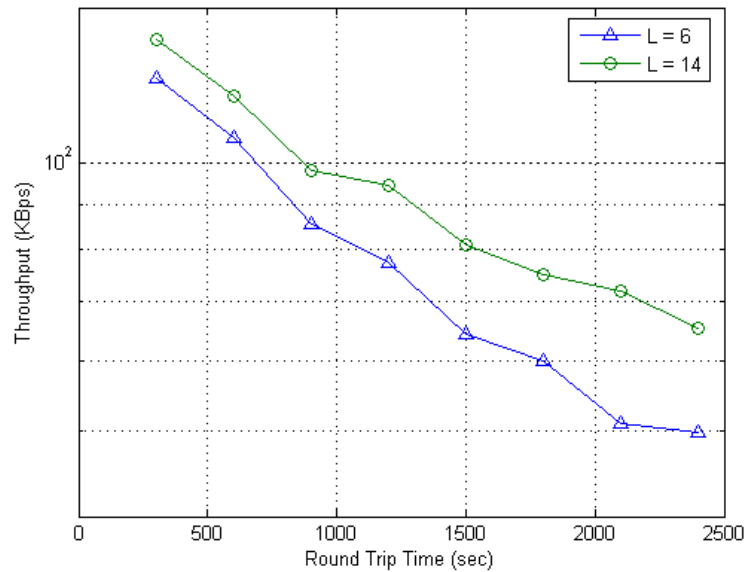


Figure 7.13: Performance de débit pour différents RTT

### Exemple: Scénarii Multimedia

Les protocoles de contrôle de débit existants comme RAP, RCS, TFRC, ne peuvent pas résoudre le problème du contrôle des taux en Internet interplanétaire, qui se caractérise par des retards de propagation extrêmement long, des erreurs de liaison, à grande bande passante asymétrique, et des pannes. Ainsi le groupe de recherche IPN a proposé un nouveau protocole de contrôle de débit, RCP-Planet, pour relever tous les défis ci-dessus. Pour montrer les performances du cadre lorsque RCP-Planet est choisi, nous avons effectué le test en utilisant la simulation pour calculer le rendement atteint par le protocole pour différentes valeurs de RTT. Les performances de débit du protocole sont indiquées à la figure 7.13 sur une liaison d'infrastructure base interplanétaire à 10 Mbit/s. Le RTT varie de 300 à 2400 s, ce qui inclut la gamme de RTT pour la communication Mars-Terre, à savoir 8,540 min fondée sur la position orbitale des planètes. Et nous pouvons voir avec la configuration par défaut de L ( $L = 14$ ) le protocole atteint un débit élevé pour presque toutes les valeurs de RTT.

### Retard de Traitement

Afin d'évaluer le retard introduit par le cadre, nous avons considéré les applications de commande et de contrôle du trafic, lorsque l'application ne peut pas tolérer de délai. Pour cette expérience nous avons pris la même topologie à 3 nœuds d'un réseau

Table 7.2: Retard de Traitement

	Processing Delay
without Framework	$3 \pm 0.96$ ms
with Framework	$230 \pm 18$ ms

interplanétaire, lorsque la station sol envoie une commande au Rover pour vérifier son système d'alimentation pour l'entretien. Le cadre, sur la réception de la demande de transmission de la couche application, vérifie le type de classe d'application et son exigence de qualité de service grâce à la table des métriques et sélectionne le protocole correspondant et évalue le protocole sélectionné par rapport aux conditions environnementales pour leur aptitude à obtenir la qualité de service requise. Le processus de sélection entrepris par le cadre introduit un délai supplémentaire qui va s'ajouter aux retards de transmission. Dans cette expérience, nous calculons le retard introduit par le cadre au cours du processus de sélection. Le tableau 7.2 fournit les résultats des expériences conduites pour calculer le retard introduit par le middleware TCL (retard introduit dans la construction du script TCL pour la simulation). Comme on peut le voir sur les résultats une quantité considérable de retard est ajoutée dans le système.

## Discussion

Dans ce travail, nous avons proposé un cadre pour la qualité de service, un middleware autonome qui réside à côté de toute pile de protocole entre l'application et les couches de transport. Le cadre aide la couche d'application à sélectionner le protocole le plus approprié pour chaque transmission en comparant ses exigences par rapport aux contraintes de l'environnement, et le configure pour optimiser les performances. Son objectif global consiste à aider les applications interplanétaires à choisir les protocoles sous-jacents de façon optimale, l'optimalité étant mesurée par la capacité à atteindre les objectifs de QoS des applications, compte tenu de l'environnement en fonction du scénario de transmission.

Les résultats positifs sont produits dans une série de scénarios à partir d'une mise en œuvre du cadre dans NS2. Comme observé dans l'analyse le cadre proposé montre un impact positif sur les performances puisqu'il identifie les contraintes de l'environnement par rapport aux besoins des applications préalablement au début de la transmission. Dans une sélection limitée de scénarios, le cadre introduit un overhead à la transmission sans amélioration parallèle de la performance. Toutefois, on garde comme perspective d'avenir, de rendre le cadre QoS entièrement dynamique (au-

tonome), de sorte que dans un environnement en évolution il saura prendre les actions transitoires pour tenir compte des changements de façon autonome

## Chapter VII: Conclusion

Les réseaux interplanétaires dédiés et les réseaux tolérant les retards sont apparus au cours des deux dernières années comme un moyen possible d'étendre l'architecture de l'Internet actuel pour soutenir les différentes missions, en particulier vers Mars. Ces réseaux sont principalement caractérisés par le fait que la connectivité entre les entités souffre de perturbations. Du point de vue du lien, des liens pourraient n'exister que par intermittence, auront des débits de données très asymétriques, présenteront de très grands retards de propagation et souffriront de taux d'erreurs élevés. A l'échelle du réseau, l'hétérogénéité des liens et des piles de communication conduit à l'échec des solutions de réseau construites sur le modèle actuel de l'Internet. CCSDS et DTN sont conçus pour répondre à ces questions. Cette thèse représente notre contribution aux problèmes de performances des réseaux interplanétaires. Nous avons abordé les questions suivantes:

- Questions de routage et de transport sur réseaux à connectivité intermittente.
- Question de gestion des ressources dans les nœuds à ressources limitées sur la surface de Mars
- Problèmes de QoS dans les applications interplanétaires.
- Interopérabilité et problèmes d'autonomie dans les nœuds du réseau interplanétaire.

Dans la Section nous décrivons ce que nous avons accompli et les résultats obtenus. En Section, on présente les orientations pour la poursuite des travaux futurs permettant d'étendre la recherche décrite ici.

### Contributions

Dans cette thèse, nous avons fait plusieurs contributions au réseau interplanétaire. Premièrement, nous avons analysé la performance et la capacité des divers protocoles proposés pour le réseau interplanétaire. En particulier, nous avons analysé la performance des protocoles CCSDS et DTN et leur limitation dans le scénario de référence proposé pour le réseau de surface de Mars, et caractérisé leurs métriques de performances



et précisé leurs limitations liées aux questions de puissance, de tailles de tampons et de connectivité.

Premièrement au chapitre 3, nous avons analysé les résultats des différents protocoles de routage sur des réseaux de référence à ressources limitées, ce qui nous a amenés à conclure à la nécessité d'une politique efficace de gestion de la mémoire tampon, puisque les nœuds dans les scénarios de référence (Rover / Landers) ont des ressources limitées. Ainsi, nous avons proposé une politique simple de gestion de ces tampons dans les nœuds à ressources limitées pour les trafics de type temps réel et best effort, en particulier dans le trafic de données d'images. Nous avons ensuite évalué la proposition de politique de gestion de la mémoire tampon et identifié qu'elle obtenait de bonnes performances sur les nœuds de ressources limitées. Bien que les résultats de la politique de gestion de la mémoire tampon dépende surtout du protocole de routage, nous nous concentrons dans le chapitre suivant sur un protocole de routage conscient des ressources disponibles qui adapte la politique de gestion de mémoire tampon.

Au chapitre 4, nous avons proposé un algorithme de sélection d'antenne pour les Rovers d'exploration de Mars (MER) Le MER peut choisir de façon autonome quelle antenne utiliser pour les communications, entre X-band direct-to-Earth (DTE) et UHF, en fonction de ses contraintes et des limitations d'utilisation permettant d'accroître les volumes de données transportées. Les algorithmes proposés rendent Rover plus autonome du point de vue du système de communication. L'algorithme proposé définit une fonction de coût pour chaque voie possible qui repose sur trois paramètres à savoir la puissance requise par le robot pour transmettre dans cette voie, le retard moyen, et le taux d'erreur. Une pondération est attribuée à chaque paramètre dans la fonction fondée sur l'exigence de Qualité de Service (QoS). L'algorithme a une faible complexité de calcul et fournit une performance optimale. L'algorithme proposé pour la sélection d'antenne ouvre la voie à la conception de nouveaux protocoles de routage par la source qu'on pourrait appeler "routage conscients des ressources".

Le protocole utilise les connaissances du système quant à la connectivité et la consommation des ressources des nœuds afin de prendre une décision de routage efficace pour acheminer tous les types de trafics sur un réseau interplanétaire où les entités communicantes sont de différentes natures. Nous présentons des résultats encourageants qui montrent un Rover plus autonome dans son processus d'acheminement des demandes de données spécifiques, en fonction de ses ressources disponibles dans la plupart des conditions extrêmes avec l'aide de la politique de la mémoire tampon efficace. La réussite des futures missions martiennes dépend de la capacité du Rover à fonctionner de façon au-

tonome dans des conditions extrêmes, de sorte que l'algorithme proposé apparaît comme une bonne approche pour faire ses communication avec d'autres entités autonomes.

Au chapitre 5, nous avons proposé, mis en place et validé un algorithme de temporisation dynamique pour le protocole CCSDS de livraison de fichiers. Le Consultative Committee for Space Data Systems (CCSDS) File Delivery Protocol (CFDP) est une nouvelle norme internationale développée par l'agence spatiale pour répondre aux besoins croissants de communication spatiale pour un transfert efficace d'information dans une grande variété de configurations de liaisons inter-espace planétaire. La performance du CFDP dans le mode " NAK différé " a été évaluée d'une façon théorique. La latence dépend principalement de la temporisation utilisée dans l'émetteur et le récepteur. Ainsi, un algorithme dynamique efficace est proposé qui fixe la valeur instantanée de ce délai en fonction de l'état actuel du réseau, conduisant à réduire les retransmissions inutiles de PDUs de données.

Les résultats de performance présentés dans l'étude prouvent la nécessité d'un choix dynamique de la valeur de cette temporisation sur une connexion de bande passante asymétrique en conditions de taux d'erreur élevés. Puisque les normes CFDP permettent aux utilisateurs de mettre en œuvre leurs propres mécanisme de compteur et de temporisation (utilisateur spécifique), notre approche semble donner une solution à ce problème qui offre une bonne efficacité de livraison sur les réseaux interplanétaire.

Enfin au chapitre 6, nous avons proposé un cadre pour la qualité de service qui agit comme un intermédiaire entre les protocoles d'application et de transport, de sorte que la couche application puisse sélectionner un protocole de transport qui aura la capacité à atteindre ses exigences de QoS. Les développements récents dans la conception de missions dans l'espace lointain ont donné lieu à de nombreuses applications interplanétaires, autres que le simple transfert de données à partir de nœuds distants. Le principal objectif de la recherche est ici de fournir des infrastructures de communication dans un environnement difficile. Dans ce but, CCSDS et DTNRG ont développé une pile de protocoles qui se comportent de façon optimale compte tenu d'un environnement donné. Ils présentent toutefois certaines limitations. En outre, aucune de ces piles de protocoles, CCSDS ou DTN, ne prend en charge les contraintes d'autonomie et de flexibilité. Les applications sur les réseaux interplanétaires sont conçues de telle manière qu'elles puissent travailler quels que soient les protocoles des couches inférieures. Pour utiliser efficacement les ressources de protocole, CCSDS a déployé un mécanisme de sélection de protocole basé sur les connaissances que possède la couche application. Trois protocoles de transport ont été proposé, pour répondre à divers critères de Qualité des

services. Si l'idée de protocoles multiples pour tenter de satisfaire différents niveaux de qualité de service est considéré comme efficace, les protocoles ne sont pas déployés. Ainsi, le cadre de QoS proposé rend la couche d'application capable d'utiliser une technique d'inter-couche (cross-layer) pour sélectionner le protocole sous-jacent le plus approprié aux conditions pour assurer la qualité de service requise. Les résultats des performances obtenues avec l'aide du cadre de la QoS sur la sélection de protocole approprié et la configuration en fonction de l'état du réseau sont réellement prometteurs. Et la mise en œuvre de l'approche multi-couche rend la couche d'application plus efficace pour garantir la qualité de service proposée.

### Perspectives

Dans cette section, nous présentons quelques-unes des perspectives de recherche que nous envisageons comme une suite de la thèse. Nous décrivons ici les travaux futurs qui pourraient en étendre les contributions.

Le travail sur le protocole de routage conscient de l'état présenté au chapitre 4 pourrait se poursuivre par une implémentation réelle du protocole pour une étude d'émulation, c'est-à-dire par la conception d'un nœud reproduisant le Rover avec ses ressources limitées et une évaluation des performances du protocole en temps réel compte tenu des ressources en énergie et en espace mémoire. Actuellement le protocole est un simple mécanisme unicast, il serait donc intéressant d'en étendre la conception à un routage multicast sur une infrastructure unicast.

Dans le chapitre 5 un algorithme de temporisation dynamique est proposé. Les performances de l'algorithme sont analysées sur une liaison UDP non fiable, pour acheminer les données d'un simple transfert de fichiers.. Il serait intéressant d'étendre ce travail en analysant d'autres classes de trafic, telles que de la diffusion audio/vidéo ou du transfert d'images médicales ou météo, applications caractérisées par des contraintes différentes, en termes de probabilité de pertes maximale tolérées. Il serait également intéressant d'analyser l'amélioration des performances amenées par la combinaison de l'algorithme de temporisation dynamique et des schémas de codage en présence de conditions extrêmes, telles que des zones d'ombre étendues, des périodes d'évanouissement du signal (fading) et des connexions intermittentes.

Enfin, au chapitre 6, où nous avons proposé un cadre pour la fourniture de la qualité de service qui satisfait 2 des 4 propriétés d'un système autonome, à savoir l'auto-optimisation et l'auto-configuration, et il reviendra aux travaux futurs de rendre ce cadre complètement autonome en y ajoutant d'autres propriétés telles que l'auto-guérison et

l'autoprotection. Actuellement la mise en œuvre du Cadre dans la simulation est semi-dynamique, en d'autres termes les décisions de choix du protocole de transport sont effectuées avec l'aide de fichiers prédéfinis (envconf et appconf). Il serait intéressant de rendre le cadre entièrement dynamique, de façon à ce que ce soit le système lui-même qui fournisse directement à l'intergiciel les informations nécessaires.

# Bibliography

- [1] CCSDS - Consultative Committee for Space Data Systems.  
URL:<http://www.ccsds.org/>.
- [2] DARPA - Defense Advanced Research Projects Agency.  
URL:<http://www.darpa.mil/>.
- [3] DTNRG - Delay Tolerant Networking Research Group.  
URL:<http://www.dtnrg.org/wiki>.
- [4] IETF - The Internet Engineering Task Force. URL:<http://www.ietf.org/>.
- [5] IPN - Interplanetary Internet Project. URL:<http://www.ipnsig.org/home.htm>.
- [6] NASA - The National Aeronautics and Space Administration.  
URL:<http://www.nasa.gov/>.
- [7] SCPS - Space Communications Protocol Specifications.  
URL:<http://www.scps.org/>.
- [8] NASA Reference Publication 1349. Twelve Year Planetary Ephemeris 1995 - 2006.  
URL:<http://sunearth.gsfc.nasa.gov/eclipse/TYPE/TYPE.html>.
- [9] Hooke A. Towards an Interplanetary Internet: A Proposed Strategy for Standardization. Space Ops 2002 Conference, October Houston, TX, USA, 2002.
- [10] L. Abusalah, A. Khokhar, and M. Guizani. A Survey of Secure Mobile Ad Hoc Routing Protocols. *IEEE Communications Surveys and Tutorials*, 10(4):78–93, 2008.
- [11] Peter Ahlf, Guy Fogleman, and David Tomko. Human Exploration and Development of Space (HEDS) Activities in Preparation for Future Exploration Missions. In *Space 2000: Seventh International Conference in Space*, volume 204, Albuquerque, New Mexico, USA, 2000.

- 
- [12] O.B. Akan, Jian Fang, and I.F. Akyildiz. TP-Planet a Reliable Transport Protocol for Interplanetary Internet. *IEEE Journal on Selected Areas in Communications*, 22(2):348–361, Feb 2004.
- [13] Ian F. Akyildiz, Özgür B. Akan, Chao Chen, Jian Fang, and Weilian Su. Inter-PlaNetary Internet: State-of-the-Art and Research Challenges. *Computer Networks*, 43:75–112, 2003.
- [14] Ian F. Akyildiz, Dario Pompili, and Tommaso Melodia. Underwater Acoustic Sensor Networks: Research Challenges. *Journal on Ad Hoc Networks*, 3(3):257 – 279, 2005.
- [15] R. Alena, B. Gilbaugh, B. Glass, and S.P. Braham. Communication System Architecture for Planetary Exploration. In *Proc. Aerospace Conference IEEE*, volume 3, pages 3/1075–3/1084 vol.3, Big Sky, Montana, USA, 2001.
- [16] D Antsos. Mars Technology Program Communications and Tracking Technologies for Mars exploration. In *Proc. IEEE Aerospace Conference*, Big Sky, Montana, USA, 2006.
- [17] W. Back and D.C. Lee. Expected File Delivery Time of Immediate NAK ARQ in CCSDS File Delivery Protocol. In *Proc. IEEE Aerospace Conference*, volume 3, Big Sky, Montana, USA, 2003.
- [18] M. Bajracharya, M.W. Maimone, and D. Helmick. Autonomy for Mars Rovers: Past, Present, and Future. *Computer*, 41(12):44–50, Dec 2008.
- [19] K. Bhasin and J.L. Hayden. Space Internet Architectures and Technologies for NASA enterprises. In *Proc. IEEE Aerospace Conference*, volume 2, page 2/931, Big Sky, Montana, USA, 2001.
- [20] I. Bisio, M. Marchese, and T. de Cola. Congestion Aware Routing Strategies for DTN-Based Interplanetary Networks. In *Proc. IEEE Global Telecommunications Conference, GLOBECOM'08*, pages 1–5, New Orleans, LA, 2008.
- [21] J.C. Breidenthal, C.D. Edwards, E. Greenberg, and G.K. Kazz, G.J.and Noreen. End-to-end Information System Concept for the Mars Telecommunications Orbiter. In *Proc. IEEE Aerospace Conference*, Big Sky, Montana, USA, 2006.

- [22] C. Caini, P. Cornice, R. Firrincieli, and D. Lacamera. A DTN approach to Satellite Communications. *IEEE Journal on Selected Areas in Communications*, 26(5):820–827, June 2008.
- [23] Cassini-Huygens. A Mission to Saturn, and Titan. URL:<http://saturn.jpl.nasa.gov/>.
- [24] CCSDS. Overview of Space Link Protocols. Green Book, URL:<http://standards.gsfc.nasa.gov/reviews/453-132/453-132.pdf>, Washington, D.C, April 2004.
- [25] CCSDS. File Delivery Protocol (CFDP) - Part 1: Introduction and Overview. Green Book. Issue 3, Washington, D.C, April 2007.
- [26] CCSDS. File Delivery Protocol (CFDP). Blue Book. Issue 4, Washington, D.C, January 2007.
- [27] CCSDS. Advanced Orbiting Systems(AOS) Space Data Link Protocol. Blue Book, Issue 2, Washington, D.C, July, 2006.
- [28] CCSDS. Proximity-1 Space Link Protocol - Data Link Layer. Blue Book, Issue 4, Washington, D.C, July 2006.
- [29] R J. Cesarone, R. C. Hastrup, D. J. Bell, D. T. Lyons, and K. G. Nelson. Architectural Design for a Mars Communications and Navigation Orbital Infrastructure. In *AAS/AIAA Astrodynamics Specialist Conference*, pages 99–300, Girdwood, Alaska, August 1999.
- [30] Chao Chen and Zesheng Chen. Routing Different Traffic in Deep Space Networks. In *Proc. International Workshop on Satellite and Space Communications IWSSC'07*, pages 237–241, Salzburg, Austria, 2007.
- [31] MARCHESE Mario DE COLA Tomaso, ERNST Harald. Performance Analysis of Ccsds File Delivery Protocol and Erasure Coding Techniques in Deep Space Environments. *Computer networks ISSN 1389-1286*, 51:4032–4049, 2007.
- [32] R.C. Durst, P.D. Feighery, and K.L. Scott. Why not use the Standard Internet Suite for the InterPlaNetary Internet? URL:<http://www.ipnsig.org/techinfo.htm>.
- [33] C.D. Edwards, B. Arnolda Jr.a, R. DePaulab, G. Kazza, C. Leea, and G. Noreena. Relay Communications Strategies for Mars exploration through 2020. *Space for*

- Inspiration of Humankind, Selected Proceedings of the 56th International Astronautical Federation Congress*, 59:310–318, Fukuoka, Japan, 2005.
- [34] K. Fall, W. Hong, and S. Madden. Custody Transfer for Reliable Delivery in Delay Tolerant Networks. Technical report, IRB-TR-03-030, July 2003.
- [35] Kevin Fall and Wei Hong. Custody Transfer for Reliable Delivery in Delay Tolerant Networks, 2003.
- [36] Jian Fang and I.F. Akyildiz. RCP-Planet: A Rate Control Protocol for InterPlanetary Internet. *International Journal of Satellite Communications and Networking*, 25(2):167–194, 2007.
- [37] S. Farrell and C. Jensen. A Flexible Interplanetary Internet. In *Proceedings of the 37th ESLAB Symposium ‘Tools and Technologies for Future Planetary Exploration’*, pages 87–94, Noordwijk, The Netherlands, 2004. ESA.
- [38] G. Goth. Delay-Tolerant Network Technologies Coming Together. *IEEE Distributed Systems Online*, 7(8):2–2, August 2006.
- [39] D.M. Hansen, M.K. Sue, C.M. Ho, M. Connally, T.K. Peng, R.J. Cesarone, and W. Home. Frequency Bands for Mars in-situ communications. In *Proc. IEEE Aerospace Conference*, volume 3, pages 3/1195–3/1208 vol.3, Big Sky, Montana, USA, 2001.
- [40] K. Harras and K. Almeroth. Transport Layer Issues in Delay Tolerant Mobile Networks. In *IFIP Networking*, Coimbra, Portugal, May 2006.
- [41] Patrik Jonsson Henrik Eriksso. Implementation and Analysis of the Bundling Protocol for Delay-Tolerant Network Architectures. Master’s thesis, Lule University of Technology, Sweden, 2005.
- [42] E. P. Jones and P. A. Ward. Routing Strategies for Delay-Tolerant Networks. *Submitted to ACM Computer Communication Review (CCR)*, 2006.
- [43] E.P.C. Jones, L. Li, J.K. Schmidtke, and P.A.S. Ward. Practical Routing in Delay-Tolerant Networks. *IEEE Transactions on Mobile Computing*, 6(8):943–959, August 2007.
- [44] T. Jonson, J. Pezeshki, V. Chao, K. Smith, and J. Fazio. Application of Delay Tolerant Networking (DTN) in Airborne Networks. In *IEEE Military Communications Conference, 2008. MILCOM’08*, pages 1–7, November 2008.



- [45] C. D. E Joseph I. Statman. Coding, Modulation, and Relays for Deep Space Communication Mars Rovers Case Study. In *IEEE 23rd Convention of Electrical Engineers in Israel*, Herzlia, Israel, 2004. Jet Propulsion Laboratory.
- [46] C.D. Edwards Jr.a and R. DePaulab. Key Telecommunications Technologies for Increasing Data Return for Future Mars Exploration. *Acta Astronautica*, page 131 138, 2007.
- [47] G. J. Kazz and E. Greenberg. Mars Communication Protocols. 19th AIAA International Communications Satellite Systems Conference, Toulouse, France, April 2001.
- [48] Fielhauer K.B. and Krupiarz C.J. MESSENGER Spacecraft In-flight Experiments: Science Return Improvement Techniques. *Proc. IEEE Aerospace Conference*, pages 1–8, March Big Sky, Montana, USA, 2008.
- [49] J. O. Kephart and D. M. Chess. The Vision of Autonomic Computing. *Computer*, 36:41–50, January 2003.
- [50] Fall Kevin. A Delay-Tolerant Network Architecture for Challenged Internets. In *SIGCOMM '03: Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 27–34, New York, NY, USA, 2003. ACM.
- [51] J. Lacan and E. Lochin. Rethinking Reliability for Long-Delay Networks. In *IEEE International Workshop on Satellite and Space Communications, IWSSC'08.*, pages 90–94, October Toulouse, France, 2008.
- [52] C. Lee and K Cheung. Power, Data Latency, and Radio Frequency Interference issues in Mars Relay Network Scheduling. In *Proc. IEEE Aerospace Conference*, Big Sky, Montana, USA, 2003.
- [53] D.C. Lee and W. Baek. Expected File-Delivery Time of deferred NAK ARQ in CCSDS File-Delivery Protocol. *IEEE transactions on communications*, 52(8):1408–1416, 2004.
- [54] Jrmie Leguay. *Heterogeneity and Routing in Delay Tolerant Networks*. PhD thesis, Pierre & Marie Curie University, France, 2007.

- [55] Chien-Shiu Lin, Wei-Shyh Chang, Ling-Jyh Chen, and Cheng-Fu Chou. Performance Study of Routing Schemes in Delay Tolerant Networks. In *22nd International Conference on Advanced Information Networking and Applications - Workshops, AINAW'08*, pages 1702–1707, March 2008.
- [56] Anders Lindgren, Avri Doria, and Olov Schelén. Probabilistic Routing in Intermittently Connected Networks. *SIGMOBILE Mob. Computer and Communication Review*, 7(3):19–20, 2003.
- [57] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow. TCP Selective Acknowledgment Options. URL:<http://www.ietf.org/rfc/rfc2018.txt>, 1996.
- [58] P. McDonald, D. Geraghty, I. Humphreys, S. Farrell, and V. Cahill. Sensor Network with Delay Tolerance (SeNDT). In *Proceedings of 16th International Conference on Computer Communications and Networks, ICCCN'07*, pages 1333–1338, August Honolulu, Hawaii USA, 2007.
- [59] F.S. Muhammad, L. Franck, and S. Farrell. Transmission Protocols for Challenging Networks: LTP and LTP-T. In *International Workshop on Satellite and Space Communications, IWSSC '07*, pages 145–149, September Salzburg, Austria, 2007.
- [60] M. Musolesi, S. Hailes, and C. Mascolo. Adaptive Routing for Intermittently Connected Mobile Ad hoc Networks. In *Sixth IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks, WoWMoM'05*, pages 183–189, June Taormina - Giardini Naxos, 2005.
- [61] NASA. Formulation of Modulation Recommendations for Future NASA Space Communications. URL:<http://coding.jpl.nasa.gov/hamkins/>.
- [62] The Network Simulator NS-2. URL:<http://www.isi.edu/nsnam/ns/>.
- [63] Krzysztof Pawlikowski. Do Not Trust All Simulation Studies of Telecommunication Networks. In *Proc. International Conference on Information Networking, ICOIN'03*, pages 899–908, GJeju Island, Korea, 2003.
- [64] C. Perkins, O. Hodson, and V. Hardman. A Survey of Packet Loss Recovery Techniques for Streaming Audio. *IEEE journal on Networks*, 12(5):40–48, 1998.
- [65] M. Prathaban and J. Kohlenberg. Resource Aware Routing In Interplanetary Mars Surface Network. In *The 2nd international conference on advanced computer theory and engineering, ICACTE'09*, Cairo, Egypt, September 2009.

- [66] M. Prathaban and J. Kohlenberg. Dynamic NAK Timer Algorithm to Improve Delivery Latency of CCSDS File Delivery Protocol in Deferred NAK Mode. In *Fifth Advanced International Conference on Telecommunications, AICT'09*, pages 304–309, May Venice, Italy, 2009.
- [67] M. Prathaban, E. Simu, and J. Kohlenberg. An Antenna Selection Algorithm for Ars Exploration Rover to Increase Data Return With Minimum Delay. In *IEEE Aerospace conference*, pages 1–10, Big Sky, Montana, USA, March 2009.
- [68] Mahendiran Prathaban and Josephine Kohlenberg. Characteristics of TCP in IPN Environment. 4th International Working Conference, Performance Modelling and Evaluation of HETerogeneous NETworkS, HET-NET'06, Ilkley, West Yorkshire, U.K, September 2006.
- [69] Mahendiran Prathaban and Joséphine Kohlenberg. Buffer Management Policy for Mars Intelligent Proximity Network. In *Asia International Conference on Modelling and Simulation, AMS'08*, pages 309–314, Kuala Lumpur, Malaysia, 2008.
- [70] V.P. Pribylov. Estimation of Efficiency of ARQ Data Transmission Protocols in Energy-Constrained Long Signal Propagation Delay Channels. In *Proc. 4th Annual 2003 Siberian Russian Workshop on Electron Devices and Materials*, pages 135–136, 2003.
- [71] CMU Monarch Project. The CMU Monarch Project's Wireless and Mobility Extensions to NS2. URL:<ftp://monarch.cs.cmu.edu/pub/monarch/wireless-sim/ns-cmu>, 1999.
- [72] Durst R. Space Communications Protocol Standards Overview. Presentation Slides URL:<http://www.scps.org/>, April 1998.
- [73] Jonathan R.Agre, Loren P.Clare, Tsun-Yee Yan, Kul Bhasin, and Jeff Hayden. Advanced Communication and Networking Technologies for Mars Exploration. In *19th International Communications Satellite Systems Conference and Exhibit*, Toulouse, France, 2001. AIAA.
- [74] Ram Ramanathan, Prithwish Basu, and Rajesh Krishnan. Towards a Formalism for Routing in Challenged Networks. In *CHANTS '07: Proceedings of the second ACM workshop on Challenged networks*, pages 3–10, New York, NY, USA, 2007. ACM.

- [75] RFC2396. Uniform Resource Identifiers (URI): Generic Syntax. URL:<http://www.ietf.org/rfc/rfc2396.txt>, August 1998.
- [76] RFC959. File Transfer Protocol - FTP. URL:<http://tools.ietf.org/html/rfc959>.
- [77] Radu D. Rugescu, Josephine Kohlenberg, and Mahendiran Prathaban. Space Imaging Infrastructure Development of PUBSAT and NERVA Orbital Vehicle. In *ICSNC '08: Proceedings of the 2008 Third International Conference on Systems and Networks Communications*, pages 298–303, Washington, DC, USA, 2008. IEEE Computer Society.
- [78] Burleigh S. Operating CFDP in the Interplanetary Internet. In *Space Ops Conference*, Houston, TX, USA, 2002.
- [79] Burleigh S., Hooke A., Torgerson L., Fall K., Cerf V., Durst B., Scott K., and Weiss H. Delay-Tolerant Networking: an approach to Interplanetary Internet. *IEEE Communications Magazine*, 41(6):128–136, June 2003.
- [80] Burleigh S. and et. al. Licklider Transmission Protocol. draft-irtf-dtnrg-ltp-l3.txt, July 2005.
- [81] Amir Said and William A. Pearlman. A New Fast and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees. *IEEE Transactions on Circuits and Systems for Video Technology*, 6:243–250, 1996.
- [82] K. Scott and S. Burleigh. Bundle Protocol Specification. draft-irtf-dtnrg-bundle-spec-03.txt, July 2005.
- [83] SCPS. Space Communications Protocol Standards: Rationale and Requirements and Application Notes. Draft Green Book, URL:<http://www.scps.org/>, August 1998.
- [84] SCPS. Space Communications Protocol Specification - File Protocol (SCPS-FP). Blue Book, Issue 1, Washington, D.C, May 1999.
- [85] SCPS. Space Communications Protocol Specification - Network Protocol (SCPS-NP). Blue Book, Issue 1, Washington, D.C, May 1999.
- [86] SCPS. Space Communications Protocol Specification - Security Protocol (SCPS-SP). Blue Book, Issue 1, Washington, D.C, May 1999.

- [87] SCPS. Space Communications Protocol Specification - Transport Protocol (SCPS-TP). Blue Book, Issue 2, Washington, D.C, October 2006.
- [88] J. M. Shapiro. Embedded Image Coding Using Zerotrees of Wavelet Coefficients. *IEEE Transactions on Signal Processing*, 41(12):3445–3462, 1993.
- [89] Raphael R. Some and Daniel S. Katz. NASA Advances Robotic Space Exploration. *Computer*, 36(1):52–61, 2003.
- [90] W. Richard Stevens. *TCP/IP Illustrated, Volume 1: The Protocols*. Addison-Wesley Professional, us ed edition, January 1994.
- [91] Jain Sushant, Fall Kevin, and Patra Rabin. Routing in a Delay Tolerant Network. In *SIGCOMM '04: Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 145–158. ACM, 2004.
- [92] S. Symington, S. Farrell, H. Weiss, and P. Lovell. Bundle Security Protocol Specification. draft-irtf-dtnrg-bundle-security-05, February 24, 2008.
- [93] Jim Taylor, Andre Makovsky, Andrea Barbieri, Ramona Tung, Polly Estabrook, and A. Gail Thomas. Mars Exploration Rover Telecommunications. In *Deep Space Communications and Navigation Systems*. Jet Propulsion Laboratory, 2005.
- [94] Cerf. V. and et al. Delay-Tolerant Network Architecture. IETF RFC 4838 and informational, April 2007.
- [95] A. Vahdat and D. Becker. Epidemic Routing for Partially Connected Ad Hoc Networks, 2000.
- [96] Anton Vanderwyst, Jim Beyer, Christian Passow, Anna Paulson, and Chad Rowland. Power Generation and Energy Usage in a Pressurized Mars Rover. In *Proc. The Martian Expedition Planning Symposium of the British Interplanetary Society*, London, England, 2003.
- [97] R. Wang, S. Pradhanang, P. Manandhar, and Qian Zhang. Performance Evaluation of CFDP in Deferred NAK Mode over Point-to-Point LEO-Satellite Links. In *Proc. IEEE Wireless Communications and Networking Conference, WCNC'07*, pages 3709–3713, Hong Kong, 2007.

- [98] Ruhai Wang, D.L. Rudraraju, P.K.V. Rapet, Youyun Xu, and Xinbing Wang. Performance Evaluation of CCSDS File Delivery Protocol (CFDP) in Deferred NAK mode over Geostationary Earth Orbit (GEO)-Satellite Links. In *Proc. IEEE International Conference on Communications ICC'07*, pages 4421–4425, Glasgow, Scotland, 2007.
- [99] Ruhai Wang, B.L. Shrestha, Xuan Wu, E. Tade, Tiaotiao Wang, and Xinbing Wang. Experimental Investigation of CCSDS File Delivery Protocol (CFDP) over Cislunar Communication Links with Intermittent Connectivity. In *Proc. IEEE International Conference on Communications ICC'08*, pages 1910–1914, Beijing, China, 2008.
- [100] Yu Wang and Hongyi Wu. Delay/Fault-Tolerant Mobile Sensor Network (DFT-MSN): A New Paradigm for Pervasive Information Gathering. *IEEE Transactions on Mobile Computing*, 6(9):1021–1034, 2007.
- [101] William J. Weber, Robert J. Cesarone, Douglas S. Abraham, Peter E. Doms, Richard J. Doyle, Charles D. Edwards, Adrian J. Hooke, James R. Lesh, and Richard B. Miller. Transforming the Deep Space Network into the Interplanetary Network. *Computer Networks - Acta Astronautica*, 58:411–421, 2006.
- [102] E.J. Wyatt, T.A. Ely, M.A. Klimesh, and C.J. Krupiarz. Telecommunications IT and navigation for future Mars exploration missions. In *Proc. IEEE Aerospace Conference*, page 12 pp, Big Sky, Montana, USA, 2006.
- [103] Jay Wyatt, Scott Burleigh, Ross Jones, Leigh Torgerson, and Steve Wissler. Disruption Tolerant Networking Flight Validation Experiment on NASA's EPOXI Mission. *International Conference on Advances in Satellite and Space Communications*, 0:187–196, 2009.
- [104] Seung-Keun Yoon, Z.J. Haas, and J.H. Kim. Tradeoff Between Energy Consumption and Lifetime in Delay-Tolerant mobile Network. In *IEEE Military Communications Conference, 2008. MILCOM'08*, pages 1–7, November 2008.
- [105] Zhensheng Zhang. Routing in Intermittently Connected Mobile Ad hoc Networks and Delay Tolerant Networks: Overview and Challenges. *IEEE Communications Surveys and Tutorials*, 8(1):24–37, 2006.
- [106] Wenrui Zhao. *Routing and Network Design in Delay Tolerant Networks*. PhD

thesis, Georgia Institute of Technology, Atlanta, GA, USA, 2006. Adviser-Ammar, Mostafa and Adviser-Zegura, Ellen.

- [107] H. Zimmermann. OSI reference model—The ISO model of architecture for open systems interconnection. pages 2–9, 1988.