



# Topics in Delay Tolerant Networks (DTNs) : reliable transports, estimation and tracking

Arshad Ali

## ► To cite this version:

Arshad Ali. Topics in Delay Tolerant Networks (DTNs) : reliable transports, estimation and tracking. Economics and Finance. Institut National des Télécommunications, 2012. English. NNT : 2012TELE0038 . tel-00790766

**HAL Id: tel-00790766**

**<https://theses.hal.science/tel-00790766v1>**

Submitted on 21 Feb 2013

**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.



**THESE DE DOCTORAT CONJOINT**  
**TELECOM SUDPARIS et L'UNIVERSITE PIERRE ET MARIE CURIE**

**Spécialité:**

Télécommunications

**Ecole Doctorale EDITE:**

Informatique, Télécommunications et Electronique de Paris

**Présentée par**

**Arshad ALI**

**Pour obtenir le grade de**

**DOCTEUR DE TELECOM SUDPARIS**

**Transport Fiable, Estimation et Poursuite dans les Réseaux**  
**Delay Tolerant Networks (DTNs)**

Soutenue le 12 novembre, 2012 devant le jury composé de :

<b>Rapporteurs</b>	<b>:</b>	<b>Konstantin AVRACHENKOV</b>	<b>Inria Sophia Antipolis, France</b>
		<b>Tamer BASAR</b>	<b>Université d'Illinois, USA</b>
<b>Examineurs</b>	<b>:</b>	<b>Guy PUJOLLE</b>	<b>Université Pierre et Marie Curie, France</b>
		<b>Lucile SASSATELLI</b>	<b>Université Nice Sophia Antipolis, France</b>
<b>Directeurs</b>	<b>:</b>	<b>Eitan ALTMAN</b>	<b>Inria Sophia Antipolis, France</b>
		<b>Tijani CHAHED</b>	<b>Telecom SudParis, France</b>

**Thèse numéro: 2012TELE0038**



**JOINT DOCTORATE OF  
TELECOM SUDPARIS AND PIERRE & MARIE CURIE UNIVERSITY**

**Speciality:**

Telecommunications

**Doctoral School EDITE:**

Computer Science, Telecommunications and Electronics (Paris)

**presented by**

**Arshad ALI**

**To obtain the Degree of**

**DOCTORATE OF TELECOM SUDPARIS**

**Topics in Delay Tolerant Networks (DTNs):  
Reliable Transport, Estimation and Tracking**

Defended on November 12, 2012 in front of the jury composed of:

<b>Reviewers</b>	<b>:</b>	<b>Konstantin AVRACHENKOV</b>	<b>Inria Sophia Antipolis, France</b>
		<b>Tamer BASAR</b>	<b>University of Illinois, USA</b>
<b>Examiners</b>	<b>:</b>	<b>Guy PUJOLLE</b>	<b>Université Pierre et Marie Curie, France</b>
		<b>Lucile SASSATELLI</b>	<b>Université Nice Sophia Antipolis, France</b>
<b>Directors</b>	<b>:</b>	<b>Eitan ALTMAN</b>	<b>Inria Sophia Antipolis, France</b>
		<b>Tijani CHAHED</b>	<b>Telecom SudParis, France</b>

**Thesis number: 2012TELE0038**

*To My Parents*  
*My Wife*  
and  
*My Daughters*



# Acknowledgements

I can honestly state that my journey for obtaining a doctorate degree would not have been possible without the support of many people.

First of all, I thank to Prof. Tijani Chahed for his excellent guidance, support and dedication during all the time I spent working on this PhD thesis. It is my great fortune to carry out my PhD studies under his guidance. His vision, direction, and significant feedback was the most beneficial to my doctoral research. I am grateful to him for everything, particularly, for granting me the freedom of developing my ideas. His suggestions, recommendations and ability to manage several diverse things together was, indeed, remarkable. I would like to take this opportunity to express my heartiest gratitude to Prof. Tijani Chahed for being such a great advisor.

I would like to say special thanks to my co-adviser, honorable Dr. Eitan Altman, whose motivation and ideas, throughout my thesis research, were the real driving force towards completion of my PhD thesis. He has highly inspired me for his ideas, and whose guidance helped me a lot towards technical quality of the thesis.

I am also grateful to honorable Prof. Tamer Baser, Prof. Konstantin Avrachenkov, Prof. Guy Pujolle and Prof. Lucile Sassatelli for being members of my thesis evaluation committee and providing their valuable comments through forming my thesis work.

I would like to express my sincere gratitude to Prof. Lucile Sassatelli for her technical support, pleasant and precise guidance at every step of my thesis. I really learned a lot of things by working with her and benefited from her experience as well as her excellent research and technical skills.

I would like to thank my friend Manoj for his continuous technical support and encouragement throughout my thesis. Especially, his friendly company and guidance made my study more enjoyable. I liked the long discussions with him a lot. With his profound knowledge, he supported me throughout my PhD study. I am highly grateful for all the suggestions and advises he made to improve the quality of this document.

I want to thank all my friends and doctoral colleagues whose friendly company and support, during my PhD work, was a source of great pleasure.

I am grateful to Higher Education Commission (HEC) of Pakistan for providing generous grants to have pursued my PhD study in France. I am also grateful to Institute Telecom SudParis for providing excellent research facilities. I would like to thank administrative and technical staff members of Telecom SudParis who have been kind enough to help in their respective roles.

Finally, my deep gratitude goes to my parents, my wife and lovely daughters. I am deeply grateful to my parents for their prayers and standing by me in everything I have done

and giving me whatever they can. I thank to my brothers and sisters for their continuous support, encouragement and love. Last but certainly not the least, I am indebted to my wife, for her love, patience and generous support through all tough times. Thanks to my lovely and cute Ayesha, Maira and Hafsa who are a great source of peace in my life. I dedicate this thesis to them. Thanks a lot.

# Abstract

Mobile Ad hoc NETWORKs (MANETs) aim at making communication between mobile nodes feasible without any infrastructure support. If the spatial density of mobile nodes in a MANET is low, then an end-to-end path between a source and a destination almost never exists; two mobile nodes can communicate only when they come within the radio range of each other. During the last few years, motivation for development of MANETs has increased with the entry of intelligent devices with short range wireless communication methods. Sparse MANETs fall into the class of Delay Tolerant Networks (DTNs) which are intermittently connected networks and where there is no contemporaneous end-to-end path at any given time. The major or core research in DTNs addresses routing aspects while relatively fewer works exist on reliable transport; and when they do, they are mainly oriented towards deep space communication.

In this thesis, we first, propose a new reliable transport scheme for DTNs based on the use of ACKnowledgments (ACKs) as well as random linear coding. We model the evolution of the network under our scheme using a fluid-limit approach. We obtain mean file transfer times based on certain optimal parameters obtained through differential evolution approach. We account for the buffer expiry time-out, quantify its impact on the optimal values of our protocol parameters and also demonstrate the adaptability of our optimal procedure to variations in expiry time-out.

Secondly, contact opportunities in DTNs are quite infrequent. This observation motivates the need to take the maximum benefit of these rare contacts. We thus propose and study a novel enhanced ACK approach called Global Selective ACKnowledgment (G-SACK) to improve reliable transport for DTNs covering both unicast and multicast flows. We make use of random linear coding at relays so that packets can reach the destination faster. We obtain reliability based on the use of so-called G-SACKs that can potentially contain global information on receipt of packets at all destinations. We obtain significant improvement through G-SACKs and coding at relays.

Finally, we tackle the problem of estimating file-spread in DTNs with direct delivery and epidemic routing. We estimate and track the degree of spread of a message/file in the network. We provide analytical basis to our estimation framework alongwith insights validated with simulations. We observe that the deterministic fluid model can indeed be a good predictor with a large number of nodes. Moreover, we use Kalman filter and Minimum-Mean-Squared-Error (MMSE) to track the spreading process and find that Kalman filter provides more accurate results as compared to MMSE.

**Keywords:** Delay Tolerant Networks, Transport, Linear Network Coding, Acknowledgements, Fluid and Diffusion Approximation, Kalman Filtering





# Résumé

Les réseaux mobiles Ad hoc (MANETs) visent à permettre à des nœuds mobiles de communiquer sans aucun support d'infrastructure. Si la densité spatiale des nœuds mobiles dans un MANET est faible, un chemin continu entre une source et une destination n'existe presque jamais ; deux nœuds mobiles peuvent communiquer uniquement quand ils entrent dans la zone de couverture radio les uns des autres. Au cours des dernières années, la motivation pour le développement des MANETs a augmenté avec l'entrée en jeu des dispositifs intelligents qui utilisent les communications dédiées à courte portée. Les MANETs dispersés entrent dans la catégorie des réseaux tolérants aux délais (DTN), qui sont des réseaux connectés par intermittence et où il n'y a aucun chemin de bout-en-bout persistant à n'importe quel temps donné. La recherche majeure dans les DTNs traite des aspects de routage tandis que les travaux sur le transport sont relativement faibles ; ceux qui existent sont principalement orientés vers la communication dans l'espace.

Dans cette thèse, nous proposons, d'abord, un nouveau protocole de transport fiable pour les DTNs basé sur l'utilisation d'accusés de réception (ACK) ainsi que le codage linéaire aléatoire. Nous modélisons l'évolution du réseau conformément à notre plan en utilisant l'approche fluide. Nous obtenons le temps de transfert d'un fichier en fonction de certains paramètres optimaux obtenus par l'approche d'évolution différentielle. Nous tenons compte de l'expiration du délai d'attente dans le tampon, et quantifions son impact sur les valeurs optimales de nos paramètres de protocole. Nous démontrons l'adaptabilité de notre procédure d'optimisation aux variations de l'expiration du délai d'attente.

Deuxièmement, les occasions de contact dans les DTNs sont peu fréquentes. Cette observation motive le besoin de prendre un avantage maximal de ces contacts rares. Nous proposons ainsi et étudions un nouveau mécanisme d'ACK augmenté, que nous appelons Global Sélective ACKnowledgement (G-SACK), pour améliorer le transport fiable pour les DTNs, pour les cas unicast et multicast. Nous nous servons du codage linéaire aléatoire aux relais pour que les paquets puissent atteindre la destination plus rapidement. Nous obtenons la fiabilité basée sur l'utilisation des G-SACKs qui peuvent potentiellement contenir des informations globales sur les paquets reçus à toutes les destinations. Nous obtenons une amélioration significative par les G-SACKs et le codage dans les relais.

Enfin, nous abordons le problème de l'estimation de propagation des fichiers dans les DTNs avec livraison directe et le routage épidémique. Nous estimons et suivons le degré de propagation d'un message/fichier dans le réseau. Nous fournissons la base analytique à notre cadre d'évaluation avec des aperçus validés en se basant sur des simulations. Nous observons que le modèle déterministe fluide peut en effet être un bon prédicateur dans le cas d'un grand nombre de nœuds. En plus, nous utilisons le filtre de Kalman et Minimum-Mean-Squared Error (MMSE) pour suivre le processus de propagation et trouvons que le filtre de Kalman fournit des résultats plus précis par rapport à MMSE.

**mots clés :** Réseaux Tolérants aux Délais (DTNs), Transport, codage linéaire aléatoire, Accusés de réception (ACK), Approximation de Fluid et Diffusion , filtre de Kalman



# Notations

<b>MANETs</b>	Mobile Ad hoc NETworks
<b>ACK(s)</b>	ACKnowledgment(s)
<b>DTN(s)</b>	Delay/Distribution Tolerant Network(s)
<b>VANETs</b>	Vehicular Ad hoc NETworks
<b>IPNs</b>	Inter-Planetary Networks
<b>PSNs</b>	Pocket Switched Networks
<b>ANs</b>	Airborn Networks
<b>MSNs</b>	Mobile Social Networks
<b>UWNs</b>	Under Water Networks
<b>UDP</b>	User Datagram Protocol
<b>TCP</b>	Transmission Control Protocol
<b>ARQ</b>	Automatic Repeat-reQuest
<b>AODV</b>	Ad hoc On-Demand Distance Vector
<b>DSR</b>	Dynamic Source Routing
<b>RLC(s)</b>	Random Linear Combination(s)
<b>DoF</b>	Degree of Freedom
<b>G-SACK</b>	Global Selective ACKnowledgment
<b>MMSE</b>	Minimum-Mean-Squared-Error
<b>DTNRG</b>	Delay Tolerant Network Research Group
<b>IRTF</b>	Internet Research Task Force
<b>BP</b>	Bundle Protocol
<b>RFC</b>	Request For Comments
<b>SCPS-TP</b>	Space Communication Protocol standards- Transport Protocol
<b>CCSDS</b>	Consultative Committee for Space Data Systems
<b>RTT</b>	Round Trip Time
<b>SNACK(s)</b>	Selective Negative ACKnowledgement(s)
<b>ICMP</b>	Internet Control Message Protocol
<b>CFDP</b>	CCSDS File Delivery Protocol
<b>LTP</b>	Licklider Transmission Protocol
<b>DS-TP</b>	Deep-Space Transport Protocol
<b>NACK</b>	Negative ACKnowledgment
<b>DAR</b>	Double Automatic Retransmission
<b>TP-Planet</b>	Transport Protocol for InterPlanetary Internet
<b>AIMD</b>	Additive Increase Multiplicative Decrease
<b>DTTP</b>	Delay-Tolerant Transport Protocol

<b>FEC</b>	Forward Error Correction
<b>MDTN</b>	Mobile DTN
<b>ODEs</b>	Ordinary Differential Equations
<b>CDF(s)</b>	Cumulative Distribution Function(s)
<b>SACK</b>	Selective ACKnowledgment
<b>G-ACK</b>	Global ACKnowledgment
<b>BSs</b>	Base Stations
<b>APs</b>	Access Points
<b>CTMC</b>	Continuous Time Markov Chain
<b>FCLT</b>	Functional Central Limit Theorem
<b>CMT</b>	Continuous Mapping Theorem
<b>SLLN</b>	Strong Law of Large Numbers
<b>CLT</b>	Central Limit Theorem

# Table of contents

<b>1</b>	<b>Introduction</b>	<b>21</b>
1.1	Motivation and Approach . . . . .	22
1.2	Contributions of the Thesis . . . . .	25
1.3	Organization of the Thesis . . . . .	26
<b>2</b>	<b>State of the Art</b>	<b>29</b>
2.1	Literature on Transport Protocols . . . . .	30
2.1.1	Transport Proposals for deep-space Communication . . . . .	30
2.1.2	Transport Proposals for Terrestrial DTNs . . . . .	35
2.2	Literature on Routing . . . . .	37
2.3	Literature on Network Coding . . . . .	40
<b>3</b>	<b>A New Reliable Transport Scheme</b>	<b>45</b>
3.1	Related Work . . . . .	47
3.2	Network Setting . . . . .	48
3.3	Our Proposed Reliable Transport Scheme . . . . .	49
3.3.1	Algorithm . . . . .	50
3.3.2	Random Linear Coding (RLC) and ACK Replication . . . . .	51
3.3.3	Implementation Issues . . . . .	53
3.4	Analytical Modeling and Optimization . . . . .	54
3.4.1	Modeling the Network Dynamics in One Cycle . . . . .	54
3.4.2	Combining the Cycles Together . . . . .	58
3.4.3	Optimization Procedure . . . . .	59
3.5	Approximation 1 . . . . .	60
3.6	Approximation 2 . . . . .	60
3.7	Validation of the Analytical Model and Performance ... . . . .	63
3.7.1	Proposed Model . . . . .	63
3.7.2	Approximation 1 . . . . .	65
3.7.3	Approximation 2 . . . . .	65
3.8	Conclusion of the Chapter . . . . .	68
3.9	Appendices . . . . .	68

3.9.1	Derivation of $f_{S_{i-1}, \dots, S_0}(s_{i-1}, \dots, s_0)$	68
3.9.2	Derivation of $E_{Direct}[T_{i \rightarrow 0}]$	70
<b>4</b>	<b>Enhancements to Reliable Transport</b>	<b>71</b>
4.1	Network Setting	72
4.2	Fundamental Limitations and their Consequences	73
4.3	Our Proposed Enhancements	75
4.4	Performance Evaluation Methodology	79
4.4.1	Simulation Settings	79
4.4.2	Performance Metrics	79
4.5	Improving the Return Path	84
4.5.1	Benefits of Selective ACKs (SACKs) over ACKs	84
4.5.2	Benefits of Global-ACK (G-ACK) over ACK	86
4.5.3	Benefits of G-SACK over SACK with Single Destination	88
4.5.4	Benefits of G-SACK over SACK with Multiple Destinations	90
4.6	Improving the Forward Path	92
4.7	Benefits of overall Proposed Scheme over Basic Scheme	96
4.8	Conclusion of the Chapter	100
<b>5</b>	<b>Tracking Message-Spread in DTNs</b>	<b>101</b>
5.1	Network Model	103
5.2	Characterization of the Process and the Observation	106
5.3	Objective and Approach	107
5.4	Fluid and Diffusion Approximations	108
5.4.1	Fluid Approximation for the Spreading Process	109
5.4.2	Diffusion Approximation for the Spreading Process	111
5.4.3	Diffusion Approximation for the Observation	112
5.5	Discrete Time Kalman Filtering	114
5.6	Linear Minimum Mean Squared Error Estimator	118
5.6.1	Derivation of the Means and (Co)variances	119
5.7	Performance of Analytical Prediction and Estimation	121
5.7.1	Direct Delivery	121
5.7.2	Epidemic Routing	125
5.8	Conclusion of the Chapter	125
5.9	Appendices	127
5.9.1	Fluid and Diffusion Models	127
5.9.2	Brownian Motion	128
5.9.3	Density-Dependent Markov Chains	129
5.9.4	Continuous Mapping Approach	132
<b>6</b>	<b>Conclusions and Perspectives</b>	<b>135</b>
6.1	Conclusions	135
6.2	Perspectives	136

<i>TABLE OF CONTENTS</i>	15
<b>Bibliography</b>	<b>139</b>
<b>A Thesis Publications</b>	<b>153</b>
<b>B Version Française</b>	<b>155</b>
B.1 Motivation pour la Thèse . . . . .	155
B.2 Notre Approche . . . . .	157
B.3 Les Contributions de la Thèse . . . . .	157
B.4 Aperçu de la thèse . . . . .	159
B.5 1ère Partie du travail: Un Nouveau Schéma de Transport Fiable . . . . .	159
B.6 2ème Partie du travail: Améliorations au Transport Fiable . . . . .	164
B.7 3ème Partie du travail: Poursuite de Diffusion de Message dans les DTNs . . . . .	167
B.8 Conclusions et Perspectives . . . . .	169





# List of Figures

3.1	Model: $T_1$ against $\tau_e$ for Scheme 1 when missing DoF = 1 . . . . .	64
3.2	Model: $T_1$ against $\tau_e$ for Scheme 2 and Scheme 3 when missing DoF = 1 . .	64
3.3	Approximation 1: $T_1$ against $\tau_e$ for Scheme 3 when missing DoF = 1 . . . .	66
3.4	Approximation 1: $T_2$ against $\tau_e$ for Scheme 3 when missing DoF = 2 . . . .	66
3.5	Approximation 1: $T_3$ against $\tau_e$ for Scheme 3 when missing DoF = 3 . . . .	66
3.6	Approximation 1: $T_4$ against $\tau_e$ for Scheme 3 when missing DoF = 4 . . . .	66
3.7	Approximation 1: $T_M = T_5$ against $\tau_e$ for Scheme 3 when missing DoF = 5	66
3.8	Approximation 2: $T_1$ against $\tau_e$ for Scheme 3 when missing DoF = 1 . . . .	67
3.9	Approximation 2: $T_2$ against $\tau_e$ for Scheme 3 when missing DoF = 2 . . . .	67
3.10	Approximation 2: $T_3$ against $\tau_e$ for Scheme 3 when missing DoF = 3 . . . .	67
3.11	Approximation 2: $T_4$ against $\tau_e$ for Scheme 3 when missing DoF = 4 . . . .	67
3.12	Approximation 2: $T_M = T_5$ against $\tau_e$ for Scheme 3 when missing DoF = 5	67
4.1	The network of $N$ relays (shown as the cloud) . . . . .	73
4.2	Three unicast flows. . . . .	73
4.3	Many Sources, Single Destination: Topology 1 . . . . .	80
4.4	Single Source, Many Destinations: Topology 2 . . . . .	80
4.5	Mixed flows: Topology 4 . . . . .	80
4.6	Three multicast flows: Topology 5 . . . . .	80
4.7	SACK over ACK: pairwise delay CDFs . . . . .	85
4.8	SACK over ACK: network-wide maximum delay CDFs . . . . .	85
4.9	G-ACK over ACK: pairwise delay CDFs . . . . .	87
4.10	G-ACK over ACK: network-wide maximum delay CDFs . . . . .	87
4.11	G-SACK over SACK (Single Destination): pairwise delay CDFs . . . . .	89
4.12	G-SACK over SACK (Single Dest.): network-wide max. delay CDFs . . . .	89
4.13	G-SACK over SACK (Multiple Destinations): pairwise delay CDFs . . . .	91
4.14	G-SACK over SACK (Multiple Dests.): network-wide max. delay CDFs . .	91
4.15	Benefits of coding: pairwise forward delay CDFs (3 flows) . . . . .	94
4.16	Benefits of coding: network-wide average forward delay CDFs(3 flows) . .	94
4.17	Benefits of coding: network-wide maximum forward delay CDFs (3 flows) .	94
4.18	Benefits of coding: pairwise forward delay CDFs(10 flows) . . . . .	94

4.19	Benefits of coding: network-wide average forward delay CDFs (10 flows) . .	94
4.20	Benefits of coding: network-wide maximum forward delay CDFs (10 flows) .	94
4.21	Proposed over Basic: pairwise delay CDFs for Topology 3 . . . . .	97
4.22	Proposed over Basic: pairwise delay CDFs for Topology 4 . . . . .	97
4.23	Proposed over Basic: pairwise delay CDFs for Topology 5 . . . . .	97
4.24	Proposed over Basic: network-wide maximum delay CDFs for Topology 3 .	97
4.25	Proposed over Basic: network-wide maximum delay CDFs for Topology 4 .	97
4.26	Proposed over Basic: network-wide maximum delay CDFs for Topology 5 .	97
4.27	Proposed over Basic: network-wide mean delays . . . . .	99
4.28	Proposed over Basic: network-wide success probabilities . . . . .	99
5.1	A DTN consisting of $S_0$ sources, $H_0$ observers and $N_0$ users. . . . .	104
5.2	MMSE and Kalman filter Estimation of process: Scenario 1 . . . . .	122
5.3	MMSE and Kalman filter Estimation of process fluctuations: Scenario 1 . .	122
5.4	MMSE and Kalman filter Estimation of process: Scenario 2 . . . . .	122
5.5	MMSE and Kalman filter Estimation of process fluctuations: Scenario 2 . .	122
5.6	MMSE and Kalman filter Estimation of process: Scenario 3 . . . . .	122
5.7	MMSE and Kalman filter Estimation of process fluctuations: Scenario 3 . .	122
5.8	MMSE and Kalman filter Estimation of process: Scenario 3 . . . . .	123
5.9	MMSE and Kalman filter Estimation of process fluctuations: Scenario 3 . .	123
5.10	$V_{xy}(t)$ as a function of time . . . . .	123
5.11	$V_{yy}(t)$ as a function of time . . . . .	123
5.12	MMSE and Kalman filter Estimation of process: Scenario 4 . . . . .	124
5.13	MMSE and Kalman filter Estimation of process fluctuations: Scenario 4 . .	124
5.14	Kalman filter Estimation of process: Scenario 1' . . . . .	126
5.15	MMSE and Kalman filter Estimation of process fluctuations: Scenario 1' . .	126
5.16	Kalman filter Estimation of process: Scenario 2' . . . . .	126
5.17	MMSE and Kalman filter Estimation of process fluctuations: Scenario 2' . .	126
5.18	Kalman filter Estimation of process: Scenario 3' . . . . .	126
5.19	MMSE and Kalman filter Estimation of process fluctuations: Scenario 3' . .	126
5.20	Kalman filter Estimation of process: Scenario 3' . . . . .	127
5.21	Kalman filter Estimation of process fluctuations: Scenario 4' . . . . .	127
B.1	Modél: $T_M$ contre $\tau_e$ pour Schéma 1 quand DoF manquant = 1 . . . . .	163
B.2	Le réseau de $N$ relais (représentée par le nuage) . . . . .	164
B.3	Three unicast flows. . . . .	164
B.4	Proposé sur Base: délai CDFs (paires) . . . . .	168
B.5	Proposé sur Base: maximale délai CDFs (l'ensemble du réseau) . . . . .	168
B.6	Un DTN composé de $S_0$ sources, $H_0$ observateurs et $N_0$ utilisateurs. . . . .	169
B.7	MMSE et filtre de Kalman Estimation de processus: . . . . .	170
B.8	MMSE et filtre de Kalman Estimation de processus fluctuations . . . . .	170

# List of Tables

4.1	SACK over ACK . . . . .	86
4.2	G-ACK over ACK . . . . .	87
4.3	G-SACK over SACK (Single Destination) . . . . .	90
4.4	G-SACK over SACK (Multiple Destinations) . . . . .	92
4.5	Benefits of coding: 3 unicast flows. . . . .	93
4.6	Benefits of coding: 10 unicast flows. . . . .	93
4.7	Proposed over Basic Scheme: Topology 3 . . . . .	98
4.8	Proposed over Basic Scheme: Topology 4 . . . . .	98
4.9	Proposed over Basic Scheme: Topology 5 . . . . .	98
B.1	Schéma Proposé sur Schéma Base . . . . .	167



# Introduction

## Contents

1.1	Motivation and Approach . . . . .	22
1.2	Contributions of the Thesis . . . . .	25
1.3	Organization of the Thesis . . . . .	26

Mobile Ad hoc NETWORKS (MANETs) aim at making communication between mobile nodes feasible without any infrastructure support. The motivation for the development of MANETs has increased in the last few years due to the entry of intelligent devices with short-range wireless communication capability. If the spatial density of mobile nodes in a MANET is low, then an end-to-end path between a source and a destination almost never exists; two mobile nodes can communicate only when they come within the radio range of each other.

Sparse MANETs fall into the class of Delay Tolerant Networks (DTNs) which are sparse and highly mobile wireless ad hoc networks with frequent link disruptions and highly intermittent connectivity. Other examples of DTNs include sparse Vehicular Ad hoc NETWORKS (VANETs) [1], Inter-Planetary Networks (IPNs) [2], Pocket Switched Networks (PSNs) [3], Airborne Networks (ANs) [4], Mobile Social Networks (MSNs) [5] and Under Water Networks (UWNs) [6]. IPNs are the deep-space application of DTNs. Terrestrial applications of DTNs include UWNs, PSNs, VANETs, ANs and networks for developing regions [7].

In DTNs, due to highly intermittent connectivity, no contemporaneous end-to-end path may ever exist [8]. There can be several reasons for intermittent connectivity such as limited wireless radio range, sparsity of mobile nodes, limited energy resources, attack, and noise [9]. Due to intermittent connectivity, the nodes in a DTN adopt the *Store-Carry-and-Forward* paradigm of routing where a source has to depend on the mobility of other nodes, which

act as “relays”, and data packets and ACKnowledgments (ACKs) get transferred between a source and a destination through one or more relays.

In the next section, Section 1.1, we first compare DTNs with traditional wired and wireless networks in terms of routing and transport solutions. Such a comparison will help us to identify some important problems that need to be addressed with priority.

## 1.1 Motivation and Approach

Traditional routing protocols designed for MANETs such as the Ad hoc On-demand Distance Vector (AODV) protocol [10] or the Dynamic Source Routing (DSR) protocol [11] perform well only when a contemporaneous end-to-end path exists. In the presence of constraints like frequent disconnections, low network density and limited device capability, such as in DTNs, conventional routing protocols would not be able to function effectively. The store-carry-and-forward routing paradigm provides a mean to enable communication in such challenged environments, even though it inherently entails a large delay of communication. The store-carry-and-forward routing approach permits the nodes to take advantage of the transmission opportunities during their contacts with other nodes for exchange of packets. Thus, such forwarding strategy is sometimes also referred to as *opportunistic routing* [12]. Several methods for spreading packets in DTNs have been investigated under the settings of opportunistic routing, for example, *spray-and-wait routing* [13], *probabilistic routing* [14], *multiperiod spraying* [15], *direct delivery* and *two-hop routing* [16], and *epidemic routing* [17].

The User Datagram Protocol (UDP) [18] and the Transmission Control Protocol (TCP) [19] are the most widely used transport protocols in the Internet today. The former provides a connectionless service and it lacks delivery guarantee whereas the latter provides a connection-oriented service and ensures reliable data delivery with delivery confirmation. By far, TCP is the *de facto* standard for reliable data transmission.

In reliable data transfer, the data source wishes to ensure that all the information it sends arrives correctly and “in order” at the destination. In TCP, data packets are given unique sequence numbers so that the destination can detect which packet(s) is/are missing. To achieve reliability, the destination sends acknowledgements back to the source for each packet (or for each other packet, in case of delayed ACKs) received without errors. A packet  $i$  is considered to be lost if either it is not acknowledged within some time period,  $T_0$ , or if the ACKs for three packets sent more recently than packet  $i$  arrive at the source before the ACK for packet  $i$  (so called *triple duplicate ACKs*). In addition to this, TCP implements a flow/congestion control mechanism wherein packets are sent within so called *congestion windows*, whose size increases with the receipt of ACKs and decreases after a

time-out or in case of triple duplicate ACKs.

TCP generally performs well in networks made up of links with low bit error rates. In networks with higher bit error rates, such as those consisting of wireless links and mobile hosts, many of the assumptions made by TCP are violated, which results in end-to-end performance degradation in such networks. For example, TCP assumes that packet loss occurs due to network congestion and TCP adapts well to network congestion. However, it misinterprets packet losses due to corruption by interference and noise as being due to congestion. TCP's congestion avoidance mechanism can only make things worse in such situations. Moreover, TCP requires at least one stable path between the source and the destination to establish the connection used in data delivery. However, such a path may not exist in wireless networks. A detailed account of the limitations of TCP in wireless networks can be found in [20].

A solution to the problems mentioned above has been provided in [21] where the authors suggest to “signal” losses due to packet corruption (rather than congestion) from the link layer to the transport layer. Balakrishnan et al [22] proposed a “snoop” protocol to improve TCP performance in wireless networks. Mahmoodi et al [23] proposed a TCP-aware dynamic Automatic Repeat-reQuest (ARQ) algorithm, which is a cross-layer design that improves TCP's performance in wireless networks with link layer adaptation.

In addition to high bit error rates, MANETs might also suffer from link failures. TCP turns out to be very inefficient for reliable transport in MANETs, because it misinterprets losses due to link failures as losses due to congestion [24]. This is even worse in the case of DTNs which suffer from frequent link failures [25]. DTNs are usually sparse and disconnected where the lack of connectivity between the source and the destination is common. Most of the time, it is impossible to establish data connection and to confirm delivery [26]. Therefore, TCP is not functional in DTNs [27].

In summary, protocols for traditional wired and wireless networks appear to be unsuitable for DTNs which motivates development of new approaches to transport and routing in DTNs. Most of the proposals for DTNs focus on the routing aspect. Transport solutions are mainly proposed for deep-space communication (see Section 2.1). Existing solutions in MANETs rely on cross-layer signaling between the transport and lower layers so as to inform the former about route failures [21], [22], [23]. This strategy cannot be used in DTNs as only opportunistic routing can be performed. Therefore, one of our main focus in this thesis is on designing an efficient reliable transport protocol for terrestrial DTNs. This pertains to reliable transport with the help of explicit ACKing by a unicast destination. The other main focus is to monitor and track the extent of data dissemination (see at the end of this section), which pertains to tracking the degree of transport completion in broadcast/multicast applications.



It is common misunderstanding that minimizing delay is not important in DTNs. In fact, the term “delay tolerant” has the connotation that protocols have to be designed in such a manner that the applications can run over the network even in the presence of long delays caused by frequent link disruptions. In this thesis, minimization of the end-to-end delay is used as the criterion for designing an efficient reliable transport protocol.

Protocols based on rateless codes are appealing alternatives to TCP in a future Internet, in general, and in DTNs, in particular [28]. Assume that a file consisting of  $K$  packets is to be transferred from a source to a destination. Protocols based on rateless codes allow the source and/or the relays to keep on sending newly generated packets, called Random Linear Combinations (RLCs), which are combinations of randomly selected packets that the nodes already have. The loss of an RLC can be compensated by another RLC, whereas without coding, the loss of a packet has to be compensated by retransmitting the same packet. With coding, the only thing that matters for the destination is to meet a certain Degree of Freedom (DoF) corresponding to the number of linear independent RLCs it has to receive in order to be able to decode the file, and not the receipt of specific packets.

In the absence of feedback from the destination, the source cannot know for sure how many packets, be they coded or not, made it successfully to the destination, and hence, only a “probabilistic” form of reliability can be obtained; the source can only ensure a certain probability of successful delivery of packets to the destination. Use of end-to-end ACK is essential to ensure that packets/RLCs have indeed reached the destination and bring in the “deterministic” form of reliability.

Coding can be used not only to minimize the delivery delay [29], [30] but also to improve the probability of successful delivery within a given time [28]. However, we still require an end-to-end ACK mechanism in order to be sure that the RLCs have indeed reached the destination, just like TCP does for packets. To achieve the deterministic form of reliability and to minimize delay in DTNs, we make use of ACKs, in addition to coding.

Since multiple copies of the same packet are allowed to spread in the network, tracking the number of copies is quite important to have an *adaptive* replication policy. For a single message delivery, it is necessary to estimate the network state to control message spreading. In some multicast/broadcast mobile applications, it is also important to control the message spreading to a certain percentage of mobile users [31]. We address this problem in the case of direct delivery and epidemic routing.

## 1.2 Contributions of the Thesis

We make the following contributions in this thesis:

- **Reliable Transport:**

This work focuses on the transfer of a file from a single source to a single destination through intermediate relay nodes. The file consists of some information packets. The source sends RLCs of information packets and the destination sends back updated ACKs. The RLCs and ACKs are replicated as in epidemic spreading, with some control measures on the spreading of RLCs. We consider coding only at the source, i.e., the intermediate relay nodes do not perform any coding. We assume exponential inter-meeting times between nodes. We also make the assumption that the relay nodes can store at most one packet or one ACK at any point in time. Our objective is to minimize the end-to-end delay ensuring deterministic reliability through coding and ACKs.

- **Proposal:** We propose and study a new reliable transport scheme based on ACKs and coding at the source (see the algorithm in Section 3.3.1). Our objective is to ensure a *deterministic* form of reliability, i.e., we want to be sure that the RLCs have indeed reached the destination, just as TCP does for packets. And so, we also make use of ACKs, in addition to coding.
- **Modeling:** We develop fluid models for the network under our proposed scheme in the case of large number of nodes and to demonstrate its accuracy by comparing with simulations.
- **Optimization:** Our transport scheme minimizes the mean round-trip delay. Using our fluid models, we carry out joint optimization of the set of optimal parameters, i.e., number of RLCs to be sent in one cycle, spreading time of RLCs and ACK wait time so as to minimize the transfer time of a file.

- **Enhancements:**

We propose a new set of enhancement schemes to improve the performance of reliable transport for unicast or multicast flows or a combination thereof. We consider multiple flows. Each source sends only one packet to its intended destination(s). We propose smart ACK mechanisms to improve the return path. Packets from different sources are combined at the relays inside the network to form RLCs (intersession coding) which helps in improving the forward path. Packets/RLCs and ACKs are replicated as in epidemic spreading.

- We propose and study several enhancement features (enhanced ACK schemes) to improve the performance of reliable transport. The enhancement features increase the probability of receiving the final ACK (i.e., the ACK indicating zero missing DoF) within a given timeout. Equivalently, they decrease the waiting times (i.e., durations of the timeouts) required to achieve a given probability of receiving the final ACK. We propose a novel Global Selective ACKnowledgement (G-SACK) scheme.
- We learn that use of network coding improves success probability, and reduces mean delay and delay variance from source(s) to intended destination(s), whereas, use of smart ACKs or G-SACKs provide huge improvement in success probability, variance and mean delays for the packet(s) receipt information generated by destination(s) to the source(s).

- **Tracking Message Spread under Direct Delivery and Epidemic Routing:**

In this work, we consider a DTN consisting of some static sources, static external observers and mobile users. The sources and the observers are connected by a wired network. Upon meeting with relay nodes, an observer counts the number of copies of the message. The relay nodes it meets inform it of whether they are carrying a copy of the packet. The problem is to get as accurate an estimation as possible of the number of nodes with copies using the measurements of the observers.

- **Kalman Filter:** we derive a discrete time Kalman filter based on diffusion approximations for the spreading process and the measurements.
- **Prediction:** we estimate/predict the time at which a given fraction of population has received copies of the file.
- **MMSE:** We derive the exact expressions for the instantaneous linear Minimum-Mean-Squared-Error (MMSE) estimator given the observer's count at time  $t$  under direct delivery and compare its performance with that of Kalman filter.

### 1.3 Organization of the Thesis

The remainder of the thesis is organized into the following chapters.

In Chapter 2, we discuss the literature on reliable transport in DTNs and find that most existing works concern mainly with deep space communication. We also cover routing aspects in DTNs with special emphasis on epidemic routing.

In Chapter 3, we describe our proposal of reliable transport, provide analytical model for our proposal and its optimization. We also propose two approximations of our proposal to overcome computational complexity.

In Chapter 4, we propose some enhancement features to our reliable transport proposal which help in minimizing transmission delays and making transmission successful in finite time.

In Chapter 5, we track message spreading in DTNs under direct delivery and epidemic routing.

In Chapter 6, we conclude our work and provide perspectives.



# State of the Art

## Contents

---

<b>2.1 Literature on Transport Protocols . . . . .</b>	<b>30</b>
2.1.1 Transport Proposals for deep-space Communication . . . . .	30
2.1.2 Transport Proposals for Terrestrial DTNs . . . . .	35
<b>2.2 Literature on Routing . . . . .</b>	<b>37</b>
<b>2.3 Literature on Network Coding . . . . .</b>	<b>40</b>

---

Identification of opportunities for deployment of DTNs in the terrestrial Internet has attracted a lot of attention of the networking research community, mainly due to its challenging characteristics. Intermittent connectivity, high mobility, unknown mobility patterns, energy and storage exhaustion are just a few of the potential issues that may be faced in a DTN environment. Research in DTNs is mainly focused on the routing aspect.

The Delay Tolerant Network Research Group (DTNRG) [9] of Internet Research Task Force (IRTF) has investigated security and transport layer issues, but mostly in the direction of space communications [32], [25], [33], [34], [35], [36]. The main focus of the DTNRG is on the Interplanetary DTN environment. Routing or transport layer issues for terrestrial DTNs have not received any attention from the DTNRG so far. After reviewing the approaches of DTNRG, Fall and Farrell [37] reported that the preliminary architecture meant for deep space communication would fit poorly into terrestrial DTNs due to lack of design flexibility. Lack of attention on reliable transport aspect for terrestrial DTNs motivates us to devise a mechanism to improve reliable transport for DTNs.

## 2.1 Literature on Transport Protocols

Much of the existing literature on DTNs focuses on the routing aspect and relatively fewer pieces of work deal with reliable transport. Harras and Almeroth [38] and Akan et al [39] probed into issues related to the use of transport in a DTN environment. Fall and McCanne [40] discussed important issues related to transport performance not specific to DTNs.

Proposals for transport layer in DTNs can be classified as: (a) deep space, and (b) terrestrial, environments.

### 2.1.1 Transport Proposals for deep-space Communication

Space Internet or space Internetworking is to enable space communication using Internet-type protocols and the Internet in the deep-space environment is generally termed as Interplanetary Internet [41]. Deep-space environment entails huge delays even in the presence of connectivity. The majority of the architectural designs and works, mainly [42], [43], [44], [32], [36], [39], [45], [46], [47], address deep-space communication.

The Bundle Protocol (BP) [42] proposed by DTNRG [9] borrows from the concept of email protocol and describes the end-to-end exchange of messages called “bundles” in DTNs. A bundle node consists of (i) a bundle protocol agent, (ii) a set of convergence layer adapters, and (iii) an application agent. The application agent makes a transmission request to initiate a bundle transmission. Selecting appropriate convergence layer adapters after the establishment of endpoints, an outbound bundle is created according to the request parameters. Then, this bundle is sent to all active nodes currently associated with the specified endpoint. A bundle receipt acknowledgment request can be made for each transmission. Bundle protocol agent performs bundle forwarding. It sends a bundle to all nodes currently associated with a specified endpoint which could be either (i) final destination endpoint or (ii) other intermediate endpoints with the ability to forward the bundle to the destination. Convergence layer adapters must notify the bundle protocol agent about the completion of the transfer. The custody transfer, a mechanism which ensures the storage/custody of a bundle copy in the forwarding node until the release of the custody, can be requested for the transaction of each bundle. The BP specifies a framework rather than a concrete protocol implementation. Besides, the current BP works only for the content whose size is already known before the transmission. The size of bundle may sometime increase considerably. As per [48], BP does not really provide an end-to-end service as claimed in the Request For Comments (RFC) 5050 [42]. [49] highlights quite interesting aspects of BP and contends that reliability in BP’s custody transfer function is prevented due to lack of error detection approach from the specification (e.g., checksum).

Reliability through custody transfer in bundle layer protocol is briefly addressed in [50].

A custodian takes the responsibility of reliable delivery of the bundle to the next custodian till bundle is delivered to the destination.

The Space Communication Protocol Standards-Transport Protocol (SCPS-TP) [43] for space communications was proposed by Consultative Committee for Space Data Systems (CCSDS). This protocol adopts major TCP functionalities and extends them in order to deal with some of the unique characteristics of deep-space links. This protocol has two modes of operations (i) TCP-Vegas [51] based Van Jacobson congestion control mode and (ii) the open loop rate control. It operates in one of the two mentioned modes during congestion indications. By choosing the mechanism of the TCP-Vegas congestion control, it can adjust the window according to change of Round Trip Time (RTT). TCP vegas is able to monitor effectively the status quo of usage of network, which has been widely employed in the network of satellite for its better fairness and effectiveness. To address the limited bandwidth and to provide more efficient loss recovery, SCPS-TP makes use of header compression and Selective Negative ACKnowledgment (SNACK) options. Its open loop rate control mode also makes use of SNACKs instead of usual ACKs. The SCPS-TP can judge status of links weather broken or not by monitoring signal strength between the spacecraft and earth-based station. When the signal strength is lower than a threshold, the sender sends Internet Control Message Protocol (ICMP) to all the receiving stations till the links become normal again. Then in order to maintain the communication, it continues to send the ICMP to readjust the rate of sending. To deal with bandwidth's asymmetry problem, the SCPS-TP adopts the time-delayed ACK approach.

The CCSDS File Delivery Protocol(CFDP) [44] has the capability to transfer files to and from space craft data storage. It is mainly an application layer protocol, but includes functionalities of transport layer too. The core procedures of CFDP provide file copy services over a single link, i.e., its operation is to copy files from a source storage medium to a target storage medium. It can offer both unreliable and reliable (based on SNACKs) services. In the absence of direct connectivity between source and destination nodes, the protocol introduces extended procedures capable of performing multiple file copy operations across each link of the path to final destination. CFDP does not support route adaptation inherently, therefore, the complete autonomous handover operations like ground station handover are established by the system functions external to CFDP. CFDP includes four modes (deferred, immediate, prompted and asynchronous) for sending negative ACKs. It also makes use of positive ACKs to ensure the receipt of critical PDUs. The extended mode of operation of CFDP refers to more complicated situations, and provides store-and-forward feature across a network having multiple links with disparate availability. Current store-and-forward approach of CFDP requires that all parts of a file follow the same path from a source to a destination. It appears to be unsuitable with increased communication



complexity like dynamic routing or file distribution through multiple relay points in parallel. A solution currently under [48] investigation is the integration of CFDP with BP [42].

For dedicated networks like deep space with very high RTTs, the Licklider Transmission Protocol (LTP) [32] offers reliable transfers based on retransmission over single-hop connections. To support BP over long and/or frequent intermittent links, it serves as a DTN convergence layer protocol which is its major design purpose, though can be used in other contexts. Such links might be encountered in interplanetary network setting. LTP's standard use case is a single hop, deep space link such as between a remote spacecraft and an earth station. It does not consider typical TCP issues such as flow and congestion control in shared networks. Thus before triggering communication, it makes use of pre-defined parameters like window size in its current implementation. In a typically very high delay point-to-point environment operating over single links, LTP handles delay tolerance and disconnection issues. LTP supports both reliable and unreliable data transmission. Since LTP is a point-to-point protocol, so there are no routing or congestion issues to consider for LTP itself. Bytes are simply transferred between two peers without any consideration of intermediate peers. LTP can transfer unnamed data blocks and brings partial reliability by dividing each block into two parts, (i) reliable "red" part and (ii) unreliable "green" part. ACKs are sent only when explicit solicitations are encountered for checkpoints (reception reports) in the sequence of incoming data segments of the block's red part. It still allows deferred transmission in case the communication link is not available. LTP runs just above link layer and may also be useful for some terrestrial applications like sensor networks [52].

Farrell and Cahill [53] proposed a transport protocol called LTP-T (a variant of LTP [32]) which targets deep-space applications. LTP-T uses the notion of custody transfer. Each LTP-T entity must accept custody for all successfully received blocks. Custody is thus passed from host to host until the final destination is reached. This can minimize end-to-end data delivery time, though the requirement of accepting custody for all received blocks can also lead to storage exhaustion problems. The protocol supports delay-tolerant transport and congestion notification. Reliability and related issues become more complex as compared to LTP since LTP-T is designed for multi-hop environment. Basically, LTP-T operates like a sequence of independent LTP sessions (one for each session) when no error occurs. In the presence of segment loss or corruption, successfully received segments are forwarded to the next node while error recovery is started for the others. This desynchronization of the initial segment sequence brings in proper checkpoint re-scheduling. However, LTP-T is only defined in a generic way, and further details of how it would operate in reality are not known.

The Deep-Space Transport Protocol (DS-TP) [36] addressed efficient and reliable communication in deep space. It favors missions with small connectivity time due to its ability

to complete file transfer faster than conventional protocols like TCP. DS-TP adopts (i) open loop approach to deal with huge propagation delays, whereas TCP's closed loop transmission approach is proven to be not sufficient to tackle with propagation delays in the order of minutes, i.e., feedback comes back at the sender when the information included in the ACK(s) may already be useless. In other words, it is possible that actual congestion event happens several minutes before triggering of congestion control and recovery mechanisms by TCP's triple duplicate ACKs, (ii) SNACKs (a common approach) to deal with bandwidth asymmetries in satellite and space communications. SNACKs are able to signal for multiple holes at the receiver's buffer in contrast to simple Negative ACKnowledgment (NACK), and (iii) fixed-rate based transmissions to allow for high link utilization without forcing increase in the transmission rate that ultimately leads to congestion losses. DS-TP focuses on the optimization of Double Automatic Retransmission (DAR), a retransmission approach of the transport protocol, to deal either with high bit error rates or blackouts and to provide proactive protection against link errors. DAR allows for fast and efficient "hole-filling" at receiver's buffer. The level of redundancy introduced by DS-TP affects both the storage space requirement of intermediate DTN nodes and the end-to-end delivery delay of data to its destination.

The Transport Protocol for InterPlanetary Internet (TP-Planet) [39] is one of the early proposals for reliable data transmission over single deep-space links. Its main functionality is to probe congestion detection and control mechanism which deals with congestion losses. It uses a blackout state procedure to cope with blackouts and also makes use of delayed Selective ACK strategy to handle bandwidth asymmetry. To be more precise, assuming the presence of IP infrastructure in space, it has deployed rate-based Additive Increase Multiplicative Decrease (AIMD) congestion control. AIMD operation depends on the decision of the congestion detection mechanism. TP-Planet adapts AIMD parameters in order to compensate the throughput degradation due to propagation delay. The TP-Planet protocol has issued two new algorithms namely (i) the initial state and (ii) the steady state. The initial state algorithm can occupy the bandwidth of the links in a very fast and controlled manner. It helps in avoiding throughput degradation which occurs due to inefficient connection starting behavior. The steady state algorithm contains four states: (i) hold or monitoring rate (ii) increase or accelerating rate, (iii) decrease or slowing rate and (iv) blackout or interrupt state. At the start of the steady state where there is no change in transmission rate, the source goes to hold rate state. The TP-Planet deploys a new congestion control scheme during steady state operation which helps in deciding transitions between the states in the steady state. Hence, it can increase, decrease or hold data transmission rate according to the current state. On the intermittent connection, the steady state algorithm has introduced the program of interrupt. TP-Planet has brought in the time-delay option

of SACK to deal with asymmetric bandwidth. The TP-Planet is an end-to-end connection, but the original node and the destination node may not exist in the end-to-end connection in the deep-space communication networks.

The RCP-Planet protocol [45] is an unreliable version of TP-Planet [39] proposed by the same authors. It employs a rate control mechanism which copes with link congestion and error rate as well as a packet-level Forward Error Correction (FEC). It deploys a blackout state procedure and FEC block-level ACKs in order to handle bandwidth asymmetry. Delivery of real-time application data to the ground or to the satellite, spacecraft etc is the main objective of RCP-Planet.

Saratoga [46] developed by [54] is a file transfer and content dissemination protocol. It was originally developed with the purpose to transfer large files from small low-Earth-orbit satellites, but is useful for many other situations including ad-hoc peer-to-peer communications and DTNs. Its intended use is for moving files or streaming data between peers with intermittent connectivity. Saratoga is capable of reliably transferring very large amount of data under adverse conditions. It provides an IP-based convergence layer in DTNs to exchange bundles between peer nodes, supporting store-and-forward of bundles. Store-and-forward delivery approach relies on reliable hop-by-hop transfers of files. Saratoga supports retransmission requests of missing segments, uncorrelated with the data sending process, and contains SNACKs in order to provide reliable retransmission of data. Based on scheduled connections, Saratoga also supports an optimistic data transfer mode, where the source, immediately after initiating a communication session, starts data transmission without waiting for an acknowledgment. There are at least two feasible options to transfer a larger file which cannot be transferred by Saratoga during a time-limited contact. These are: (1) proactive fragmentation can be used by the application to create multiple smaller-sized files, enabling Saratoga to transfer some of these smaller files completely during a contact. (2) use of HOLESTOFILL packet to avoid file fragmentation. The receiver can retain a partially-transferred file and request transfer of the un-received parts during a later contact by using HOLESTOFILL packet in order to make clear that how much of the file has been successfully received and where transfer should be resumed from. The best case of Saratoga requires just two acknowledgements: one for session establishment and one after successful data receipt. Saratoga also supports an optimistic transfer mode which can be used in situations involving well known communicating parties and scheduled transactions. Generally, it can be stated that, although Saratoga may not always work so reliable, it is more flexible than TCP. It can be adapted to different communication scenarios, providing storage and forwarding of bundles. It provides efficient means for data transfer, minimizing overheads and maximizing throughput through the best exploitations of available resources.

Delay-Tolerant Transport Protocol (DTTP) [47] is proposed in order to increase reliability and efficiency (in terms of resource utilization) in DTNs. To increase efficiency of space data transfer, the authors claim to introduce dynamic characteristics therein. For the same DTTP session, through potential multiple paths, it offers multi-hop data transfer services. Practically, it relies jointly on an open-loop model and a closed-loop system for transmission scheduling and administration respectively. DTTP provides both reliable and unreliable communication with trade-off between reliability and end-to-end delivery delay. To provide reliable communication, DTTP agents run on every node along multi-hop paths, and reliability is offered separately on each link of the path. In order to ensure complete transfer of data, it supports cumulative and selective ACKs used collaboratively in its ACK packets. Its custody transfer approach transfers reliable transfer responsibility from one node to the next on the path till final destination. Full reliability requires extensive retransmissions, which obviously extends overall transfer time.

### 2.1.2 Transport Proposals for Terrestrial DTNs

Terrestrial DTNs entail Internet like communication in terms of delays once the connectivity is there. Now, we detail a few proposals for terrestrial DTN environments [38, 55–58].

The main function of the Persistent Connection Management Protocol (PCMP) [56] is to keep the TCP connection alive during disconnections in DTN environments like Drive-Thru. The idea of Drive-Thru Internet is to provide hot spots along the road within a city, on a highway, or even on high-speed freeways such as autobahns [59]. The approach of PCMP is similar to Snoop-TCP [22]. Important aspects related to data transfers within connectivity islands in Drive-Thru environments are addressed by PCMP. The connectivity of a car entering a road based WLAN is split into the entry, production and exit phases. In case the transfer does not complete during the production phase, then this issue is handled by keeping the state of the connection alive till the arrival of the car to the next connectivity island. The authors in their study [56], through real road experiment, claim transfer of big amounts of data even for high speed car. PCMP assumes that the next connectivity island for a car moving on a highway is known in advance opposed to real, wide-deployment cases where the movement of cars is normally arbitrary.

Schutz et al [58] measure the behavior of Internet communication across a dynamically changing, intermittently connected path through experiments. Through analysis of experimental results, they observe that the address changes, transport-layer timeout and retransmission behavior (characteristics of intermittent connectivity) are the main limiting factors. On the basis of these results, they propose enhancement protocol by combining the Host Identity Protocol (HIP) [60] with the TCP User Timeout Option [61] and the TCP Retransmission Trigger [62]. Their enhanced protocol can avoid connection aborts due to

disconnection periods, utilize more efficiently connectivity periods and avoid connection aborts due to IP address change by using TCP User Timeout Option, TCP Retransmission Trigger and HIP, respectively.

Harras and Almeroth in [38] elaborate on acknowledgment approaches for Mobile DTN (MDTN) environment. The scenario considered is as follows: a source node forwards a message to a destination node through relay nodes. The work considers four different acknowledgment strategies. Trade-offs between delivery reliability, queuing time and delivery ratio are investigated based on the acknowledgment strategies. The acknowledgment strategies are:

- In the Hop-by-hop approach, each relay node takes responsibility of custody transfer. Each relay, upon receiving the message successfully, sends back an ACK. ACKs are sent along every hop in the path. The source as well as the relay nodes try to infect as many nodes as possible. The source assumes that the message will eventually reach the destination due to provision of enough time and mobility of nodes. Thus, hop-by-hop ensures some level of reliability, but it does not ensure end-to-end reliability.
- Next, in the active receipt approach, the destination generates an ACK (receipt) after receiving the message. The generated ACK actively flows back to the infected nodes in order to cure them. Actually, nodes treat this ACK as a new message needed to be forwarded.
- To overcome the cost related to active receipt, the destination node sends a kill message (implicit/passive receipt) which travels back to the source. Relays carrying the kill message do not forward it, they simply wait for the active infected node to come in their communication range and they stop them sending the source message further.
- The network-bridged approach uses alternative technologies, like cellular network, in order to acknowledge the message receipt. It could make use of the cellular network, exploiting its availability, as an alternative path for communication, as the same could be used to transfer small control information due to its bandwidth limitations of transferring large amount of data. Thus, the cellular network functions as a bridge between MDTN nodes. This approach adds complexity of bridging the DTMN network with cellular network, the availability of the same cannot be guaranteed in a DTN setup.

Approaches to avoid storage congestion in DTNs are investigated by Seligman et al [55, 57]. When a node becomes congested, it forwards some fraction of its stored messages to other nodes who take custody responsibility. Its storage routing can utilize an extended

push-pull concept of custody transfer. Such forwarding de-congests the node and makes it available to serve incoming traffic. The algorithm consists of two parts. In the first part, a node selects a message from the stored messages in order to forward the same. Then, it selects a node in its  $k$ -hop neighborhood to whom the message is to be forwarded. These works conclude that huge storage capacity might be required in order to achieve delivery reliability. Delivery delay may be increased potentially due to message circulation, i.e., data is exchanged between neighbor nodes and not forwarded towards its ultimate destination node. Buffer management issues in DTNs are also discussed by Krifa et al [63, 64].

**Comparison of our work with literature on transport:** As stated earlier, most of the existing proposals for DTNs focus on the routing aspect and relatively fewer works address reliable transport. Moreover, transport solutions are mainly proposed for deep-space communication. In this thesis, we mainly focus on designing an efficient end-to-end reliable transport protocol for terrestrial DTNs.

Like BP [42], we provide a framework for DTNs rather than a concrete protocol implementation solution and our proposal works when the content size is known a priori. But, unlike BP, we ensure an end-to-end reliable delivery. We make use of smart ACK-ing similar to [43]. We combine the ACK information to represent them compactly as G-SACKs. Unlike [43], our proposal is not crucially dependent on the existence of specific links. In fact, due to terrestrial settings and availability of a sufficiently large number of highly mobile nodes, we can use epidemic-type spreading and take benefit of multiple parallel paths. Unlike [43], in our case, bandwidth asymmetry is due to finite buffer and we propose killing/replacing of the less important ACKs by the more important ones and final ACK also killing RLCs. In [43] and [44], retransmission is triggered by sending SNACKs, whereas we use time-outs to trigger retransmission. [44] requires that all parts of the file follow the same path from source to destination. We make use of multiple paths via epidemic routing through multiple relay links in parallel. We offer reliable transfer based on end-to-end retransmissions, whereas in [32] retransmissions are over single-hop connections. Saratoga [46] and [47] are reliable transfer protocols based on hop-by-hop transfer of files, whereas our protocol provides an end-to-end reliability where intermediate relays provide only best effort service and drop the packets at specified timeouts. Unlike [46], we use coded packets, and hence, we do not need to recover specific missing packets.

## 2.2 Literature on Routing

As stated earlier, traditional routing algorithms like AODV [10], OLSR [65], LAR [66], DSR [11] and many others [67–69] proposed for MANETs require contemporaneous end-to-end connectivity and these conventional protocols can not function effectively in DTNs

due to frequent disconnections. The store-carry-and-forward routing approach makes communication possible in these disconnected environments. This approach permits the nodes to take advantage of the transmission opportunities during their contacts with other nodes for exchange of packets. Thus, such forwarding strategy is sometimes also referred to as *opportunistic routing* [12]. The routing aspect in DTNs has been addressed widely. It has become almost an independent research area. There exists not only a huge literature but also new works appear continually on routing in DTNs.

Many surveys such as [70–75] provide detailed and extensive literature on routing in challenged networks. Most surveys categorized routing protocols in flooding and forwarding schemes. Flooding schemes require relay nodes to store-and-forward message copies independently through creating multiple duplications of a message in the network. During message transmission in forwarding schemes, only one copy of each message remains in the network like traditional routing strategies.

Pelusi et al [73] surveyed the interesting case studies related to opportunistic networking and organized a taxonomy for the routing approaches therein. The case studies include pocket switched networks in the European Commission funded project called “Haggle Project” (<http://www.haggleproject.org>), wildlife monitoring [6, 76] and opportunistic networks for developing regions [7, 77]. They classified the algorithms in two main categories, (i) algorithms without infrastructure designed for completely flat ad-hoc networks, and (ii) infrastructure based algorithms which exploit some form of infrastructure in order to forward messages opportunistically. They further divided the infrastructure based algorithms into *fixed infrastructure* [6, 78] and *mobile infrastructure* [79, 80] and approaches without infrastructure into *dissemination-based* [81–84] and *context-based* [85, 86].

Zhang [74] surveyed and categorized the routing protocols based on information used. The survey discussed tree, space-time and the modified shortest path based approaches for deterministic time evolving networks where future network topology is predictable which allows pre-scheduling of transmission. Epidemic, randomized flooding, prediction or history based, model based, controlled movement based and coding based approaches were also discussed for stochastic time evolving networks where future network topology is completely unknown denying scheduling of message forwarding ahead of time.

Evan and Paul [70] divided DTN routing strategies into flooding and forwarding families based on two different properties called *replication* and *knowledge*. Replication is used in flooding strategy while knowledge is used in forwarding category in order to find the destination. Direct contact, two-hop routing, tree-based routing and epidemic routing protocols come in flooding family while location based routing, gradient routing and link metrics routing belong to forwarding family. Moreover, Shen et al [71] added prioritized epidemic routing, probabilistic routing and reconfigurable ubiquitous networked embedded



systems routing in flooding family while source routing, per-hop routing, per-contact routing and hierarchical routing in forwarding family. Lui et al [72] in a recent survey classified forwarding schemes into infrastructure-based, prediction-based and social-based forwarding strategies while flooding schemes into spray series, social-based flooding, intention oriented, coding-based and hybrid-based strategies.

Khabbaz et al [75] provided a comprehensive survey regarding advancements and challenges in DTNs. They classified the DTN routing works on the basis of following forwarding approaches: opportunistic, vector-based, delegation, probabilistic, load balancing-based, encounter-based, resource allocation-based, spray-and-wait and network coding based forwarding.

*Direct Delivery* (DD), *two-hop*, *epidemic* and *spray-and-wait* are mostly used routing approaches in DTNs. In DD [16], source node keeps the message until it encounters the intended destination. In two-hop-routing [16], the source replicates each generated packet to intermediate relay nodes which do not replicate/spread the message to other relays. So, the message is delivered to the destination when intermediate node carrying source packet encounters with the destination. Thus, transmission completes in two hops.

Vahdat et al [81] proposed *epidemic routing* for routing in disconnected networks. In epidemic routing, data packets are flooded to all nodes in the network to minimize the delay. A node buffers the received packet, carries it as it moves and passes the packet on to a new encountered node that does not have a copy of that packet. Similar to the spread of infectious diseases, the carrier of packet infects the new encountered node by passing on a copy of the packet; which, in turn, behaves in the similar manner. The destination node, upon its first meeting with an infected node, receives the packet. Thus, the quick replication of the message throughout the network reduces the delivery delay. Epidemic routing could guarantee the maximum delivery ratio due to replication.

Many more works followed the original proposal on epidemic routing. A link-state protocol is proposed by Jones et al [87]. Global knowledge is disseminated epidemically and packets are forwarded on a single path. Many factors like weather, system failure, radio interference and in particular, unpredictable mobility may affect the routing on a single path making it error prone. The destination will not receive the data in case of data transmission failure at this path.

[88] and [89] studied performance analysis of epidemic routing. The authors in [90], for opportunistic networks, demonstrated the performance of epidemic routing using network coding in comparison to the use of replication. The authors in [29] combined network coding with epidemic routing [81] to attain a lower overhead ratio specially for dissemination of large number of messages.

Spyropoulos et al [91] proposed an innovative algorithm *spray-and-wait* to control flood-



ing. It is a combination of epidemic routing [81], two hop routing and DD [16]. Algorithms with controlled replications distribute only a small number of copies to other nodes. After distributing a number of copies to its neighbors, the source waits hoping that one of the neighbor nodes will meet the destination. Such forwarding approaches help in reducing the overhead induced by flooding. The approach can be adjusted to meet specific deadlines, for example, lower delay is achievable by shorting wait periods. [13, 83, 87, 92–94] study algorithms with controlled replications. The assumptions about the network made by these algorithms differ from each other.

The performance of routing and transport in DTNs is very much dependent both on the mobility of the nodes and the packet replication method [12].

## 2.3 Literature on Network Coding

Network coding is a technique where, instead of simply relaying the packets of information they receive, the nodes of a network take several packets and combine them together for transmission. This can be used to attain the maximum possible information flow in a network. Network coding research originally studied throughput performance without delay considerations for channels with no erasures and no feedback [95], [96], [97].

Ahlsweede et al [95] initiated the study of network coding. They showed examples that demonstrated the benefit of network coding, in terms of throughput improvement. They also proved the fundamental result that, for a multicast transmission in a directed network, if a rate can be achieved for each receiver independently, it can also be achieved for the entire session. According to [95], in random linear coding, each network node can forward along the path from source-to-destination random linear combinations of the received data rather than forwarding packets unchanged.

Ho et al [98] proposed the random coding technique originally. They defined the failure probability of random linear network as the probability with which at least one sink node cannot decode the messages correctly which is upper bounded. Li et al [96] showed linear codes over a network to be sufficient to establish any feasible multicast connection. Katti et al [99] showed that network coding can improve the throughput in unicast wireless communication. Zhang et al [30] investigated the benefits of using random linear coding for unicast communications in a mobile DTNs under epidemic routing.

Difference between network coding and erasure coding is provided by Fujimura et al [100]. In the former, instead of simply relaying the received packets of information, the nodes of a network take several packets and combine them together for transmission to attain the maximum possible information flow in a network, while in the latter, only source node can encode the message. Fountain codes can be applied to the reliable multicast [100].

Oh et al. [101] compared network coding to rateless coding (for multicast). They discovered that in extreme disruption (mobility + jamming) and dense networks, network coding outperforms end to end erasure coding and they expect the same to hold true for unicast as well.

#### **Proposals based on coding:**

Protocols based on rateless codes are appealing alternatives to TCP. For low-complexity rateless codes, such as Fountain and Raptor codes [102] [103], if the number  $K'$  of coded packets received at the destination is a little more than the initial number of information packets  $K$ , then with high probability, the destination can reconstruct the whole file. This success probability can be set as high as desired by properly choosing  $K'$ . For Random coding, or random linear network coding [104],  $K'$  can be equal to  $K$  to get high success probability, but the encoding and decoding complexities are, in this case, maximum.

Wang et al [105] combined erasure coding with two-hop-relay [16], which splits and encodes the message into a set of smaller size blocks. Upon receiving a portion of these encoded blocks, the original message is reconstructed by the receiver. A hybrid routing approach provided in [106] is enhanced version of [105]. The approach in [106] generates a copy of each encoded block, performing transmission for both of them at each contact opportunity. The transfer of original block is similar to [105]. However, aggressive forwarding is used to transmit its copy during the remaining encounter duration soon after the first block is sent out. The use of a smaller coding rate provides the opportunity to split the message into many encoded blocks of smaller size. Though, reliable delivery is promoted in this way but more redundancy would be generated. On the other hand, a higher coding rate might not be sufficient for delivery. Inspired by this consideration, a proposal to adopt rateless codes instead of adopting to erasure coding is provided by Vellambi et al [107].

For opportunistic networks, Lin et al [90] demonstrated the performance of epidemic routing using network coding in comparison to the use of replication. Through simulations they showed that under bandwidth constraint, smaller block delivery delay can be achieved by using random linear coding than non-network coded packet forwarding. If buffer space within DTN nodes is limited, then they observe the further increase in relative benefit as compared to an uncoded dissemination.

Zhang et al [30] investigated the use of random linear coding under epidemic routing for unicast applications in mobile DTNs. When bandwidth is constrained and packets are destined for the same destination from the same source, random linear coding over non-coded packets achieves smaller delay. This benefit further increases with limiting the buffer space in DTNs. With relatively loaded network, random linear coding achieves improvement over non-coding scheme only by appropriately controlling the spreading of information. Further, they observed that the benefit achieved by random linear coding

for the *multiple sources-single destinations* case is smaller than for the *single source-single destination* case. They claimed that if there is a single block of packets in the network, for the single source/multiple sources-single destination, random linear coding achieves the minimum delay with high probability.

Bulut et al [108], present multiperiod spraying algorithm for routing in DTNs. Erasure coding based technique is used to route message in order to ensure reliability against failures and increase the success rate of delivery. Their proposed scheme divides the original message into blocks which are individually transmitted. The destination decodes the original message by receiving only subset of original message or fewer blocks. Erasure based coding increases the chances of recovery of original message even if transmission failures occur. They address several cost reduction schemes maintaining the delivery rate and delay objectives. The number of periods is pre-decided. Other erasure coding based routing proposals are [105] and [109].

Lin et al [110] proposed E-NCP, an efficient protocol for DTNs. They combine network coding with binary spray-and-wait so as to control the number of transmissions significantly. Though, the reduction comes at the cost of slight increase in data transmission delay. However, using network coding makes the pre-determined maximum number of transmissions be more efficient for the delivery delay.

Jain et al [111] considered the problem of routing in a DTN in the presence of path failures. They tried to address issues like how the copies should be distributed, or how many of them amongst the available paths. They presented a theoretical approach to determine set of paths to be used, with known and independent path failure probabilities. They formulated DTN routing as a resource allocation problem, with reliability as the goal.

Chen et al [112] present a hybrid algorithm which integrates erasure coding with encounter prediction. They present message scheduling algorithms that enhance data delivery capabilities of the hybrid erasure coding scheme for file transfers. They propose a content-centric framework to facilitate data dissemination for video transfers and web surfing applications. Their proposed framework combines layered coding and multiple descriptions coding. The authors claim that their proposed schemes can achieve a much better latency performance for file transfers. CFP algorithm presented by Die et al [113] is an optimal probabilistic forwarding scheme using fountain code to encode message and provide the forwarding rule. They model the probabilistic forwarding problem as an optimal stopping problem.

In [114], a coding-based forwarding protocol called vCF is proposed for vehicular ad hoc networks with scheduled routes such as bus systems. In vCF, each message is fragmented into a set of blocks. The blocks are encoded via linear network coding. Each node, upon seeing a contact opportunity schedules coded blocks to transmit, and drops the blocks when

there is buffer overflow.

Widmer and Le Boudec [115] propose a communication algorithm based on network coding. Their proposed algorithm reduces the overhead of probabilistic routing algorithms. The authors claim through simulations that their algorithm achieves the reliability and robustness of flooding at a small fraction of the overhead. According to [116], the work in [115] is regarded as a hybrid of Gossip routing [117] and Epidemic routing [81].

A forwarding strategy called HubCode is proposed in [118]. Hubcode replicates the message only to the nodes deemed as being hub in the social based underlying topology of the DTN. The hubs use random linear network coding to encode multiple messages which are addressed to the same destination. Message are delivered to the destinations by using the hubs as relays. An extra exchange overhead occurs during the exchange of coefficient matrix required to check the linear independence.



# A New Reliable Transport Scheme

## Contents

<b>3.1</b>	<b>Related Work . . . . .</b>	<b>47</b>
<b>3.2</b>	<b>Network Setting . . . . .</b>	<b>48</b>
<b>3.3</b>	<b>Our Proposed Reliable Transport Scheme . . . . .</b>	<b>49</b>
3.3.1	Algorithm . . . . .	50
3.3.2	Random Linear Coding (RLC) and ACK Replication . . . . .	51
3.3.3	Implementation Issues . . . . .	53
<b>3.4</b>	<b>Analytical Modeling and Optimization . . . . .</b>	<b>54</b>
3.4.1	Modeling the Network Dynamics in One Cycle . . . . .	54
3.4.2	Combining the Cycles Together . . . . .	58
3.4.3	Optimization Procedure . . . . .	59
<b>3.5</b>	<b>Approximation 1 . . . . .</b>	<b>60</b>
<b>3.6</b>	<b>Approximation 2 . . . . .</b>	<b>60</b>
<b>3.7</b>	<b>Validation of the Analytical Model and Performance ... . . . .</b>	<b>63</b>
3.7.1	Proposed Model . . . . .	63
3.7.2	Approximation 1 . . . . .	65
3.7.3	Approximation 2 . . . . .	65
<b>3.8</b>	<b>Conclusion of the Chapter . . . . .</b>	<b>68</b>
<b>3.9</b>	<b>Appendices . . . . .</b>	<b>68</b>
3.9.1	Derivation of $f_{S_{i-1}, \dots, S_0}(s_{i-1}, \dots, s_0)$ . . . . .	68
3.9.2	Derivation of $E_{Direct}[T_{i \rightarrow 0}]$ . . . . .	70

We propose a new reliable transport scheme for DTNs based on the use of acknowledgments as well as random linear coding. In the absence of feedback from the destination, the source cannot know for sure how many packets, be they coded or not, made it successfully to the destination, and hence, only a *probabilistic* form of reliability is obtained; the source can only ensure a certain probability of successful delivery of packets to the destination. Coding is used to enhance the delay and/or the throughput performance by increasing the probability of receiving the message/file within a given time. Our aim, however, is to ensure a *deterministic* form of reliability, i.e., we want to be sure that the coded packets have indeed reached the destination. Hence, in addition to coding, we also make use of ACKs to cater to the losses in the network. Specific packets are not important for the destination with coding, instead, the only thing that matters for the destination is to meet a certain Degree of Freedom (DoF), corresponding to the number of linear independent Random Linear Combinations (RLCs) it has to receive in order to be able to decode the file.

We develop fluid models to derive expressions for the delay performance of our proposed reliable transport scheme. Employing a differential evolution optimization algorithm proposed by Price and Storn [119], we obtain the optimal values for the number of random linear combinations to be sent before time-out as well as the optimal value of the time-out itself, which, in turn, minimize the file transfer time. We refer to [120] and [121] for details on differential evolution.

We propose a scheme in which transmission is organized in cycles. During one cycle, the source sends a given number of RLCs, which is a function of the number of missing DoFs in a back to back manner without waiting for any feedback from the destination. The source allows the RLCs to propagate to the destination and then waits for the ACKs to come back until a so called *cycle time-out* occurs. Each RLC received at the destination triggers the sending back of an ACK, indicating to the source the number of DoFs still missing at the destination. In order that no flow or subset of flows monopolizes the (finite size) buffers at the relays, we allow for a so called *buffer expiry time-out* mechanism for each individual RLC in the relay buffer, at the end of which the given RLC is dropped. The cycle ends at the end of the cycle time-out, at which time, the source sends new RLCs, the number of which depends on the missing number of DoFs as indicated by the most up-to-date ACK the source has received so far. The process is repeated until the completion of the transmission of the whole file, i.e., until the destination has all DoFs required to recover the file.

First, we develop an analytical model for our proposed scheme, which we call “the model”. Then for missing DoF equal to 1, we validate our model by comparing analytical results provided by optimal procedure against simulation results obtained under optimal

settings. Then, due to complexity of optimal computation, we propose two approximations which eliminate the complexity of computation of optimal parameter settings to a large extent, but still provide quite accurate results.

**Our contributions:** We make the following contributions:

- We propose a new reliable transport scheme for DTNs based on ACKs and coding at the source.
- We develop fluid models for the network dynamics under our proposed scheme and demonstrate their accuracy by comparing with simulations.
- Our transport scheme is designed to minimize the mean round-trip delay. We carry out an optimization of the number of RLCs to be sent in one cycle, as well as of the value of the time-out, so as to minimize the transfer time of a file.

### 3.1 Related Work

Lucani et al [122] proposed a broadcast scheme for wired networks, which is optimal in terms of the mean time to complete the transmission of a fixed number of data packets. The sender can transmit an optimal number of coded packets back-to-back before stopping to wait for an ACK from each receiver. The optimal number of coded data packets, in terms of mean completion time, to be sent before stopping to listen depends on the latency, erasure probabilities and missing DoF at the destination. Their proposal considers only one cell without relaying (single base station).

A standard reference for Ordinary Differential Equations (ODEs) based models with epidemic routing is [88]. Zhang et al [88] used fluid modeling or differential equations to model system dynamics in DTNs in order to analyze the delay. They modeled the system dynamics with ODEs under epidemic routing, assuming that the inter-meeting times between pairs of nodes follow an exponential distribution. It uses fluid approximations and infection spreading models.

Bulut et al [15] propose a multiperiod spraying approach for routing in DTNs and try to optimize the dissemination efficiency. The goal of the algorithm is to minimize the number of copies used while delivering a predefined percentage of all messages by a predefined delivery deadline. A time-dependent copying scheme is introduced for making copying decisions. This scheme actually considers the time remaining to the given delivery deadline. The algorithm partitions time from message creation to the predefined deadline into several variable-length periods consisting of a message spraying phase followed by a wait phase for message delivery. To switch from one period to another, they assume some “feedback”. However, the method of obtaining the feedback is not defined.



Our work differs from [122], [88] and [15] as follows. In [122], erasure probabilities are given, while we derive them using fluid models. They take full cycle time for delay calculation, while we model delay as actual time at which final ACK is received. [88] models the transfer of a single packet of the source, while we model our approach for multiple packets and use it for optimization. In [15], number of periods is pre-decided, while it depends on the situation in our case. [15] brings only probabilistic reliability, while ours ensures deterministic form of reliability. Above all, we model ACK generation and spreading behavior.

### 3.2 Network Setting

To better explain our reliable transport scheme and to be able to develop a simple analytical model, we focus on the transfer of packets from a single source to a single destination corresponding to a single *flow*. The network consists of  $N_0 + 2$  mobile nodes. There is one source node, one destination node, and the remaining  $N_0$  nodes act as relays for the considered flow. The source wants to transfer a file consisting of  $M$  *information* packets to the destination. The source generates *coded* packets by forming RLCs of the  $M$  information packets and the destination sends back ACKs indicating the number of *Degrees of Freedom* still missing at the destination.

Due to the opportunistic nature of communication in DTNs, the RLCs and ACKs are replicated employing *epidemic routing* in which each mobile keeps forwarding a copy of its packet (or ACK) to the other mobiles it encounters, which also spread the packet (or the ACK) in that way. We say that two nodes “meet” when they come within the communication range of each other.

We assume that the successive inter-meeting times between any two specific relays, say  $i$  and  $j \neq i$ , are i.i.d. exponential random variables with mean  $1/\beta_r$ . We assume that the successive inter-meeting times between the source (resp. the destination) and any specific relay are i.i.d. exponential random variables with mean  $1/\beta_s$  (resp.  $1/\beta_d$ ). For simplicity, we ignore the possibility that the destination may receive the packet directly from the source, which occurs in DTNs with negligible probability.

Our assumption of i.i.d. exponential inter-meeting times is motivated by the works [88, 123, 124], in which the authors have shown via simulations that, for “random waypoint” mobility models such as the “random walk” or the “random direction”, the assumption of i.i.d. exponential inter-meeting times provides extremely accurate approximations for actual inter-meeting times, provided that the communication range  $r \ll L$ , where  $L \times L$  denotes the area of the network.

We assume that the relays have buffer capacity to store at most one packet or one ACK

at any point in time. In reality, a relay could store multiple packets and ACKs. However, in real DTNs, there would be several other competing flows sharing the buffer space at the relays. Thus, our assumption of buffer capacity of one packet or ACK can be viewed as saying that the buffer capacity of a relay is limited to one packet or ACK *per flow*. Furthermore, to make room for packets and ACKs of other flows, a packet (of the flow under consideration) is retained in a relay buffer only for a duration  $\tau_e$ , called the *buffer expiry time-out*, and then dropped. However, we assume that ACKs are never dropped to make room for other packets or ACKs, since ACKs are much smaller than packets and they contain valuable information. The duration  $\tau_e$  depends on several factors such as the number of simultaneous flows, the buffer capacity at the relays etc. We view the buffer expiry time-out  $\tau_e$  as a constraint imposed by such external factors, and for our purpose, we assume that  $\tau_e$  is a given network parameter.

We consider coding only at the source. The source generates RLCs of the information packets it wishes to send to the destination. The RLCs are generated over  $\mathbb{F}_q$ , where  $\mathbb{F}_q$  denotes the Galois field of size  $q$  [125, 126]. The random coefficients used to generate an RLC, called the *encoding vector*, is included in the header of the coded packet.

### 3.3 Our Proposed Reliable Transport Scheme

The objective is to transfer a file consisting of  $M$  information packets from the source to the destination in a reliable manner. Upon meeting relays, the source generates RLCs of the  $M$  information packets. The relays carry and replicate the RLCs as detailed in Section 3.3.2. To recover the  $M$  information packets, the destination needs to receive  $M$  DoFs, i.e., the rank of the matrix, formed by accumulating the encoding vectors of the received RLCs, must be equal to  $M$ .

The ACKs generated by the destination and received at the source provide the source with the necessary information to evaluate the progress of the transfer. *With coding, the objective of the destination is to meet  $M$  (DoFs) by accumulating “any”  $M$  linearly independent RLCs, and not the reception of specific information packets.* Accordingly, the ACKs in our scheme indicate the number of DoFs missing at the destination, and not the receipt of specific information packets. In particular, as in [122], ACK  $l$  indicates that there are  $l$  DoFs missing at the destination. To ensure reliability in presence of packet drops due to buffer expiry time-out, our scheme: (1) evaluates the progress of the transfer at appropriate intervals, called cycles, using the feedback information provided by the ACKs, and then (2) takes corrective actions accounting for the feedback information provided by the ACKs. Thus, the  $M$  information packets are transferred from the source to the destination in a reliable manner over multiple *cycles*.

### 3.3.1 Algorithm

Our scheme involves several timers, namely, the *spreading time*,  $\tau_{i,S}$ , the *waiting time*,  $\tau_{i,W}$ , the *cycle timeout*,  $\tau_i$ , and the *buffer expiry timeout*,  $\tau_e$ , each of which is a function of the missing DoF,  $i$ , from the point of view of the source indicated by the ACKs (which might be different from the actual DoF at the destination). In Section 3.3.3, we show how the timers can be implemented. Computation of timeout values corresponding to minimum transfer delay is described in Section 3.4.3. Recall that each relay has 1 buffer space, which might be occupied by either an RLC, or an ACK or it might be unoccupied or “empty”. Our scheme is detailed as follows:

- **Initialization:**  $i \leftarrow M$ .
- **While**  $i > 0$ ,
  - A new cycle begins with  $i$  missing DoFs (as viewed by the source). The source sends  $M_i$  RLCs back to back, where  $M_i$  is a function of  $i$ . Each time an empty relay meets the source, the source gives a new RLC to the relay until  $M_i$  RLCs have been sent.
  - Each RLC is spread among the relays until  $t_{M_i} + \tau_{i,S}$ , according to the replication scheme described in Section 3.3.2, where  $t_{M_i}$  is time required for  $M_i$  meetings of the source with  $M_i$  relays and  $\tau_{i,S}$  is called the *spreading time*.
  - Each time a relay meets the destination, the destination sends an ACK informing the source how many DoFs are still needed to recover the  $M$  information packets. (The ACK generation and replication scheme is detailed in Section 3.3.2.)
  - After spreading of the  $M_i$ -th RLC, the source waits further for a duration  $\tau_{i,W}$ , called the *waiting time*, i.e., the source waits for ACKs from  $t_{M_i} + \tau_{i,S}$  to  $t_{M_i} + \tau_{i,S} + \tau_{i,W}$ . The purpose of the waiting time is to allow the ACKs to reach the source.
  - Replication of the RLCs stops during the ACK-wait phase. However, replication of the ACKs continues throughout the cycle. Furthermore, a copy of an RLC is retained in a relay buffer only for a duration  $\tau_e$ , called the *buffer expiry time*, whereas a copy of an ACK is retained in the relay buffer until the end of the cycle.
  - The cycle lasts for a total duration

$$\tau_i := t_{M_i} + \tau_{i,S} + \tau_{i,W}.$$

At the end of the cycle: (i) all the relays drop the copy of the RLC or ACK they have, and (ii) the source considers the minimum of the missing DoFs indicated by all the ACKs it has received during the cycle. Let the minimum of the missing DoFs indicated by the ACKs be  $j$ .

– **Update:**  $i \leftarrow j$ .

• **End While.**

### 3.3.2 Random Linear Coding (RLC) and ACK Replication

For the sake of brevity, we call a cycle, which begins with  $i$  missing DoFs, an  $i$ -cycle. For example, the first cycle begins with  $M$  missing DoFs, i.e., the first cycle is an  $M$ -cycle. In a  $i$ -cycle, the source sends up to  $M_i$  RLCs back to back. Each time an empty relay meets the source, the source gives a new RLC to the relay until  $M_i$  RLCs have been sent in the current  $i$ -cycle. Note that each of the  $M_i$  RLCs is spread by the source only once. The source makes use of the first  $M_i$  transmission opportunities to send  $M_i$  different RLCs.

In any  $i$ -cycle, we index the RLCs by  $k$ ,  $k = 1, 2, \dots, M_i$ , and index the ACKs by  $l$ ,  $l = 0, 1, \dots, i-1$ . Note that, *ACK  $l$  contains the information that the destination still needs  $l$  DoFs to recover the  $M$  information packets.* When a relay with a copy of RLC  $k$  meets with an empty relay (before  $t_{M_i} + \tau_{i,S}$ ), the empty relay gets a copy of RLC  $k$ . Thus, each RLC, after being sent by the source, is spread until  $t_{M_i} + \tau_{i,S}$ , after which it is not replicated any more. Furthermore, a copy of an RLC is retained in a relay buffer only for a buffer expiry time-out of duration  $\tau_e$ . An empty relay, which had an RLC (or an ACK) earlier but dropped the RLC (or the ACK) due to buffer time-out or cycle time-out, is allowed to receive, carry and spread another RLC (or even the same RLC) or ACK subsequently, i.e., we allow *re-infection*. This is desirable from the point of view of implementation, since the relays do not have to remember the history of the RLCs and the ACKs they have already carried. This also allows for better delay performance with smaller number of relays.

When two nodes, which have different RLCs, meet, then there is no exchange.

When the destination receives an RLC, it recomputes the rank of the decoding matrix, then updates the missing DoFs, generates an ACK indicating the missing DoF, and the RLC in the relay gets replaced with the latest ACK. When the destination is in a state with  $l$  missing DoFs,  $l = 0, 1, \dots, i-1$ , it gives ACK  $l$  to all the relays it meets, be they empty or not, except to those who already have ACK  $l$ . Note that, upon meeting an empty relay, the destination does not generate an ACK if it is in a state with  $i$  missing DoFs, i.e., if it has not received any useful RLC in the current  $i$ -cycle. When a relay with ACK  $l$  meets an empty relay, the empty relay gets a copy of ACK  $l$ . When a relay with ACK  $l$  meets another relay with ACK  $l'$ ,  $l' > l$ , then ACK  $l'$  is replaced by ACK  $l$ , since an

ACK indicating a smaller number of missing DoFs provides more recent and more accurate information to the source. When a relay with ACK 0 meets with another relay with an RLC, then RLC in the relay gets replaced with ACK 0, since ACK 0 indicates complete reception of the file and no more RLCs are required to reach the destination.

Replication of the RLCs occur only during the RLC-spread phase, whereas replication of the ACKs continues throughout the cycle.

At the end of each cycle, the relays drop the copy of the RLC or ACK they have. This choice is to maintain the stability of the network as follows. Dropping the RLCs at the end of each cycle makes room for the RLCs and the ACKs of the same flow in subsequent cycles as well as for other flows. More importantly, the relays would not know when the transfer is complete. Thus, if the RLCs and the ACKs are not dropped at the end of each cycle, they would remain in the network, unnecessarily, for a very long time. The RLCs would ultimately be dropped due to buffer expiry, but the ACKs will not. The ACKs of the flow under consideration must be dropped, ultimately. Due to these practical considerations, we propose to drop the RLCs and the ACKs from the relay buffers at the end of each cycle.

We call our reliable transport proposal described above as Scheme 1 which can be summarized as follows:

- **Scheme 1:**

1. There is no contention between ACKs. The more important ACKs replace less important ACKs.
2. The contention between RLCs and ACKs exists with the exception that final ACK can kill RLCs.

**Two alternative schemes:** We also propose two alternative schemes, namely, Scheme 2 and Scheme 3. These alternative schemes are:

- **Scheme 2:**

1. There is no contention between ACKs. The more important ACKs replace less important ACKs. The same feature as present in Scheme 1.
2. Full contention between RLCs and ACKs exists. In contrast to Scheme 1, ACK indicating 0 DoF missing at destination (i.e., the final ACK) can not replace RLCs.

- **Scheme 3:**

1. There is contention between ACKs at relay nodes. The more important ACKs replace less important ACKs.
2. Full contention between RLCs and ACKs exists as in Scheme 2.

In fact, Scheme 1 and Scheme 2 contain a few additional features than Scheme 3. In Schemes 1 and 2, ACKs indicating the more updated missing DoF at the destination may replace those indicating less important missing DoFs. Further, ACK indicating 0 (final ACK) DoF missing at destination may replace all other ACKs (in Schemes 1 and 2) as well as RLCs (in Scheme 1 only).

Scheme 3 might represent situations in which such replacements may not be allowed from security point of view. It may happen that some malicious nodes may generate RLCs and ACKs at their own. The relay nodes do not have any interest in knowing the authenticity of RLCs/ACKs (origin of RLCs/ACKs) and forward the same upon meeting with other nodes. But, we consider that destination and source nodes can distinguish between original and malicious RLCs/ACKs. Thus, if we allow replacement of less important ACKs with more important ones and replacement of RLCs and all other ACKs by final ACK, then it is possible that malicious nodes may generate final ACK indicating 0 degree of freedom missing at destination which can replace all RLCs and other ACKs upon meetings. This situation may deny arrival of required RLCs at destination even after infinite delay. To avoid such situations, we assume contention between RLCs and ACKs as well as among ACKs too.

### 3.3.3 Implementation Issues

The nodes can implement the timers in our reliable transport scheme in a distributed manner as follows. First, the source and the destination must agree on the values of the number of information packets,  $M$ , and the coding field size,  $q$ , by a handshaking mechanism. This is similar to the “connection set-up” phase of TCP in which basic variables are exchanged and agreed upon. A final “connection release” mechanism is required in which the source informs the destination to clear all the variables corresponding to the flow under consideration. In the remainder of the chapter, we shall not discuss the connection set-up and release mechanisms any further.

The cycle time-out  $\tau_i$  and the spreading time  $\tau_{i,S}$  are included in each RLC generated by the source, and are kept, as is, in every copy of the RLC. The buffer expiry time-out  $\tau_e$  is generated afresh by each relay at the time of receiving a copy of the RLC, since  $\tau_e$  is local to each relay. Each RLC, after sent by source, is spread till duration  $\tau_{i,S}$ , and is dropped from

the relay buffer at the earliest of the time-outs  $\tau_i$  and  $\tau_e$ . Since the destination generates ACKs only after receiving an RLC in the current cycle, the cycle time-out  $\tau_i$  is copied into the destination's buffer and subsequently included in the ACKs as well.

### 3.4 Analytical Modeling and Optimization

Our reliable transport proposal is organized in cycles. In this Section, first, we model the network dynamics of our proposal in one cycle (Section 3.4.1). Next, in Section 3.4.2, we derive the expression for expected overall delay by combining the cycles together. Finally, in Section 3.4.3, we optimize the mean time to transfer the complete file which depends on certain parameters chosen in an optimal manner.

#### 3.4.1 Modeling the Network Dynamics in One Cycle

We model the dynamics of an  $i$ -cycle to study the spreading of the  $M_i$  RLCs and the  $i$  ACKs. We can model the network dynamics of any  $i$ -cycle independent of the earlier cycles, since the relays drop the copy of the RLC or ACK they have at the end of each cycle. Thus, we reset the time variable  $t$  to zero in the beginning of each cycle and study the network dynamics in an  $i$ -cycle for  $t \in [0, \tau_i]$ .

To model the drop of RLCs in the relay buffers due to expiry, we approximate the constant buffer expiry time-out  $\tau_e$  by an exponentially distributed time-out with the same mean  $\tau_e$ . We shall see that this is a very good approximation in the sense that the resulting analytical model provides accurate predictions of the file transfer delay. Note, however, that *in the simulations, we actually use a constant buffer expiry time-out*. Let  $\beta_e := 1/\tau_e$  denote the rate of expiry.

Our model is based on the fluid-limit approach. We consider a sequence of networks indexed by the number of relays,  $N$ , in the DTN. We scale the number of sources and destinations as well as the pairwise meeting rates appropriately in the sequence of networks, let  $N$  to infinity in a certain sense, and obtain a fluid-limit model.

Let  $n_s^{(N)}$  and  $n_d^{(N)}$  denote the number of sources and destinations, respectively. Let  $X_k^{(N)}(t)$ ,  $k = 1, 2, \dots, M_i$ , denote the number of relays that have a copy of RLC  $k$  at time  $t$ . Let  $Y_l^{(N)}(t)$ ,  $l = 0, 1, \dots, i-1$ , denote the number of copies of ACK  $l$  at time  $t$ . The superscript  $N$  emphasizes the fact that the evolution of the  $X_k^{(N)}(t)$ 's and  $Y_l^{(N)}(t)$ 's depend on the total number of relays,  $N$ , in the network. Our specific network contains  $N = N_0$  relays with  $n_s^{(N_0)} = 1$ ,  $n_d^{(N_0)} = 1$ .

When the number of relays,  $N$ , is small enough, a numerical approach is possible. However, when  $N$  is large, an explicit characterization of the transient behavior is difficult due to the presence of non-linear and non-homogeneous transition rates [124].

To obtain numerical results and gain insights when  $N$  is large, we analyze the network in the limit, as  $N \rightarrow \infty$ . To that end, as in [88], we require that, as  $N \rightarrow \infty$ :

1. The pairwise meeting rates  $\beta_r, \beta_s$  and  $\beta_d$  scale down with  $N$  such that the quantities  $\lambda_r := N\beta_r$ ,  $\lambda_s := N\beta_s$ , and  $\lambda_d := N\beta_d$ , remain constant,
2. The number of sources  $n_s^{(N)}$  and the number of destinations  $n_d^{(N)}$  scale up with  $N$  such that the ratios  $s := n_s^{(N)}/N$  and  $d := n_d^{(N)}/N$  remain constant, and
3. The initial values  $X_k^{(N)}(0)$  and  $Y_l^{(N)}(0)$  scale up with  $N$  such that the ratios  $x(0) := \frac{X_k^{(N)}(0)}{N}$  and  $y(0) := \frac{Y_l^{(N)}(0)}{N}$  remain constant.

The above *fluid-limit* scaling amounts to increasing the number of sources, destinations and relays to infinity by increasing the area of the network, but keeping their densities (in number per unit area) constant [88]. The number of sources,  $n_s^{(N)}$ , and the number of destinations,  $n_d^{(N)}$ , are both equal to 1 in our specific network; they, however, must scale up with  $N$ , as  $N \rightarrow \infty$ , so that the ratios  $n_s^{(N)}/N$  and  $n_d^{(N)}/N$  remain equal to their corresponding values in the actual network.

Let  $S_l$  denotes the random time at which the missing DoF at the destination changes from  $l+1$  to  $l$ ,  $\forall l, l = 0, 1, \dots, i-1$ , with the convention that  $S_i = 0$ . Note that  $[S_{l+1}, S_l]$  denotes the time interval during which the destination remains at state  $l+1$ , *i.e.*, with missing DoF =  $l+1$ .

In order to obtain the dynamic equations for ACKs, we first condition on  $S_l = s_l, \forall l, l = 0, 1, \dots, i-1$ . We then solve the dynamic equations for ACKs together with that of the RLCs. Finally, we uncondition over all possible values of the random variable  $S_l, l = 0, 1, \dots, i-1$  using their joint density function.

Applying Theorem 3.1 of [127], we observe that, for large  $N$  (Theorem 5.9.1 of Appendix 5.9.3),  $\forall k, k = 1, 2, \dots, M_i, \forall l, l = 0, 1, \dots, i-1$ , the expectations  $E(X_k^{(N)}(t))$  and  $E(Y_l^{(N)}(t))$  are well-approximated by  $Nx_k(t)$  and  $Ny_l(t)$ , respectively, where  $x_k(t)$  and  $y_l(t)$ , are the unique solution of the Ordinary Differential Equations (ODEs) (3.1) and (3.2).

$$\frac{dx_k(t)}{dt} = \begin{cases} (s\lambda_s q_k(t) + \lambda_r x_k(t))(1 - x(t) - y(t)) - d\lambda_d x_k(t) \\ -\beta_e x_k(t) - \lambda_r x_k(t)y_0(t) \\ \text{for } 0 < t \leq \frac{M_i}{\lambda_s} + \tau_{i,S}, \\ \\ -d\lambda_d x_k(t) - \beta_e x_k(t) - \lambda_r x_k(t)y_0(t) \\ \text{for } \frac{M_i}{\lambda_s} + \tau_{i,S} < t \leq \tau_i. \end{cases} \quad (3.1)$$



$$\frac{dy_l(t)}{dt} = \begin{cases} 0 & \text{for } 0 \leq t < s_l, \text{ and } 0 \leq l \leq i-1, \\ (d\lambda_d + \lambda_r y_l(t))(1 - x(t) - y(t)) + d\lambda_d x(t) \\ + \lambda_r y_l(t) \sum_{m>l} y_m(t) - \lambda_r y_l(t) \sum_{m<l} y_m(t) & \text{for } s_l < t \leq s_{l-1}, \text{ and } 1 \leq l \leq i-1, \\ (d\lambda_d + \lambda_r y_l(t))(1 - y_l(t)) & \text{for } s_0 < t \leq \tau_i, \text{ and } l = 0, \end{cases} \quad (3.2)$$

with initial conditions,  $\forall k = 1, 2, \dots, M_i$ ,  $x_k(0) = 0$  and  $\forall l = 0, 1, \dots, i-1$ ,  $y_l(0) = 0$  for  $0 \leq t < s_l$ , and where  $x(t) := \sum_{k=1}^{M_i} x_k$ ,  $y(t) := \sum_{l=1}^i y_l$  and  $q_k(t)$  denotes the probability that source has *exactly*  $k-1$  meetings with relays up to time  $t$ , given by

$$q_k(t) = \text{Poisson}(\lambda, k-1) = e^{-\lambda_s t} \frac{(\lambda_s t)^{k-1}}{(k-1)!},$$

To understand Equations (3.1) and (3.2), the quantities  $x_k(t)$  and  $y_l(t)$  may be interpreted as the *mean fraction* of relays that have a copy of RLC  $k$  and ACK  $l$ , respectively. With this interpretation, the quantity  $x(t) = \sum_{k=1}^{M_i} x_k$  and  $y(t) = \sum_{l=1}^i y_l$  denote the mean fraction of relays that have a copy of the RLCs and the ACKs, respectively, and  $1 - x(t) - y(t)$  denotes the mean fraction of relays that are empty. Note that the mean fraction of relays that have a copy of RLC  $k$  is equal to zero until the  $k$ th meeting of the source with any relay occurs *i.e.*, until the first copy of  $k$ -th RLC appears in the network. The term  $s\lambda_s q_k(t)(1 - x(t) - y(t))$  corresponds to forwarding of the RLCs by the source. Forwarding of RLCs by source happens during time  $0 < t \leq \frac{M_i}{\lambda_s} + \tau_{i,S}$  where  $\frac{M_i}{\lambda_s} = t_{M_i}$  denotes the mean time equal to  $M_i$  meetings of the source with relays. In order to control that source spreads RLC  $k$  only once, we make use of  $q_k(t)$  which denotes that  $k$ -th meeting of source with an empty relay occurs at time  $t$  with some probability which is exponentially distributed. The terms  $x_k(t)(1 - x(t) - y(t))$  and  $y_l(t)(1 - x(t) - y(t))$  correspond to the spreading of the RLCs and the ACKs, respectively, to empty relays. The term  $d\lambda_d x_k(t)$  corresponds to the replacement of an RLC by an ACK at the destination. The term  $\beta_e x_k(t)$  corresponds to the dropping of the RLCs due to buffer expiry time-out. The term  $\lambda_r y_l(t) \sum_{m>l} y_m(t)$  and  $\lambda_r y_l(t) \sum_{m<l} y_m(t)$  correspond to the replacement of older ACKs by newer ACKs. The term  $d\lambda_d(1 - y_l(t))$  and  $\lambda_r y_l(t)(1 - y_l(t))$ , for  $l = 0$ , correspond to the replacement of RLCs and ACKs at relays by ACK 0 when relays carrying some RLCs or ACK, other than ACK 0, meet with destination and relays, respectively, with ACK 0.

It remains to obtain the  $P_{X_k}(t)$ 's and the  $P_{Y_l}(t)$ 's, which denote the probability that the destination and source has received RLC  $k$  and ACK  $l$ , respectively, by time  $t$ . We denote the Cumulative Distribution Function (CDF) associated with the delay of ACK  $l$ ,

conditioned upon  $S_{i-1} = s_{i-1}, S_{i-2} = s_{i-2}, \dots, S_0 = s_0$ , by  $P_{Y_l}(t, s_{i-1}, s_{i-2}, \dots, s_0)$ . As in [28, 128],  $P_{X_k}(t)$ 's and  $P_{Y_l}(t)$ 's are given by

$$\begin{aligned} \frac{dP_{X_k}(t)}{dt} &= \lambda_d x_k(t)(1 - P_{X_k}(t)) \\ \frac{dP_{Y_l}(t, s_{i-1}, s_{i-2}, \dots, s_0)}{dt} &= \lambda_s y_l(t)(1 - P_{Y_l}(t, s_{i-1}, s_{i-2}, \dots, s_0)) \end{aligned} \quad (3.3)$$

with initial conditions  $P_{X_k}(0) = 0$  and  $P_{Y_l}(0, s_{i-1}, s_{i-2}, \dots, s_0) = 0$ , where  $x_k(t)$  and  $y_l(t)$  are obtained by solving Equation (3.1) and (3.2) respectively.

The unconditional CDF  $P_{Y_l}(t)$  is obtained by un-conditioning as

$$P_{Y_l}(t) = \int_{s_{i-1}=0}^t \int_{s_{i-2}=s_{i-1}}^t \dots \int_{s_0=s_{i-1}}^t f_{S_{i-1} \dots S_0}(s_{i-1}, \dots, s_0) P_{Y_l}(t, s_{i-1}, \dots, s_0) ds_{i-1}, \dots, ds_0$$

where  $f_{S_{i-1} \dots S_0}(s_{i-1}, \dots, s_0)$  denotes the joint density of the random variables  $S_l$ ,  $\forall l$ ,  $l = 0, 1, \dots, i-1$ , which we derive in Appendix 3.9.1.

### Network Dynamics of Scheme 2 and Scheme 3:

Equations (3.4) and (3.5) jointly provide the dynamics for Scheme 2, while Equations (3.4) and (3.6) jointly give the dynamics of Scheme 3. Note that Equation (3.4) is common for Scheme 2 and Scheme 3 since there is complete contention between RLCs and ACKs in both schemes. The only difference is that, in Scheme 3, there is contention between ACKs too, whereas in Scheme 2, the more important ACKs can replace less important ACKs.

$$\frac{dx_k(t)}{dt} = \begin{cases} (s\lambda_s q_k(t) + \lambda_r x_k(t))(1 - x(t) - y(t)) - d\lambda_d x_k(t) \\ -\beta_e x_k(t) \\ \text{for } 0 < t \leq \frac{M_i}{\lambda_s} + \tau_{i,S}, \\ \\ -d\lambda_d x_k(t) - \beta_e x_k(t) \\ \text{for } \frac{M_i}{\lambda_s} + \tau_{i,S} < t \leq \tau_i. \end{cases} \quad (3.4)$$

$$\frac{dy_l(t)}{dt} = \begin{cases} 0 & \text{for } 0 \leq t < s_l, \text{ and } 0 \leq l \leq i-1, \\ \\ (d\lambda_d + \lambda_r y_l(t))(1 - x(t) - y(t)) + d\lambda_d x(t) \\ + \lambda_r y_l(t) \sum_{m>l} y_m(t) - \lambda_r y_l(t) \sum_{m<l} y_m(t) \\ \text{for } s_l < t \leq s_{l-1}, \text{ and } 1 \leq l \leq i-1, \\ \\ d\lambda_d(1 - y_l(t)) + \lambda_r y_l(t)(1 - x(t) - y_l(t)) \\ \text{for } s_0 < t \leq \tau_i, \text{ and } l = 0, \end{cases} \quad (3.5)$$

$$\frac{dy_l(t)}{dt} = \begin{cases} 0 & \text{for } 0 \leq t < s_l, \text{ and } 0 \leq l \leq i-1, \\ (d\lambda_d + \lambda_r y_l(t))(1 - x(t) - y(t)) + d\lambda_d x(t) \\ & \text{for } s_l < t \leq s_{l-1}, \text{ and } 1 \leq l \leq i-1, \\ (d\lambda_d + \lambda_r y_l(t))(1 - x(t) - y(t)) + d\lambda_d x(t) \\ & \text{for } s_0 < t \leq \tau_i, \text{ and } l = 0, \end{cases} \quad (3.6)$$

### 3.4.2 Combining the Cycles Together

We now proceed with the description of the sequence of cycles. Let  $\Delta_n$  denote the number of DoFs missing at the destination in the beginning of the  $n$ -th cycle. It is easy to see that,  $\{\Delta_n, n \geq 1\}$  is a Markov chain with state space  $\{0, 1, 2, \dots, M\}$ . The Markov chain  $\{\Delta_n, n \geq 1\}$  begins with  $\Delta_1 = M$ , and gets absorbed in state 0. Let  $P_{ij}$  denote the transition probability from state  $i$  to state  $j$ .

As in [122], the transition probabilities can be expressed in terms of the *erasure probabilities* as seen by the source. In our context, given that a cycle begins with  $i$  missing DoFs (from the point of view of the source), the erasure probabilities correspond to  $(1 - P_{Y_l}(\tau_i))$ ,  $l = 0, 1, 2, \dots, i-1$ , i.e., the probability that the source has not received ACK  $l$  by the end of the  $i$ -cycle. Thus, the transition probabilities  $P_{ij}$ ,  $j = 0, 1, \dots, i-1$ , are given by

$$P_{ij} = P_{Y_j}(\tau_i) \prod_{l=0}^{j-1} (1 - P_{Y_l}(\tau_i)), \quad (3.7)$$

and

$$P_{ii} = 1 - \sum_{j=0}^{i-1} P_{ij}.$$

Let  $T_i$ ,  $i = 1, 2, \dots, M$ , denote the expected time to reach the state with 0 missing DoFs, starting from the beginning of an  $i$ -cycle. Clearly,  $T_M$  represents the expected completion time for the transfer of the whole file. Let  $E_{Direct}(T_{i \rightarrow 0})$  denotes the expected time to reach state 0 directly in the same cycle, starting from  $i$  missing DoFs. Then,

$$T_i = \sum_{j=1}^i P_{ij}(\tau_i)(T_j + \tau_i) + P_{i0}(\tau_i)E_{Direct}[T_{i \rightarrow 0}], \quad (3.8)$$

where (See Appendix 3.9.2)

$$E_{Direct}[T_{i \rightarrow 0}] = \tau_i - \int_0^{\tau_i} \frac{P_{i0}(t)}{P_{i0}(\tau_i)} dt, \quad (3.9)$$

which leads to

$$P_{i0}(\tau_i)E_{Direct}[T_{i \rightarrow 0}] = \tau_i P_{i0}(\tau_i) - \int_0^{\tau_i} P_{i0}(t)dt,$$

Indeed, by putting the value of  $P_{i0}(\tau_i)E_{Direct}[T_{i \rightarrow 0}]$  in Equation 3.8 and simplifying, we get,

$$T_i = \frac{\sum_{j=1}^{i-1} P_{ij}T_j + \tau_i - \int_0^{\tau_i} P_{i0}(t)dt}{1 - P_{ii}(\tau_i)}. \quad (3.10)$$

### 3.4.3 Optimization Procedure

Our objective is to minimize the mean time to transfer the complete file, i.e., the time until receiving an ACK indicating 0 missing DoFs. In our transport scheme, the overall delay depends on the parameters  $\{M_i, \tau_{i,S}, \tau_{i,W}\}$ ,  $i = 1, \dots, M$ , which are chosen in an optimal way as follows. We minimize the mean completion time for the transmission of all the  $M$  packets, i.e., to minimize  $T_M$ . We can see from Equation 3.10 that the optimization can be recursive.

Since  $T_i$ ,  $i = 2, \dots, M$ , depends only on  $j = 1, \dots, i-1$ , we can perform a three-dimensional optimization (over the parameters  $\{M_i, \tau_{i,S}, \tau_{i,W}\}$ ) at each step  $i$ , starting at  $i = 1$ , and then substituting  $T_j$ , for  $j = 1, \dots, i-1$ , by the values obtained in the previous steps as

$$\min(T_i) = \frac{\sum_{j=1}^{i-1} P_{ij} \min(T_j) + \tau_i - \int_0^{\tau_i} P_{i0}(t)dt}{1 - P_{ii}(\tau_i)}.$$

We thereby obtain the optimal values of  $\{M_i, \tau_{i,S}, \tau_{i,W}\}$ ,  $i = 1, \dots, M$ , in a recursive manner. It is important to note that number of equations change with  $M_i$ . So, we need to optimize the number of equations as well which makes the optimization procedure difficult to solve.

For each step  $i$ , the three-dimensional optimization of the parameters  $\{M_i, \tau_{i,S}, \tau_{i,W}\}$  pertains to the class of nonlinear optimization problems. Many general algorithms for solving such problems have been developed. We experimented with an algorithm called Differential Evolution (DE) [129]. DE is a robust optimizer for multivariate functions. We do not describe DE here, but only say that this algorithm is in part a hill climbing algorithm and in part a genetic algorithm.

The above optimization procedure does not have to be performed at the source node, but is rather performed offline, and the resulting optimal parameters for each possible  $i$ -cycle,  $i = 1, 2, \dots, M$ , are stored in memory at the source node, so as to be used as needed. Note that our optimization procedure requires the knowledge of  $\lambda_r = N\beta_r$ ,  $\beta_s$ ,  $\beta_d$  and  $\beta_e$  which can be estimated by using the history of node meetings.

### 3.5 Approximation 1

We make an approximation of our proposed model in order to overcome the complexity of optimal computation. In the first approximation we propose, we obtain the most likely values of  $S_l$ ,  $l = 0, 1, \dots, i-1$  and then solve the set of ODEs as before. Note that no un-conditioning is required in this case, and hence, we need to solve the set of ODEs only once, unlike the case of the model, which needs to be solved for several different values of  $s_l$ ,  $l = 0, 1, \dots, i-1$  (in principle, an uncountably infinite values, but to be able to numerically compute we solve for a large finite number of values to get sufficiently accurate results), which then need to be averaged. The most likely values  $s_l^*$ ,  $l = 0, 1, \dots, i-1$ , are, essentially, the maximum likelihood estimates (modes) of appropriately chosen density functions. More precisely, we follow the sequential procedure given below.

- At the beginning, there is no contention with ACK, so we have

$$s_{i-1}^* = \arg \max_{s_{i-1}} f_{S_{i-1}}(s_{i-1}).$$

- Then for the second ACK, ACK  $i-2$ , the most likely value of  $S_{i-2}$  depends on  $S_{i-1}$ . Hence, we have

$$s_{i-2}^* = \arg \max_{s_{i-2}} f_{S_{i-2}|S_{i-1}}(s_{i-2}|s_{i-1}^*),$$

and so on.

This simplified approximation model still allows to get accurate results for various values of expiry times.

### 3.6 Approximation 2

We propose another approximation of our model, which is also based on the fluid-limit approach. This can be seen as an approximation to the model described in Section 3.4.1 as follows. The approach of conditioning on ACK generation times is replaced with the approach of using probabilities of ACK generation.

As before, we model the dynamics of an  $i$ -cycle to study the spreading of the  $M_i$  RLCs and the  $i$  ACKs, and approximate the constant buffer expiry time-out  $\tau_e$  by an exponentially distributed time-out with the same mean  $\tau_e$ .  $X_k^{(N)}(t)$ ,  $k = 1, 2, \dots, M_i$ , denote the number of relays that have a copy of RLC  $k$  at time  $t$  and  $Y_l^{(N)}(t)$ ,  $l = 0, 1, \dots, i-1$ , denote the number of relays that have a copy of ACK  $l$  at time  $t$ .

Let  $Q_l^{(i)}(t)$  denote the probability that the number of missing DoFs at the destination at time  $t$  is  $l$ . Note that,  $Q_l^{(i)}(t)$  represents the probability that the destination has received

$i - l$  DoFs in the current  $i$ -cycle by time  $t$ . To keep our model simple, we assume that the reception of any RLC decreases the number of missing DoFs at the destination by one. Then,  $Q_l^{(i)}(t)$  is given

$$Q_l^{(i)}(t) = \sum_{E \subset \{1, \dots, M_i\} : |E|=i-l} \prod_{m \in E} P_{X_m}(t) \prod_{m' \in \{1, \dots, M_i\} \setminus E} (1 - P_{X_{m'}}(t)). \quad (3.11)$$

In reality, reception of an RLC which is linearly independent of the previously received RLCs decreases the missing DoFs by one, and hence, Equation (3.11) is an approximation.

Applying Theorem 3.1 of [127], we observe that, for large  $N$  (Theorem 5.9.1 of Appendix 5.9.3),  $\forall k, k = 1, 2, \dots, M_i, \forall l, l = 0, 1, \dots, i - 1$ , the expectations  $E(X_k^{(N)}(t))$  and  $E(Y_l^{(N)}(t))$  are well-approximated by  $Nx_k(t)$  and  $Ny_l(t)$ , respectively, where  $x_k(t)$  and  $y_l(t)$ , are the unique solution of the ODEs of Equations (3.12) and (3.13).

$$\frac{dx_k(t)}{dt} = \begin{cases} 0 & \text{for } 0 \leq t \leq \frac{k-1}{\lambda_s}, \\ (s\lambda_s + \lambda_r x_k(t))(1 - x(t) - y(t)) - d\lambda_d x_k(t) \\ -\beta_e x_k(t) - \lambda_r x_k(t)y_0(t) & \text{for } \frac{k-1}{\lambda_s} < t \leq \frac{k}{\lambda_s}, \\ \lambda_r x_k(t)(1 - x(t) - y(t)) - d\lambda_d x_k(t) \\ -\beta_e x_k(t) - \lambda_r x_k(t)y_0(t) & \text{for } \frac{k}{\lambda_s} < t \leq \frac{M_i}{\lambda_s} + \tau_{i,S}, \\ -d\lambda_d x_k(t) - \beta_e x_k(t) - \lambda_r x_k(t)y_0(t) & \text{for } \frac{M_i}{\lambda_s} + \tau_{i,S} < t \leq \tau_i. \end{cases} \quad (3.12)$$

$$\begin{aligned} \frac{dy_l(t)}{dt} &= \lambda_r y_l(t)(1 - x(t) - y(t)) + d\lambda_d Q_l^{(i)}(t)(1 - y_l(t)) \\ &\quad + \lambda_r y_l(t) \sum_{m>l} y_m(t) - \lambda_r y_l(t) \sum_{m<l} y_m(t) \\ &\quad + 1_{\{l=0\}} \lambda_r y_l(t)x(t), \text{ for } 0 < t \leq \tau_i, \end{aligned} \quad (3.13)$$

with initial conditions,  $\forall k = 1, 2, \dots, M_i, x_k(0) = 0$ , and  $\forall l = 0, 1, \dots, i - 1, y_l(0) = 0$ , where  $x(t) := \sum_{k=1}^{M_i} x_k$ ,  $y(t) := \sum_{l=1}^i y_l$  and  $Q_l^{(i)}(t)$  is given by Equation (3.11).

Equations (3.12) and (3.13) represent our Scheme 1. Approximation 2 differs slightly from the model in Section 3.4.1 as follows. In model, we capture the spreading of RLCs by the source for duration  $\tau_{i,S}$  by probabilities  $q_k(t)$ . Here, we explicitly write different evolution equations for durations  $\frac{k-1}{\lambda_s} \leq t \leq \frac{k}{\lambda_s}$  and  $\frac{k}{\lambda_s} \leq t \leq \frac{k}{\lambda_s} + \tau_{i,S}$ . The term in

the evolution equation which captures the spreading by the source is restricted to just 1 intermitting interval.

To understand Equations (3.12) and (3.13), the term  $s\lambda_s(1 - x(t) - y(t))$  corresponds to forwarding of the RLCs by the source, and this forwarding occurs for a mean duration  $1/\lambda_s$ , i.e., for a duration equal to one inter-meeting time between the source and any relay, and the mean number of RLCs forwarded by the source in that duration is equal to one. Note that the mean fraction of relays that have a copy of RLC  $k$  is equal to zero until the  $k$ th meeting of the source with any relay, which roughly occurs in the interval  $[(k-1)/\lambda_s, k/\lambda_s]$ , at which time the first copy of RLC  $k$  appears in the network. The term  $d\lambda_d Q_l^{(i)}(t)(1 - y_l(t))$  corresponds to the spreading of the ACKs by the destination which captures the replacement of approach of conditioning on ACK generation times with the approach of using probabilities of ACK generation. The term  $1_{\{l=0\}}\lambda_r y_l(t)x(t)$  correspond to the replacement of the RLCs by ACK 0. The other quantities and terms correspond to the same meaning as explained earlier for Equations (3.1) and (3.2) in Section 3.4.1.

The  $P_{X_k}(t)$ 's and the  $P_{Y_l}(t)$ 's, which denote the probability that the destination and source has received RLC  $k$  and ACK  $l$ , respectively, by time  $t$ , as in [28, 128], are given by

$$\begin{aligned}\frac{dP_{X_k}(t)}{dt} &= \lambda_d x_k(t)(1 - P_{X_k}(t)) \\ \frac{dP_{Y_l}(t)}{dt} &= \lambda_s y_l(t)(1 - P_{Y_l}(t))\end{aligned}\tag{3.14}$$

with initial conditions  $P_{X_k}(0) = 0$  and  $P_{Y_l}(0) = 0$ , where  $x_k(t)$  and  $y_l(t)$  are obtained by solving Equations (3.12) and (3.13).

**Dynamics of Scheme 2 and Scheme 3:** Equation (3.16) provides the dynamics of Scheme 2, while in Equation (3.17), the dynamics of Scheme 3 are shown. Again,  $dx_k(t)$  given in (3.15) is same for Scheme 2 and 3 as there is complete contention between RLCs and ACKS in both schemes.

$$\frac{dx_k(t)}{dt} = \begin{cases} 0 & \text{for } 0 \leq t \leq \frac{k-1}{\lambda_s}, \\ (s\lambda_s + \lambda_r x_k(t))(1 - x(t) - y(t)) - d\lambda_d x_k(t) \\ -\beta_e x_k(t) & \text{for } \frac{k-1}{\lambda_s} < t \leq \frac{k}{\lambda_s}, \\ \lambda_r x_k(t)(1 - x(t) - y(t)) - d\lambda_d x_k(t) \\ -\beta_e x_k(t) & \text{for } \frac{k}{\lambda_s} < t \leq \frac{M_i}{\lambda_s} + \tau_{i,S}, \\ -d\lambda_d x_k(t) - \beta_e x_k(t) & \text{for } \frac{M_i}{\lambda_s} + \tau_{i,S} < t \leq \tau_i. \end{cases}\tag{3.15}$$

$$\begin{aligned}
\frac{dy_l(t)}{dt} &= \lambda_r y_l(t)(1 - x(t) - y(t)) + d\lambda_d Q_l^{(i)}(t)(1 - y_l(t)), \\
&\quad + \lambda_r y_l(t) \sum_{m>l} y_m(t) - \lambda_r y_l(t) \sum_{m<l} y_m(t) \\
&\quad \text{for } 0 < t \leq \tau_i,
\end{aligned} \tag{3.16}$$

$$\begin{aligned}
\frac{dy_l(t)}{dt} &= \lambda_r y_l(t)(1 - x(t) - y(t)) + d\lambda_d Q_l^{(i)}(t)(1 - y_l(t)), \\
&\quad \text{for } 0 < t \leq \tau_i,
\end{aligned} \tag{3.17}$$

### 3.7 Validation of the Analytical Model and Performance of the Optimal Procedure

To solve the ODE model, and in the optimal procedure, we take  $M = 5$ ,  $N = 100$ ,  $\beta = 0.05$ , and  $\tau_e$  is taken from 0.20 to 1 with interval of 0.10, from 1 to 2 with interval of 0.20 and from 2 to 10 with interval of 1. In the simulations, in addition to the above setting of parameters, we took Galois field of size  $q = 1$ , i.e., we generated RLCs with *binary* random coefficients. Random inter-meeting intervals were generated using exponential distributions. Simulation results have been averaged over 1000 runs.

Notice that we have not specified the units of  $\beta$ 's and  $\tau_e$ . Assuming that  $\beta = \beta_s = \beta_d = \beta_r$ , one can think of  $\tau_e = 2$  as  $\tau_e = 2 \text{ sec}$ , in which case,  $\beta = 0.05$  has to be  $\beta = 0.05 \text{ sec}^{-1}$ . Similarly, one can think of  $\tau_e = 2$  as  $\tau_e = 2 \text{ min}$  (resp.  $\tau_e = 2 \text{ hr}$ ), in which case,  $\beta = 0.05$  has to be  $\beta = 0.05 \text{ min}^{-1}$  (resp.  $\beta = 0.05 \text{ hr}^{-1}$ ).

*Remark 3.7.1.* In simulations, we set  $t_{M_i}$  to be a fixed quantity equal to its expected value, i.e., we set  $t_{M_i} = \frac{M_i}{\lambda_s}$  and not the actual  $M_i$ -th meeting in a particular simulation run. However, due to randomness of the inter-meeting times, the source might not meet with  $M_i$  empty relays within  $\frac{M_i}{\lambda_s}$ .  $\square$

#### 3.7.1 Proposed Model

Figures 3.1 and 3.2 correspond to Scheme 1 and Scheme 3, respectively, of our proposal. In these figures, we compare mean file transfer times provided by the optimal procedure with mean file transfer times obtained by simulations performed by using the optimal settings for missing DoFs equal to 1. We plot the mean file transfer times against various expiry times. It can be observed that the mean file transfer times from simulations, under the optimal settings of the parameters  $M_i$ ,  $\tau_{i,S}$  and  $\tau_{i,W}$ , are in good agreement with the optimal mean file transfer times for both schemes. This validates our overall procedure of minimization



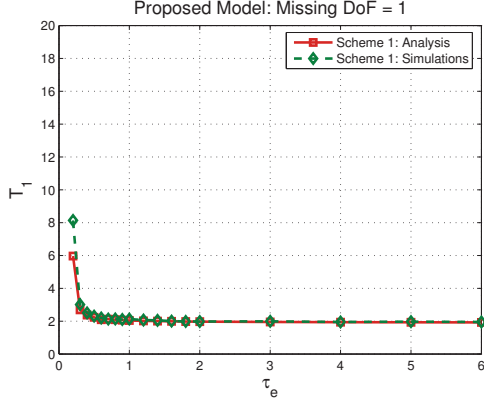


Figure 3.1: Scheme 1: Optimal values of  $T_1$  against  $\tau_e$  and comparison with simulation when missing DoF = 1.

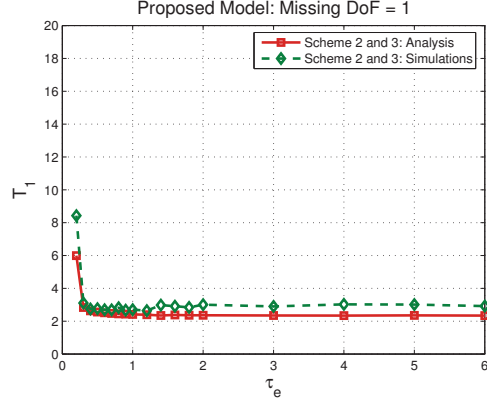


Figure 3.2: Scheme 2 and Scheme 3: Optimal values of  $T_1$  against  $\tau_e$  and comparison with simulation when missing DoF = 1.

of mean file transfer time based on our proposed model in Section 3.4.1 for missing DoF = 1. In other words, it validates our ODE model near the optimal settings where the network will operate for missing DoF = 1. Note, however, that our analytical model is also accurate in other regimes, but since it is not possible to show the accuracy of our model in the entire parameter space, we have chosen the optimal settings to demonstrate the accuracy of our model. Moreover, mean file transfer times of Scheme 1 are obviously less than those of Scheme 3 for all values of  $\tau_e$ . This benefit comes from the following additional feature of Scheme 1 as compare to Scheme 3: when a relay  $i$  carrying final ACK meets with a relay  $j$  carrying some RLC, relay  $j$  replaces its RLC by the final ACK allowing more relays to carry the final ACK.

*Remark 3.7.2.* When missing DoF is equal to 1, then there is no difference between Scheme 2 and Scheme 3 as there is only one final ACK in the network, so there is no possibility of replacing less important ACKs by the final ACK. Thus, Scheme 2 and 3 are identical when missing DoF is 1.

*Remark 3.7.3.* As stated earlier, due to complexity of optimal computation of the model, we provide results only for missing DoF equal to 1 for the model described in Section 3.4.1. For higher missing DoFs, we obtain results with our two proposed approximations. These approximations eliminate the complexity of computation of the optimal setting of parameters to a large extent and enable us to obtain results without compromising their accuracy too much.

### 3.7.2 Approximation 1

In Figures, 3.3-3.7, we provide results for Scheme 3 under Approximation 1 of our proposed model. In these figures, we compare mean file transfer times obtained through simulations under the optimal settings with mean file transfer times obtained analytically at the optimal settings for missing DoF  $i = 1, 2, 3, 4, 5$ , respectively.

We observe that the mean file transfer times from simulations, under the optimal settings of the parameters  $M_i$ ,  $\tau_{i,S}$  and  $\tau_{i,W}$ , are in good agreement with the optimal mean file transfer times. Further, we observe that Approximation 1 provides quite accurate results with respect to our proposed model. Again, these results validate minimization of mean file transfer time under Approximation 1 for all missing DoFs. We also observe that mean file transfer time increases for increases missing DoFs (Figures 3.3-3.7).

### 3.7.3 Approximation 2

In Figures 3.8-3.12, we provide results for Scheme 3 obtained by Approximation 2 of our proposed model. Again, we compare the mean file transfer time from simulation under the optimal settings with the mean file transfer time obtained analytically at the optimal settings for initial missing DoF  $i = 1, 2, 3, 4, 5$ , at the start of transfer (Figures 3.8-3.12 against various expiry times).

Again, we observe that the mean file transfer times from simulations, under the optimal settings of the parameters  $M_i$ ,  $\tau_{i,S}$  and  $\tau_{i,W}$ , are in good agreement with the optimal mean file transfer times. This validates our overall procedure of minimization of mean file transfer time using Approximation 2 based on our fluid-limit model. In other words, it validates our ODE model near the optimal settings where the network will operate.

*Remark 3.7.4.* Further, we make experiments for Scheme 1 and 2 and obtain analysis and simulation results (not reported here as we come across similar trends). We observe from results that mean file transfer time of Scheme 2 is less as compared to that of Scheme 3 for all pair of missing DoFs and  $\tau_e$ . This decrease actually comes from the return path as the more important ACKs replace the less important ACKs enabling ACKs reach back to the source quickly in Scheme 2 compared to scheme 3 where there is complete contention in the network. Moreover, we also observe that mean transfer time of file of Scheme 1 is less as compared to that of Scheme 2 for any same values of missing DoF and  $\tau_e$ . This reduction comes when we allow final ACK replacing RLCs too. So the final ACK, once generated, travels back to the source faster in Scheme 1 than in Scheme 2. Further in Scheme 2 and Scheme 3,  $T_i$  increases as reverse path is more blocking in these schemes due to finite number of relays. This fact is more prominent in simulations.

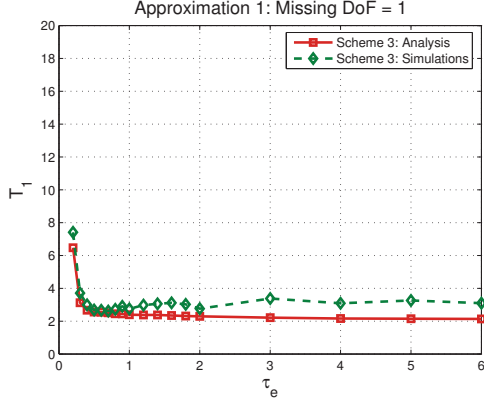


Figure 3.3: Scheme 3: Optimal values of  $T_1$  against  $\tau_e$  and comparison with simulation when missing DoF = 1.

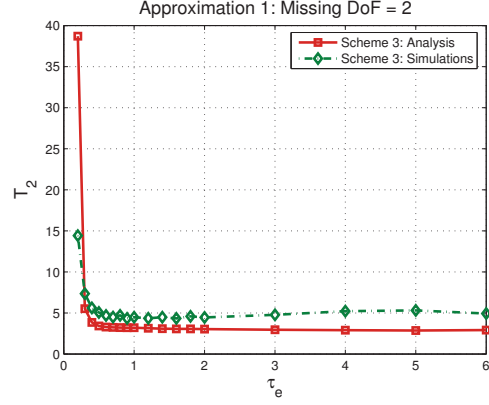


Figure 3.4: Scheme 3: Optimal values of  $T_2$  against  $\tau_e$  and comparison with simulation when missing DoF = 2.

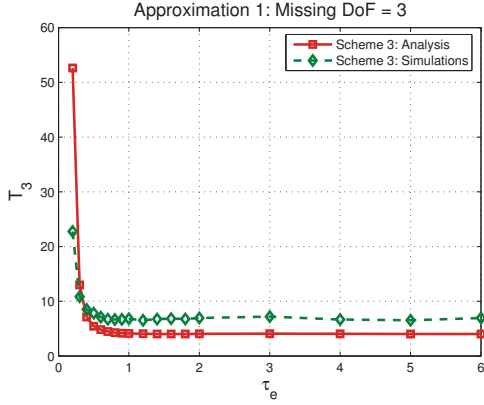


Figure 3.5: Scheme 3: Optimal values of  $T_3$  against  $\tau_e$  and comparison with simulation when missing DoF = 3.

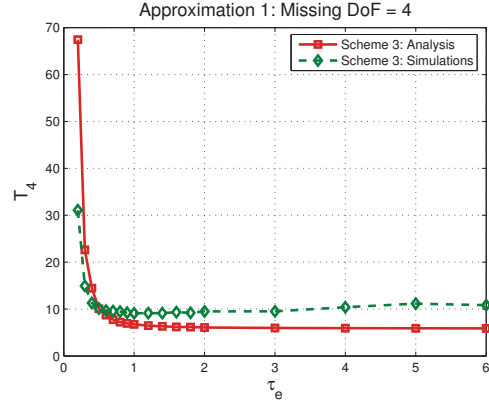


Figure 3.6: Scheme 3: Optimal values of  $T_4$  against  $\tau_e$  and comparison with simulation when missing DoF = 4.

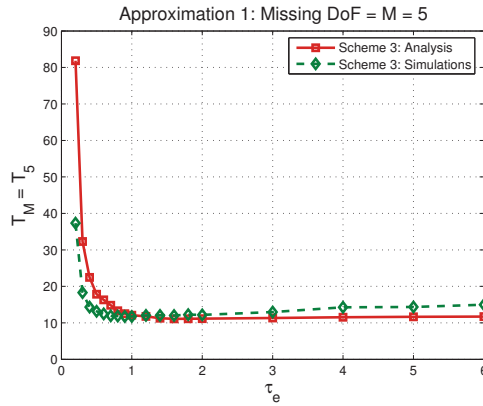


Figure 3.7: Scheme 3: Optimal values of  $T_M$  against  $\tau_e$  and comparison with simulation when missing DoF = M = 5.

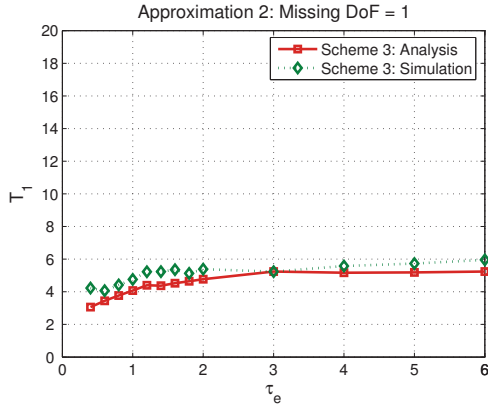


Figure 3.8: Scheme 3: Optimal values of the file transfer time  $T_1$  against  $\tau_e$  and comparison with simulation when missing DoF = 1.

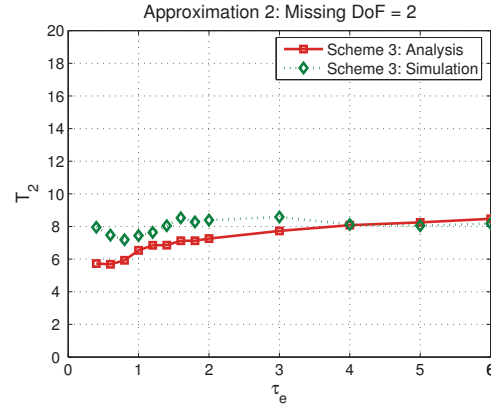


Figure 3.9: Scheme 3: optimal values of the file transfer time  $T_2$  against  $\tau_e$  and comparison with simulation when missing DoF = 2.

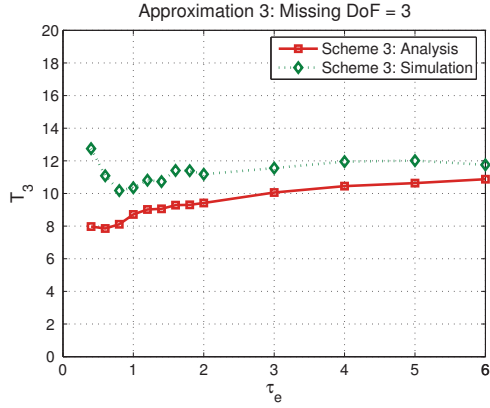


Figure 3.10: Scheme 3: optimal values of the file transfer time  $T_3$  against  $\tau_e$  and comparison with simulation when missing DoF = 3.

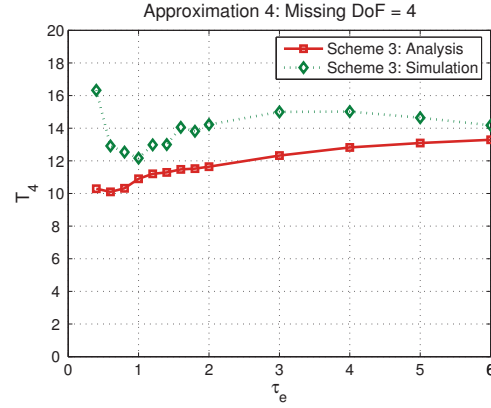


Figure 3.11: Scheme 3: optimal values of the file transfer time  $T_4$  against  $\tau_e$  and comparison with simulation when missing DoF = 4.

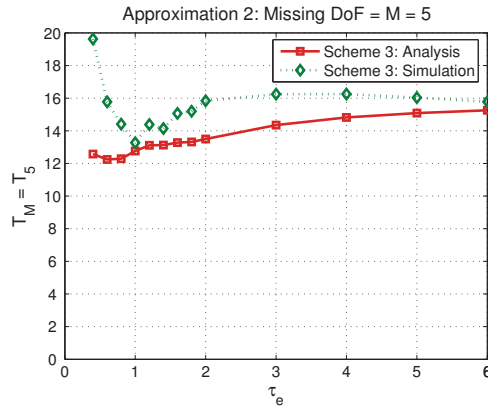


Figure 3.12: Scheme 3: optimal values of the file transfer time  $T_M$  against  $\tau_e$  and comparison with simulation when missing DoF = M = 5.

### 3.8 Conclusion of the Chapter

We have proposed a new reliable transport scheme for DTNs based on the use of ACKs and coding. We have modeled the evolution of the network under our scheme using a fluid-limit approach. We obtained mean file transfer times and showed, by comparing numerical analysis versus simulations, the validity of our models. Using our fluid-limit model, we optimized the number of outstanding RLCs to be sent back to back before the expiration of a time-out as well as the value of the time-out itself, so as to minimize the file transfer time. We accounted for the buffer expiry time-out, quantified its impact on the optimal values of our protocol parameters and also demonstrated the adaptability of our optimal procedure to variations in expiry time-out.

We have proposed two approximations of our model to overcome its computation complexity. Proposed approximation optimize mean file transfer times for any missing DoFs fast enough.

### 3.9 Appendices

#### 3.9.1 Derivation of $f_{S_{i-1}, \dots, S_0}(s_{i-1}, \dots, s_0)$

In this appendix, we derive the joint probability density function of the random variables  $S_l$ ,  $l = 0, 1, \dots, i-1$ . We can write

$$\begin{aligned}
 & f_{S_{i-1}, \dots, S_0}(s_{i-1}, \dots, s_0) \\
 &= f_{S_{i-1}}(s_{i-1}) f_{S_{i-2}, \dots, S_0 | S_{i-1}}(s_{i-2}, \dots, s_0 | s_{i-1}) \\
 &= f_{S_{i-1}}(s_{i-1}) f_{S_{i-2} | S_{i-1}}(s_{i-2} | s_{i-1}) f_{S_{i-3}, \dots, S_0 | S_{i-1}, S_{i-2}}(s_{i-3}, \dots, s_0 | s_{i-1}, s_{i-2}) \\
 &\vdots \\
 &= f_{S_{i-1}}(s_{i-1}) f_{S_{i-2} | S_{i-1}}(s_{i-2} | s_{i-1}) \dots f_{S_0 | S_{i-1}, \dots, S_1}(s_0 | s_{i-1}, \dots, s_1)
 \end{aligned}$$

We have

$$\begin{aligned}
 & f_{S_{i-1}}(s_{i-1}) \Delta s_{i-1} \\
 &\approx P(\text{ACK } i-1 \text{ is generated for the first time between } s_{i-1} \text{ and } s_{i-1} + \Delta s_{i-1}) \\
 &= \sum_{k_1=1}^{M_i} P \left( \begin{array}{l} \text{RLC } k_1 \text{ is not received up to } s_{i-1}, \\ \text{RLC } k_1 \text{ is received in } [s_{i-1}, s_{i-1} + \Delta s_{i-1}], \\ \text{no other RLC is received up to } s_{i-1} + \Delta s_{i-1} \end{array} \right) \\
 &= \sum_{k_1=1}^{M_i} (1 - P_{X_{k_1}}(s_{i-1})) (P_{X_{k_1}}(s_{i-1} + \Delta s_{i-1}) - P_{X_{k_1}}(s_{i-1})) \prod_{k_2 \neq k_1} (1 - P_{X_{k_2}}(s_{i-1} + \Delta s_{i-1})) \\
 &\approx \sum_{k_1=1}^{M_i} (1 - P_{X_{k_1}}(s_{i-1})) f_{X_{k_1}}(s_{i-1}) \Delta s_{i-1} \prod_{k_2 \neq k_1} (1 - P_{X_{k_2}}(s_{i-1})),
 \end{aligned}$$

where, we assume that the time duration  $\Delta s_{i-1}$  is so small that at most one event can occur in that duration. Letting  $\Delta s_{i-1} \rightarrow 0$ , this leads to

$$f_{S_{i-1}}(s_{i-1}) = \sum_{k_1=1}^{M_i} (1 - P_{X_{k_1}}(s_{i-1})) f_{X_{k_1}}(s_{i-1}) \prod_{k_2 \neq k_1} (1 - P_{X_{k_2}}(s_{i-1})),$$

where the density function  $f_{X_k}(s_{i-1})$  of  $X_k$  can be computed by

$$f_{X_k}(s_{i-1}) = \frac{dP_{X_k}(u)}{du} \Big|_{u=s_{i-1}} = \beta_d X_k(s_{i-1})(1 - P_{X_k}(s_{i-1})).$$

Similarly, we have

$$\begin{aligned} & f_{S_{i-1}, S_{i-2}}(s_{i-1}, s_{i-2}) \Delta s_{i-1} \Delta s_{i-2} \\ & \approx P \left( \begin{array}{l} \text{ACK } i-1 \text{ is generated for the first time between } s_{i-1} \text{ and } s_{i-1} + \Delta s_{i-1}, \\ \text{ACK } i-2 \text{ is generated for the first time between } s_{i-2} \text{ and } s_{i-2} + \Delta s_{i-2} \end{array} \right) \\ & = \sum_{k_1=1}^{M_i} P \left( \begin{array}{l} \text{RLC } k_1 \text{ is not received up to } s_{i-1}, \\ \text{RLC } k_1 \text{ is received in } [s_{i-1}, s_{i-1} + \Delta s_{i-1}], \\ \text{no other RLC is received up to } s_{i-2}, \\ \cup_{k_2 \neq k_1} \text{RLC } k_2 \text{ is received in } s_{i-2} + \Delta s_{i-2} \end{array} \right) \\ & = \sum_{k_1=1}^{M_i} (1 - P_{X_{k_1}}(s_{i-1})) (P_{X_{k_1}}(s_{i-1} + \Delta s_{i-1}) - P_{X_{k_1}}(s_{i-1})) \prod_{k_2 \neq k_1} (1 - P_{X_{k_2}}(s_{i-2})) \\ & \quad \times \sum_{k_2=1, k_2 \neq k_1}^{M_i} (P_{X_{k_2}}(s_{i-2} + \Delta s_{i-2}) - P_{X_{k_2}}(s_{i-2})) \\ & = \sum_{k_1=1}^{M_i} (1 - P_{X_{k_1}}(s_{i-1})) f_{X_{k_1}}(s_{i-1}) \Delta s_{i-1} \prod_{k_2 \neq k_1} (1 - P_{X_{k_2}}(s_{i-2})) \sum_{k_2=1, k_2 \neq k_1}^{M_i} f_{X_{k_2}}(s_{i-2}) \Delta s_{i-2}. \end{aligned}$$

Letting  $\Delta s_{i-1}, \Delta s_{i-2} \rightarrow 0$ , this leads to

$$f_{S_{i-1}, S_{i-2}}(s_{i-1}, s_{i-2}) = \sum_{k_1=1}^{M_i} (1 - P_{X_{k_1}}(s_{i-1})) f_{X_{k_1}}(s_{i-1}) \prod_{k_2 \neq k_1} (1 - P_{X_{k_2}}(s_{i-2})) \sum_{k_2=1, k_2 \neq k_1}^{M_i} f_{X_{k_2}}(s_{i-2}).$$

The conditional density  $f_{S_{i-2}|S_{i-1}}(s_{i-2}|s_{i-1})$  is obtained by

$$f_{S_{i-2}|S_{i-1}}(s_{i-2}|s_{i-1}) = \frac{f_{S_{i-1}, S_{i-2}}(s_{i-2}, s_{i-1})}{f_{S_{i-1}}(s_{i-1})}.$$

The general expression for the joint density is given by

$$\begin{aligned} & f_{S_{i-1}, \dots, S_0}(s_{i-1}, \dots, s_0) \\ & = \sum_{k_1=1}^{M_i} (1 - P_{X_{k_1}}(s_{i-1})) f_{X_{k_1}}(s_{i-1}) \prod_{k_2 \neq k_1} (1 - P_{X_{k_2}}(s_{i-2})) \sum_{k_2=1, k_2 \neq k_1}^{M_i} f_{X_{k_2}}(s_{i-2}) \\ & \quad \times \dots \sum_{k_0=1, k_0 \neq k_1, k_0 \neq k_2, \dots, k_0 \neq k_{M_i-1}}^{M_i} (1 - P_{X_{k_0}}(s_0)) f_{X_{k_0}}(s_0) \end{aligned}$$

### 3.9.2 Derivation of $E_{Direct}[T_{i \rightarrow 0}]$

Let  $X$  be a random variable with density function  $f_X(x)$ . Let the range of  $X$  be 0 to  $b$ . Then the expectation of  $X$  is given by

$$E[X] = \int_0^b x f_X(x) dx = \int_{x=0}^{x=b} \int_{t=0}^{t=x} dt dF_X(x), \quad (3.18)$$

Equation 3.18 can be rewritten as

$$\begin{aligned} E[X] &= \int_{t=0}^{t=b} \left( \int_{x=t}^{x=b} dF_X(x) \right) dt \\ &= \int_{t=0}^{t=b} (F_X(b) - F_X(t)) dt \\ &= \int_{t=0}^{t=b} F_X(b) dt - \int_{t=0}^{t=b} F_X(t) dt \\ &= b F_X(b) - \int_{t=0}^{t=b} F_X(t) dt, \end{aligned}$$

Now mapping of  $E_{Direct}[T_{i \rightarrow 0}]$  with last part of above Equation is done as under:

$$E_{Direct}[T_{i \rightarrow 0}] = \int_0^{\tau_i} t f_{T_{i \rightarrow 0}}(t) dt = \tau_i F_{T_{i \rightarrow 0}}(\tau_i) - \int_0^{\tau_i} F_{T_{i \rightarrow 0}}(t) dt, \quad (3.19)$$

In Equation 3.19,  $F_{T_{i \rightarrow 0}}(t)$  is equal to  $\frac{P_{i0}(t)}{P_{i0}(\tau_i)}$  because  $T_{i \rightarrow 0}$  is a conditional random variable on the fact that transfer completes within  $\tau_i$ .

Equation 3.19 can be rewritten as

$$E_{Direct}[T_{i \rightarrow 0}] = \tau_i - \int_0^{\tau_i} \frac{P_{i0}(t)}{P_{i0}(\tau_i)} dt, \quad (3.20)$$

and we get by simplification

$$P_{i0}(\tau_i) E_{Direct}[T_{i \rightarrow 0}] = \tau_i P_{i0}(\tau_i) - \int_0^{\tau_i} P_{i0}(t) dt,$$

# Enhancements to Reliable Transport

## Contents

<b>4.1</b>	<b>Network Setting</b>	<b>72</b>
<b>4.2</b>	<b>Fundamental Limitations and their Consequences</b>	<b>73</b>
<b>4.3</b>	<b>Our Proposed Enhancements</b>	<b>75</b>
<b>4.4</b>	<b>Performance Evaluation Methodology</b>	<b>79</b>
4.4.1	Simulation Settings	79
4.4.2	Performance Metrics	79
<b>4.5</b>	<b>Improving the Return Path</b>	<b>84</b>
4.5.1	Benefits of Selective ACKs (SACKs) over ACKs	84
4.5.2	Benefits of Global-ACK (G-ACK) over ACK	86
4.5.3	Benefits of G-SACK over SACK with Single Destination	88
4.5.4	Benefits of G-SACK over SACK with Multiple Destinations	90
<b>4.6</b>	<b>Improving the Forward Path</b>	<b>92</b>
<b>4.7</b>	<b>Benefits of overall Proposed Scheme over Basic Scheme</b>	<b>96</b>
<b>4.8</b>	<b>Conclusion of the Chapter</b>	<b>100</b>



Designing appropriate ACK mechanisms in DTNs is still an open issue [130]. In this Chapter, we propose a new set of enhancement schemes to improve reliable transport, both for unicast and multicast flows, in DTNs. Reliability is ensured through the use of new type of acknowledgements which contain detailed (and potentially global) information about the receipt of packets at all the destinations improving the return component of reliable transport. The motivation for using novel acknowledgment mechanisms comes from the observation that one should take the maximum advantage of the contact opportunities which occur quite infrequently in DTNs. We also allow for random linear coding at the relay nodes improving forward component of reliable transport.

We propose enhanced ACK mechanism to improve reliable transport for DTNs which is based on a novel Global Selective ACKnowledgment (G-SACK) scheme. A G-SACK can potentially contain *global* information about the receipt of packets at each destination in the network. We also allow the use of random linear coding [122] of packets from different sources so as to share the “packet payload space” between packets of different flows in the network.

## 4.1 Network Setting

We consider a DTN consisting of  $S + D + N$  mobile nodes. There are  $S$  *source* nodes,  $D$  *destination* nodes and  $N$  *relay* nodes (see Figure 4.1). We allow for both *unicast* (one source to one destination) and *multicast* (one source to multiple destinations) data transfer. Also, a node can be a destination for multiple sources. The sources send packets to their destinations through the relays, and the destinations send back acknowledgments, for every received packet, to their corresponding sources through the relays. We study packet transfer with and without network coding of packets, and with three different acknowledgment schemes – a simple *baseline* scheme and two enhancements to it (Section 4.3). Packets and acknowledgments are transferred through the relays as in epidemic routing [17] (with slight modifications, the details of which will be provided later).

Each source-destination pair  $(i, j)$ ,  $i = 1, \dots, S$ ,  $j = 1, \dots, D$ , defines a *flow*. A multicast flow  $(i, \mathcal{J})$  consists of multiple flows with the same source and multiple (different) destinations, i.e.,

$$(i, \mathcal{J}) := \{(i, j') : j' \in \mathcal{J}, \mathcal{J} \subset \{1, \dots, D\}, |\mathcal{J}| \geq 2\}.$$

The *flow matrix*  $A = [a_{ij}]$  is an  $S \times D$  matrix, where, for all  $i$  and  $j$ ,  $i = 1, \dots, S$ ,  $j = 1, \dots, D$ , the entry  $a_{ij} = 1$ , if source  $i$  has a packet to send to destination  $j$ ; otherwise,  $a_{ij} = 0$ . For example, the flow matrix corresponding to the network in Figure 4.2, which

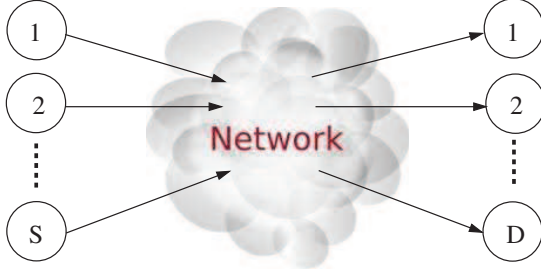


Figure 4.1: The network of  $N$  mobile relays (shown as the cloud) connecting the  $S$  sources and the  $D$  destinations.

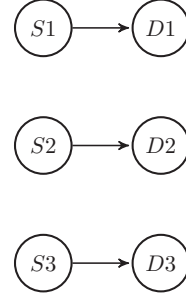


Figure 4.2: Three unicast flows.

consists of three unicast flows, is given by

$$A_U = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

We denote the number of (pairwise) flows by  $N_{pw}$ . Clearly,  $N_{pw} = \sum_{i=1}^S \sum_{j=1}^D a_{ij}$ .

Two nodes are said to “meet” when they come within the *communication range* of each other. We assume that the successive inter-meeting times of any two nodes are i.i.d. exponential random variables with mean  $1/\beta$ . Our assumption of i.i.d. exponential inter-meeting times is motivated by [88, 123, 124], in which the authors provide simulation results to show that, for *random waypoint* mobility models such as the *random direction* and *random walk* mobility models, the assumption of i.i.d. exponential inter-meeting times provides extremely accurate approximations for actual inter-meeting times.

## 4.2 Fundamental Limitations and their Consequences

Let flow  $(i, j)$ ,  $i = 1, \dots, S$ ,  $j = 1, \dots, D$ , consists of  $N_{ij}$  packets. The unicast flow  $(i, j)$  is said to be *complete* when the acknowledgments for all the  $N_{ij}$  packets reach source  $i$ . A multicast flow  $(i, \mathcal{J})$  is said to be complete when all the constituent flows  $(i, j')$ ,  $j' \in \mathcal{J}$ , are complete. From the performance perspective, it is desirable that (i) each flow is complete with as small a delay as possible, and (ii) there is fairness across the (pairwise) flows with respect to *throughput* (i.e., number of packets transferred per unit time). If the packet buffering and scheduling policy at the relays treat all flows equally, then one can expect to achieve *long-term* fairness (i.e., over a sufficiently large number of packet transfers for each flow). However, it is desirable to achieve *short-term* fairness as well (i.e., over a few packet transfers).

There are fundamental limitations that need to be addressed before one could achieve the desirable properties. The two primary limitations are *randomness* and *finite capacity*. Randomness refers to the randomness in the meeting/contact process pertaining to the mobility of the nodes. The *inter-meeting times* (i.e., the time duration between successive meetings of two nodes) and the *contact durations* (i.e., the time duration for which two nodes remain within each other's communication range) are random. Finite capacity refers to a finite number of relays with finite buffer space and the finiteness of the contact durations. Note that *the amount of data that can be exchanged in a meeting is determined by the contact duration*.

The round-trip delay associated with the transfer of a packet consists of a *forward* component (the delay after which the destination(s) receive the packet) and a *return* component (the delay after which the source receives the corresponding acknowledgment(s)). As the capacity (i.e., the number of relays and/or the buffer space at the relays and/or the contact duration) decreases and/or the mean inter-meeting time increases, the expected values of the forward, the return, and hence, the round-trip delay, increases. The randomness in the meeting/contact process, combined with finite capacity, determines the second and higher-order moments of the delays. Randomness makes the network inefficient as follows:

**Inefficiency:** With little or no information about the contact patterns among the nodes, a significant fraction (or percentage) of transmissions gets wasted in forwarding copies of packets (resp. acknowledgments) to those relays that do not contribute toward delivering the packets (resp. the acknowledgments) to the intended destinations (resp. sources).<sup>1</sup> Similarly, of the meetings a destination (resp. source) has with a relay, a significant fraction gets wasted because the packets (resp. the acknowledgments) carried by the relay are not intended for the destination (resp. source).

More importantly, randomness, combined with finite capacity, degrade the performance as follows:

**Non-reachability:** In reality, a source can wait for an acknowledgment only for a finite time  $T$ , and, with positive probability, does not receive the acknowledgment within  $T$ . We call this Type-I non-reachability. Due to randomness and finite capacity, with multiple competing flows, a source may not be able to transfer any of its packets even if it waits for infinite time. With positive probability, a source may never meet with a relay having the required amount of buffer space; indeed, due to the randomness of node meetings, the buffer

---

<sup>1</sup>The order in which relays get copies of a packet from a source can be modeled as a “tree”. Starting with the source as the root, the tree can be constructed as follows. Whenever a relay having the packet meets with a relay that does not have the packet but has buffer space to get a copy of the packet, a new leaf node is added by a link to the existing node which gives the packet. There is a unique path from the source to the destination on this tree, and the links on this path are “contributing”. All links in the tree that do not correspond to a path from the source to one of the destinations are “non-contributing”.

space of every relay that meets with a source might already be completely occupied with packet(s) and acknowledgment(s) of other sources and destinations. We call this Type-II non-reachability, which can be solved by expunging the packet(s) and acknowledgment(s) from the relay buffers using (random or deterministic) *expiry timeouts* so that packets from all sources get access to relay buffers sooner or later. However, if the mean time to expiry is smaller than the mean time between relay meetings, then, with positive probability, the packet(s) will not reach the destination(s).

**Inter-Dependence:** A less severe but more likely problem is that of inter-dependence among the packet transfers, which occurs because they share the same resources (relay buffers and transmission times during contacts). Due to randomness and finite capacity, *in a particular realization of competing packet transfer processes*, one particular packet might get replicated at a much faster rate than others, leaving less resources for others. Due to the epidemic-type replication, small differences during the initial phase (which is unavoidable due to the randomness of node meetings) can become significant over time, since a packet with a larger number of copies replicates at a faster rate. Also, acknowledgments for a fast-spreading packet are generated before the other packets could reach their destinations. This implies that acknowledgments for a fast-spreading packet compete with slow-spreading packets for getting replicated. Similarly, the acknowledgments of different packets also compete for getting access to the finite relay buffers and for getting forwarded within the finite contact durations.

Thus, *one packet transfer being faster automatically implies that other packet transfers being slower*. This inequity results in large delay variance and short-term unfairness.

In the next section, we propose enhancement schemes to address the issues identified in this section.

### 4.3 Our Proposed Enhancements

In Section 4.2 we identified that, with little or no information about the contact patterns among the nodes, randomness makes the network inefficient. Furthermore, there is competition (a) between packets, (b) between acknowledgments, and (c) between packets and acknowledgments, of the same or different flows for getting access to the finite relay buffers and getting replicated within the finite contact durations, which result in longer expected delays, larger delay variances and short-term unfairness.

The obvious solutions to these problems are to: (1) increase the buffer capacity of the relays, (2) increase the transmit powers and/or apply sophisticated physical layer techniques (modulation/coding schemes) to make the communication range larger, thereby, make the contact durations longer, (3) learn the contact patterns of the node meetings (which re-

quires more processing and involves a “learning delay”), and (4) devise appropriate buffer management and scheduling policies for (i) admitting new packets and acknowledgments from other nodes, (ii) transmitting buffered packets and acknowledgments to other nodes, and (iii) expunging buffered packets and acknowledgments using expiry timeouts.

Our goal in this chapter, however, is to improve the performance with the limited available resource, with existing physical layer implementation, without using any information about the contact pattern, and over time-scales finer than the expiry timeouts. We propose and study our enhancements without referring to any buffer management and scheduling policy, since the study of buffer management and scheduling policies is out of the scope of this work. Interested readers may refer to [63, 64].

We avoid addressing the issue of buffer management and scheduling by restricting our study to **one** packet per source (i.e.,  $N_{ij} = 1$  for all  $i, j$ ), and relay buffers with capacity  $B$  to store at the most **one** packet of length  $L$ ,  $L \leq B < 2L$ , or a few acknowledgments that can fit into the buffer capacity  $B$ . We also assume that the contact durations are long enough so that the relays can exchange all the information stored in their respective buffers with one another in a single meeting.

Our enhancements are detailed as follows:

1. **Generation of Selective Acknowledgments at the destination:** Upon receipt of a packet, a destination generates a Selective ACKnowledgment (SACK) indicating the set of sources from which it has already received the packet(s). This is in contrast with the baseline acknowledgment scheme, henceforth called “the ACK scheme”, in which only the currently received packet is acknowledged. A SACK can acknowledge *multiple sources* about the receipt of packets at a *specific destination*.<sup>2</sup> Of course, if a node is a destination for a single source, then it can only generate an ACK.
2. **Updating ACKs/SACKs inside the network to form Global-ACKs/SACKs:** When a node carrying an ACK (or a SACK) generated by a destination, on its way back to the source(s), meets with other nodes carrying ACKs/SACKs generated by other destinations (or more recent SACKs generated by the same destination), the information contained in the ACKs/SACKs are combined to form Global SACKs (G-SACKs). In the special case where all destinations generate only ACKs, we call the combined packet receipt information as Global ACKs (G-ACKs). A G-SACK (resp. G-ACK) scheme is in contrast with the SACK (resp. the ACK) scheme in which the SACKs (resp. ACKs) generated by the destination(s) reach the sources

---

<sup>2</sup>Note that our notion of SACK is slightly different from the traditional one. In our case, SACK(s) from a destination provide information about receipt of packets from “different” sources and not different packets from the same source.

without being updated inside the network. A G-ACK/G-SACK can acknowledge *multiple sources* about the receipt of packets at *multiple destinations*.

3. **Random linear coding of data packets:** Packets are combined at the relays to form random linear combinations. This allows packets from different sources to share the packet payload space.

The key ideas that motivate our proposed enhancements are the following:

- I The competition between packets of different flows can be mitigated by random linear coding of packets at the relays. A coded packet is more likely to contain useful information for each destination.
- II The competition between acknowledgments of different flows can also be mitigated by coding of acknowledgments at the relays. However, since acknowledgments are much smaller than packets, we can combine the acknowledgment information, in uncoded form, efficiently as a matrix (discussed in detail below). This would allow us to store, carry and forward the information about the receipt of packets at a larger number of destinations within the same available buffer space. Thus, a G-SACK is more likely to contain useful information for each source.

In the following we elaborate on the implementation and other aspects of our enhancements. First, we define the *source-degree* and *destination-degree* as follows:

- **Source-degree:** We define the source-degree of a destination as the number of sources for which it is a destination. For example, the source-degree  $d_j^S$  of destination  $j$  is given by  $d_j^S = \sum_{i=1}^S a_{ij}$ . The average source-degree  $d^S$  in the network is defined by

$$d^S = \frac{\sum_{j=1}^D d_j^S}{D} = \frac{\sum_{j=1}^D \sum_{i=1}^S a_{ij}}{D} = \frac{N_{pw}}{D}.$$

- **Destination-degree:** We define the destination-degree of a source as the number of destinations for which it is a source. For example, the destination-degree  $d_i^D$  of source  $i$  is given by  $d_i^D = \sum_{j=1}^D a_{ij}$ . The average destination-degree  $d^D$  in the network is defined by

$$d^D = \frac{\sum_{i=1}^S d_i^D}{S} = \frac{\sum_{i=1}^S \sum_{j=1}^D a_{ij}}{S} = \frac{N_{pw}}{S}.$$

It turns out that the benefits of our enhancements improve with increase in the source degree and/or the destination degree (see Section 4.7). Higher values of source and destination degrees also enable us to implement the G-SACKs in an efficient manner as matrices.

**G-SACK as a matrix:** An ACK consists of the 3-tuple {sourceID, destinationID, receiptFlag}. To specify the packet receipt information for all source-destination pairs, one requires  $\sum_{i=1}^S \sum_{j=1}^D a_{ij} = N_{pw}$  of such 3-tuple entries. This requires  $N_{pw}(2L_I + 1)$  bits, where  $L_I$  denotes the number of bits needed to specify a source ID or a destination ID. However, when the same source needs to send its packet to multiple destinations (i.e., multicast) and/or a node is a destination for multiple sources, one can efficiently represent the G-SACK information as an  $S \times D$  matrix. The matrix representation would avoid the need for repeating the sourceID and the destinationID, which consume more bits than the receiptFlag. For the matrix implementation, one requires a mapping from the source IDs to the row indices, another mapping from the destination IDs to the column indices, and one bit for each entry of the  $S \times D$  matrix. A total of  $(S + D)L_I + SD$  bits are required. The savings with the matrix implementation is given by

$$N_{pw} \left( 2 - \frac{1}{d^D} - \frac{1}{d^S} \right) L_I + \left( N_{pw} - \frac{N_{pw}^2}{d^S d^D} \right),$$

which increases with  $d^S$  and/or  $d^D$ .

**Updating of G-SACKs:** The G-SACK matrix of each relay could be different. Ideally, each relay should contain complete “global” information. However, only “local” information is available initially. The “ideal” *G-SACK matrix*  $G = [g_{ij}]$  is an  $S \times D$  matrix, where,  $\forall i = 1, \dots, S$ , and  $\forall j = 1, \dots, D$ , the entry  $g_{ij} = 1$  if destination  $j$  has already received the packet from source  $i$ ; otherwise,  $g_{ij} = 0$ . However, the G-SACK information at a particular relay node might differ from the ideal G-SACK matrix. For example, even after the packet from source  $i$  has already been received at destination  $j$  (and a corresponding acknowledgment has already been generated), the entry  $g_{ij}$  of the (local) G-SACK matrix  $G_R$  at some relay node ‘R’ may still be equal to 0 if ‘R’ has not yet come in contact with a node having this information.

When all the packets have been received at their intended destinations, and sufficient mixing has been occurred inside the network, the packet receipt information at all the destinations is contained in the G-SACKs. Then, the G-SACK matrix  $G_R$  at a relay node ‘R’ becomes equal to the flow matrix  $A$ . For example, for the network in Figure 4.2 and flow matrix  $A_U$ , a G-SACK matrix  $G_R = A_U$  at a relay node ‘R’ indicates that all the packets have been received at their destinations, and a G-SACK matrix

$$G_R = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

at relay ‘R’ indicates that all the packets, except that the packet from source 1 to destination 1, have been received by their destinations. This may happen due to two reasons: (i) either



destination 1 has not yet received the packet from source 1, (ii) or the acknowledgment for the said packet has not yet been mixed with the above G-SACK inside the network. Clearly, different relays may contain different G-SACKs.

## 4.4 Performance Evaluation Methodology

In this section, we describe the simulation settings and performance metrics.

### 4.4.1 Simulation Settings

We demonstrate the performance benefits due to each enhancement feature of our proposal by constructing appropriate replication/spreading schemes (Sections 4.6-4.7), progressively incorporating one enhancement feature at a time, and then by comparing with the *Basic Scheme* described in the following:

**The Basic Scheme:** When relay  $i$ , which is empty (i.e., relay  $i$  has neither a packet nor an acknowledgment), meets with a source, relay  $i$  gets a copy of the packet from the source. When relay  $j$ , which is empty, meets with relay  $i$ , which has a packet, relay  $j$  gets the packet. When relay  $j$ , which has a packet, meets with a destination (of the packet), the destination gets the packet and the packet in relay  $j$  is replaced with an acknowledgment for the currently received packet, henceforth, called an ACK. When relay  $k$ , which is empty, meets with relay  $j$ , which has an ACK, relay  $k$  gets the ACK. When relay  $k$ , which has an ACK, meets with a source (which is the intended recipient of that ACK), the source gets the ACK.  $\square$

We develop a customized simulator in MATLAB for DTNs of the type discussed in Section 4.1. Given any flow matrix, our simulator can simulate the considered schemes. In particular, we compare the schemes by using the topologies shown in Figures 4.2-4.6.

The simulation setting is as follows: number of relays  $N = 100$ , Galois field size  $q = 4$  (when there is coding at relays), and mean inter-meeting time  $1/\beta = 20$  units of time. We simulate each scheme for  $M = 1000$  times, each run for a duration  $T = 100$  units of time.

### 4.4.2 Performance Metrics

We quantify the gain due to each enhancement feature described in Section 4.3 through the following performance metrics:

**Forward delays:** We denote the forward delay from source  $i$  to destination  $j$  for the  $k$ -th simulation run by  $D_{ij}^f(k)$ . It refers to the delay between the sending of the first copy of the packet from source  $i$  and the receipt of the first copy at destination  $j$ . If destination  $j$  does not receive the packet from source  $i$  within the simulation time  $T$ , then we take



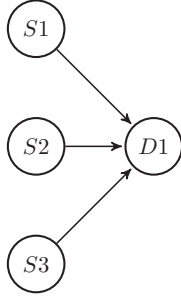


Figure 4.3: Topology1

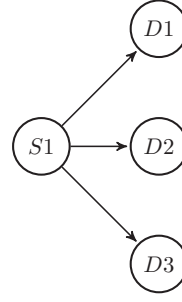


Figure 4.4: Topology2

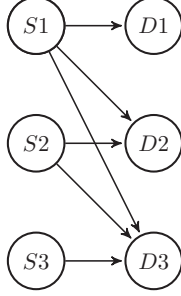


Figure 4.5: Topology 4

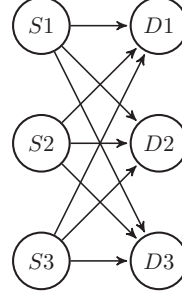


Figure 4.6: Topology 5

$D_{ij}^f(k) = T$ . For  $M$  simulation runs, we obtain a vector

$$\mathbf{D}_{ij}^f = (D_{ij}^f(1), D_{ij}^f(2), \dots, D_{ij}^f(M))$$

of  $M$  samples for the *random* forward delay  $D_{ij}^f$  from source  $i$  to destination  $j$ .

Next, we discard the samples with value  $T$  and obtain a new vector

$$\tilde{\mathbf{D}}_{ij}^f = (\tilde{D}_{ij}^f(1), \tilde{D}_{ij}^f(2), \dots, \tilde{D}_{ij}^f(M_{ij}^f))$$

of  $M_{ij}^f$  samples, where  $M_{ij}^f$ ,  $M_{ij}^f \leq M$ , denotes the number of simulation runs in which the forward path from source  $i$  to destination  $j$  is complete. The expected value  $E[D_{ij}^f]$  and the standard deviation  $\sigma[D_{ij}^f]$  of the random forward delay  $D_{ij}^f$  from source  $i$  to destination  $j$  are computed by

$$E[D_{ij}^f] := \frac{1}{M_{ij}^f} \sum_{k=1}^{M_{ij}^f} \tilde{D}_{ij}^f(k) \quad ; \quad \sigma[D_{ij}^f] = \sqrt{\frac{1}{M_{ij}^f - 1} \sum_{k=1}^{M_{ij}^f} (\tilde{D}_{ij}^f(k) - E[D_{ij}^f(k)])^2}.$$

Clearly,  $E[D_{ij}^f]$  and  $\sigma[D_{ij}^f]$  represent the *sample mean* and the *sample standard deviation*, respectively, computed using the  $M_{ij}^f$  samples corresponding to the simulation runs in which the forward path is complete.

We denote by  $M_{nw}^f$  the number of simulation runs in which the forward paths for *all* the (pairwise) flows in the network are complete within the simulation time  $T$ . Clearly,  $M_{nw}^f \leq M_{ij}^f \leq M$  for all  $i, j$ . Restricting to the  $M_{nw}^f$  simulation runs, we obtain, for each source-destination pair  $(i, j)$ , the vector

$$\hat{\mathbf{D}}_{ij}^f = (\hat{D}_{ij}^f(1), \hat{D}_{ij}^f(2), \dots, \hat{D}_{ij}^f(M_{nw}^f))$$

of  $M_{nw}^f$  samples. We denote the (network-wide) *average* (resp. *maximum*) forward delay for the  $k$ -th simulation run by  $D_{avg}^f(k)$  (resp.  $D_{max}^f(k)$ ), which refers to the average (resp. maximum) of the forward delays of all source-destination pairs (given that the forward path for all the flows in the network is complete within the simulation time  $T$ ), i.e.,

$$D_{avg}^f(k) := \frac{\sum_{i=1}^S \sum_{j=1}^D a_{ij} \hat{D}_{ij}^f(k)}{N_{pw}} \quad ;$$

$$D_{max}^f(k) := \max\{\hat{D}_{ij}^f(k) : a_{ij} = 1, i = 1, \dots, S, j = 1, \dots, D\}.$$

For the  $M_{nw}^f$  simulation runs in which the forward paths for all the (pairwise) flows in the network are complete within the simulation time  $T$ , we obtain the vectors

$$\mathbf{D}_{avg}^f = (D_{avg}^f(1), D_{avg}^f(2), \dots, D_{avg}^f(M_{nw}^f)), \text{ and}$$

$$\mathbf{D}_{max}^f = (D_{max}^f(1), D_{max}^f(2), \dots, D_{max}^f(M_{nw}^f)),$$

each of  $M_{nw}^f$  samples, for the random average forward delay  $D_{avg}^f$  and the random maximum forward delay  $D_{max}^f$ , respectively. The corresponding expected values and standard deviations are computed as for the pairwise forward delays, but now using the  $M_{nw}^f$  samples in  $\mathbf{D}_{avg}^f$  and  $\mathbf{D}_{max}^f$ , respectively.

**Return delays:** We denote the return delay from destination  $j$  to source  $i$  for the  $k$ -th simulation run by  $D_{ij}^r(k)$ . It refers to the delay between the receipt of (the first copy of) the packet from source  $i$  at destination  $j$  and the receipt of (the first copy of) the corresponding acknowledgment at the source  $i$ . Note that  $D_{ij}^r(k)$  is meaningful only if destination  $j$  has received the packet from source  $i$ . Hence, we restrict to the  $M_{ij}^f$  simulation runs in which the forward path from source  $i$  to destination  $j$  is complete within the simulation time  $T$ , and obtain a vector

$$\mathbf{D}_{ij}^r = (D_{ij}^r(1), D_{ij}^r(2), \dots, D_{ij}^r(M_{ij}^f))$$

of  $M_{ij}^f$  samples for the *random* return delay  $D_{ij}^r$  between destination  $j$  and source  $i$ . We take the return delay equal to  $T$  if the return path for the flow under consideration is not complete within  $T$ .

Next, we discard the samples with value  $T$  and obtain a new vector

$$\tilde{\mathbf{D}}_{ij}^r = (\tilde{D}_{ij}^r(1), \tilde{D}_{ij}^r(2), \dots, \tilde{D}_{ij}^r(M_{ij}^r))$$

of  $M_{ij}^r$  samples, where  $M_{ij}^r, M_{ij}^r \leq M_{ij}^f \leq M$ , denotes the number of simulation runs in which the return path from destination  $j$  to source  $i$  is complete (conditioned upon completion of the forward path between the same pair). The expected value  $E[D_{ij}^r]$  and the standard deviation  $\sigma[D_{ij}^r]$  of the random return delay  $D_{ij}^r$  are computed as before, but using the  $M_{ij}^r$  samples.

We denote by  $M_{nw}^r$  the number of simulation runs in which the return paths for *all* the (pairwise) flows in the network are complete within  $T$  (conditioned upon the completion of the forward paths for *all* the (pairwise) flows). Clearly,  $M_{nw}^r \leq M_{ij}^r \leq M_{ij}^f \leq M$  for all  $i, j$ . Restricting to the  $M_{nw}^r$  simulation runs, we obtain, for each source-destination pair  $(i, j)$ , the vector

$$\hat{\mathbf{D}}_{ij}^r = (\hat{D}_{ij}^r(1), \hat{D}_{ij}^r(2), \dots, \hat{D}_{ij}^r(M_{nw}^r))$$

of  $M_{nw}^r$  samples. We denote the (network-wide) *average* (resp. *maximum*) return delay for the  $k$ -th simulation run by  $D_{avg}^r(k)$  (resp.  $D_{max}^r(k)$ ), which refers to the average (resp. maximum) of the return delays of all source-destination pairs, i.e.,

$$D_{avg}^r(k) := \frac{\sum_{i=1}^S \sum_{j=1}^D a_{ij} \hat{D}_{ij}^r(k)}{N_{pw}} ;$$

$$D_{max}^r(k) := \max\{\hat{D}_{ij}^r(k) : a_{ij} = 1, i = 1, \dots, S, j = 1, \dots, D\}.$$

For the  $M_{nw}^r$  simulation runs, similar to forward paths, we obtain the vectors

$$\mathbf{D}_{avg}^r = (D_{avg}^r(1), D_{avg}^r(2), \dots, D_{avg}^r(M_{nw}^r)), \text{ and}$$

$$\mathbf{D}_{max}^r = (D_{max}^r(1), D_{max}^r(2), \dots, D_{max}^r(M_{nw}^r)),$$

each of  $M_{nw}^r$  samples, for the random average return delay  $D_{avg}^r$  and the random maximum return delay  $D_{max}^r$ , respectively. The corresponding expected values and standard deviations are computed as for the pairwise return delays, but now using the  $M_{nw}^r$  samples in  $\mathbf{D}_{avg}^r$  and  $\mathbf{D}_{max}^r$ , respectively.

**Round trip delay:** It is simply equal to the sum of the forward and return delays. Observing that the the round trip path is complete whenever the return path is complete, we obtain  $M_{ij}^{rt} = M_{ij}^r$  samples for the random round trip delay  $D_{ij}^{rt}$  for flow  $(i, j)$ . The corresponding expected value and standard deviation are computed as before using the  $M_{ij}^{rt} = M_{ij}^r$  samples. Similarly, we obtain  $M_{nw}^{rt} = M_{nw}^r$  samples for the random average round trip delay  $D_{avg}^{rt}$  as well as for the random maximum round trip delay  $D_{max}^{rt}$ . The

corresponding expected values and standard deviations are computed as before using the  $M_{nw}^{rt} = M_{nw}^r$  samples.

**Forward success probability:** For a source-destination pair  $(i, j)$ , it refers to the fraction of simulation runs in which the packet reaches from source  $i$  to destination  $j$  within the simulation time  $T$ , and is denoted by  $P_{ij}^f$ . By definition,  $P_{ij}^f := P(D_{ij}^f \leq T)$ , and for  $M$  simulation runs, we compute  $P_{ij}^f$  by

$$P_{ij}^f = \frac{M_{ij}^f}{M}.$$

The network-wide forward success probability  $P^f$  refers to the fraction of simulation runs in which the forward paths for all the flows in the network are successful within simulation time  $T$ , computed by

$$P^f = \frac{M_{nw}^f}{M}.$$

**Return success probability:** For a source-destination pair  $(i, j)$ , it refers to the fraction of simulation runs in which the acknowledgment reaches from destination  $j$  to source  $i$  within the simulation time  $T$  conditioned upon the completion of the forward path. We denote this probability by  $P_{ij}^r$ . By definition,

$$P_{ij}^r := P(D_{ij}^r \leq (T - D_{ij}^f) | D_{ij}^f \leq T) = P(D_{ij}^f + D_{ij}^r \leq T) / P(D_{ij}^f \leq T),$$

and, we compute  $P_{ij}^r$  by

$$P_{ij}^r = \frac{M_{ij}^r}{M_{ij}^f}.$$

The network-wide return success probability  $P^r$  refers to the fraction of simulation runs in which the return path for all flows under consideration is complete within simulation time  $T$  conditioned upon completion of the forward path for all flows, computed by

$$P^r = \frac{M_{nw}^r}{M_{nw}^f}.$$

**Overall success probability:** It refers to the fraction of simulation runs in which the round trip path is complete within the simulation time  $T$ . It is easy to see that the overall success probability is the product of the forward and return success probabilities. For a source-destination pair  $(i, j)$ , by definition, the overall success probability  $P_{ij}^{rt}$  is given by

$$P_{ij}^{rt} := P(D_{ij}^{rt} \leq T) := P(D_{ij}^f + D_{ij}^r \leq T) := P_{ij}^f P_{ij}^r = \frac{M_{ij}^{rt}}{M}.$$

Similarly, the network-wide round trip success probability  $P^{rt}$  is computed by

$$P^{rt} = \frac{M_{nw}^{rt}}{M}.$$

## 4.5 Improving the Return Path

In this section, we show the benefits of selective acknowledgments and mixing of acknowledgment information inside the network which result in improvement in return path.

We adopt the following convention. When the destinations generate acknowledgment information intended for a single source and there is no mixing of this acknowledgment information inside the network, we call it the (plain) ACK Scheme. When the destinations generate acknowledgment information for multiple sources, but there is no mixing of acknowledgment information inside the network, we add the prefix ‘S’ with ACK and call it the “Selective” ACK (SACK) Scheme. If the destinations generate acknowledgment information for a single source and there is mixing of acknowledgment information inside the network, then we add the prefix ‘G’ and call it the “Global” ACK (G-ACK) Scheme. Similarly, if the destinations generate acknowledgment information for multiple sources and there is mixing of acknowledgment information inside the network, then we add the prefixes ‘G’ and ‘S’ and call it the G-SACK Scheme.

### 4.5.1 Benefits of Selective ACKs (SACKs) over ACKs

**The SACK Scheme:** This Scheme differs from Basic Scheme in only one aspect, namely, the destination generates a SACK (instead of an ACK) indicating the set of packets it has successfully received so far.  $\square$

Thus, in the SACK Scheme, if a destination receives packets from multiple sources, then it generates SACKs instead of (plain) ACKs.

In order to observe the benefits of SACKs over ACKs, we consider the example network shown in Figure 4.3 (we call it Topology 1). In this example network, there are three sources and one destination (common to all sources). Each source unicasts its packet to the destination. For this topology, the average source-degree and destination-degree is equal to 3 and 1, respectively. There is no coding at relays.

We observe the benefits of SACK over ACK which lie in the return paths when ACK/SACK travel back from destination to source(s) as the forward paths are identical with SACKs and plain ACKs, which implies that the overall improvement in delay performance is due to the SACK feature alone. Unlike a plain ACK, which is useful only when it meets with the unique source it acknowledges, a SACK serves its purpose when it meets with *any* of the sources it can acknowledge. This is due to the fact a single SACK provides packet receipt information at a *specific destination* to *multiple sources*.

In Figure 4.7, we compare the Cumulative Distribution Functions (CDFs) of the return delay (random variable) in the Basic Scheme and the SACK Scheme between one source/destination pair. In Figure 4.8, we compare the overall (for whole network) max-

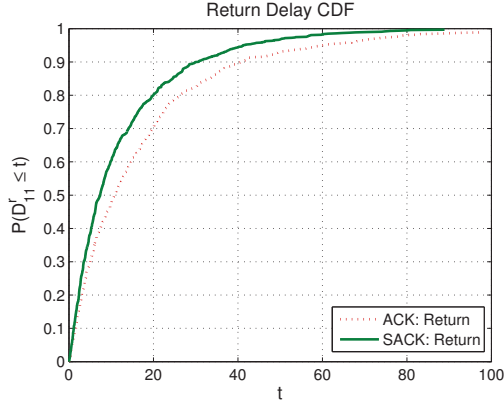


Figure 4.7: Comparison of pairwise return delay CDF of ACK versus SACK: Topology 1.

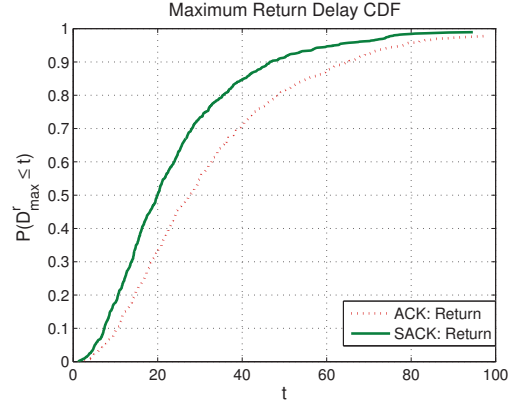


Figure 4.8: Comparison of network-wide maximum return delay CDF of ACK versus SACK: Topology 1.

imum return delay CDFs. Mean delays, success probabilities and corresponding standard deviations alongwith improvement are provided in Table 4.1.

Referring to Figures 4.7-4.8 and Table 4.1, we make the following observations:

- O1: We observe that the delay CDFs with SACKs stay above the delay CDFs with ACKs, implying that the delay with SACKs is *stochastically smaller* than the delay with ACKs. The SACK Scheme improves return delay quite significantly as compared to the Basic Scheme. We observe approximately (25%) improvement in pairwise as well as network wide maximum return and round trip delays even with the simplest Topology shown in Figure 4.3.

Further, we observe that the success probability within in any time budget  $t$  is higher with SACK than ACK. The SACK improves the network wide return success probability resulting in improvement in overall round trip success probability. For example at time  $t = 20$ ,  $P_{11}^f$  with the SACK is 0.76 as compared to 0.58 with the ACK at the same time (see Figure 4.7).

- O2: We observe significant improvement in terms of variance both for return and round trip delays. Benefit of SACK increases with increase of destination-degree. The variance with SACK is not much reduced due to the fact that there is no mixing of ACK/SACK information inside the network. Later, we shall see that by mixing of ACK/SACK information inside the network, maximum delay as well as variance will be much reduced.

Table 4.1: SACK over ACK

Scenario	Forward			Return			Round Trip		
	$P_{11}^f$	$E[D_{11}^f]$	$\sigma[D_{11}^f]$	$P_{11}^r$	$E[D_{11}^r]$	$\sigma[D_{11}^r]$	$P_{11}^{rt}$	$E[D_{11}^{rt}]$	$\sigma[D_{11}^{rt}]$
ACK	0.982	2.3758	4.6167	0.9898	16.4023	17.6114	0.972	18.6968	18.1203
SACK	0.985	2.2712	3.9634	0.9969	12.2160	13.8255	0.982	14.4364	14.8146
Improvement(%)	—	—	—	0.72	25.52	21.50	1.03	22.79	18.24
Scenario	$P^f$	$E[D_{avg}^f]$	$\sigma[D_{avg}^f]$	$P^r$	$E[D_{avg}^r]$	$\sigma[D_{avg}^r]$	$P^{rt}$	$E[D_{avg}^{rt}]$	$\sigma[D_{avg}^{rt}]$
ACK	0.945	2.5167	3.0573	0.9788	16.6185	9.6863	0.925	19.0592	10.1628
SACK	0.945	2.2197	2.1982	0.9894	11.8276	6.9466	0.935	14.0360	7.4399
Improvement(%)	—	—	—	1.08	28.83	28.28	1.08	26.36	26.79
Scenario	$P^f$	$E[D_{max}^f]$	$\sigma[D_{max}^f]$	$P^r$	$E[D_{max}^r]$	$\sigma[D_{max}^r]$	$P^{rt}$	$E[D_{max}^{rt}]$	$\sigma[D_{max}^{rt}]$
ACK	0.945	5.2864	8.7493	0.9788	31.4170	19.6121	0.925	34.6804	19.9445
SACK	0.945	4.4206	6.3285	0.9894	23.7265	16.2619	0.935	27.0219	17.4024
Improvement(%)	—	—	—	1.08	24.48	17.08	1.08	22.08	12.75

#### 4.5.2 Benefits of Global-ACK (G-ACK) over ACK

ACKs from different destinations are mixed inside the network to form G-ACKs. Packets and G-ACKs are separate, and there is no coding at the relays. This Scheme is applicable to topologies where each destination generates only one ACK.

**The G-ACK Schemes:** The G-ACK Scheme differs from the Basic Scheme with the following additional feature. When relay  $i$ , which has an ACK of one destination meets with another relay  $j$ ,  $j \neq i$ , which contains an ACK of a different destination, then both relays combine ACKs to form a G-ACK.  $\square$

In order to observe the benefit of G-ACK over ACK, we consider the example network shown in Figure 4.4 (we call it Topology 2). In this network, there is only one source node which is multicasting its packet to three destination nodes. Each destination generates a (plain) ACK. The average source-degree and destination-degree is equal to 1 and 3, respectively.

We observe the benefits of G-ACK over ACK due to the fact that ACK from different destinations are mixed inside the network resulting in fast return path. Again, this benefit of G-ACK over ACK in the return path is due to the fact that in G-ACK, the meeting between relays both carrying different ACK information results in mixing/updating of ACK at both relays, as compare to ACK scheme where there is no mixing of ACKs inside the network.

In Figure 4.9, we compare CDFs of the return delay in the (plain) ACK Scheme and the G-ACK Scheme between one source/destination pair. In Figure, 4.10, we compare the overall (for whole network) maximum return delay CDFs. Mean delays, success probabilities and corresponding standard deviations alongwith improvement are provided in Table 4.2.

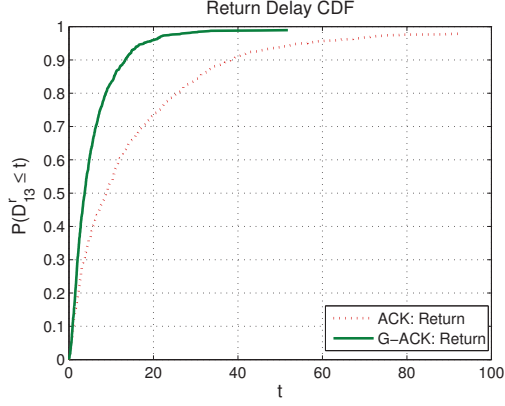


Figure 4.9: Comparison of pairwise return and round trip delay CDF of Plain ACK Scheme and G-ACK Scheme: Topology 2.

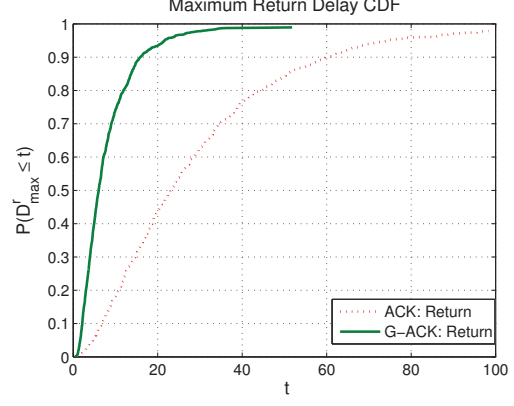


Figure 4.10: Comparison of network-wide maximum round trip delay CDF of Plain ACK Scheme and G-ACK Scheme: Topology 2.

Table 4.2: G-ACK over ACK

Scenario	Forward			Return			Round Trip		
	$P_{11}^f$	$E[D_{11}^f]$	$\sigma[D_{11}^f]$	$P_{11}^r$	$E[D_{11}^r]$	$\sigma[D_{11}^r]$	$P_{11}^{rt}$	$E[D_{11}^{rt}]$	$\sigma[D_{11}^{rt}]$
ACK	1.0	1.0981	0.4589	0.998	13.9772	15.6054	0.998	15.0762	15.7103
G-ACK	0.999	1.1136	0.4240	1.0	5.5053	5.6731	0.999	6.6189	5.7434
Improvement(%)	—	—	—	—	60.61	63.65	—	61.90	63.44
Scenario	$P^f$	$E[D_{avg}^f]$	$\sigma[D_{avg}^f]$	$P^r$	$E[D_{avg}^r]$	$\sigma[D_{avg}^r]$	$P^{rt}$	$E[D_{avg}^{rt}]$	$\sigma[D_{avg}^{rt}]$
ACK	1.0	1.0785	0.3055	0.991	14.0118	9.2984	0.991	15.0916	19.3387
G-ACK	0.999	1.1005	0.3123	1.0	5.4104	4.3494	0.999	6.5109	4.4214
Improvement(%)	—	—	—	0.91	61.39	53.22	0.81	56.86	77.14
Scenario	$P^f$	$E[D_{max}^f]$	$\sigma[D_{max}^f]$	$P^r$	$E[D_{max}^r]$	$\sigma[D_{max}^r]$	$P^{rt}$	$E[D_{max}^{rt}]$	$\sigma[D_{max}^{rt}]$
ACK	1.0	1.3965	0.4192	0.991	27.5326	19.7823	0.991	28.7424	19.8375
G-ACK	0.999	1.4082	0.4191	1.0	7.7784	6.2427	0.999	8.8783	6.2779
Improvement(%)	—	—	—	0.91	71.75	68.44	0.81	69.11	68.35



Referring to Figures 4.9-4.10 and Table 4.2, we make the following observations:

- O3: The return delay CDFs with G-ACK stay above the return delay CDFs with ACK implying that the return delay with G-ACKs is *stochastically smaller* than the return delay with ACKs. The gap between the CDFs is more in this case. Moreover, as compared to Basic Scheme, G-ACK improves return success probability in quite less time as the same is evident from respective delay CDFs. For example, for a pairwise flow (refer to Figure 4.9), at time  $t = 20$ , we observe that return success probability is 0.95 with the G-ACK as compared to 0.74 with the (plain) ACK Scheme.
- O4: Due to mixing of ACK information inside the network, the G-ACK reduces variance significantly as compared to that achieved by the SACK over ACK.

### 4.5.3 Benefits of G-SACK over SACK with Single Destination

In this section, we show the benefits of G-SACK over SACK. Idea of G-SACK is that the destination(s) generate SACK(s) and the SACK(s) are updated inside the network to form G-SACKs.

**The G-SACK Scheme:** The G-SACK Scheme differs from the SACK Scheme as follows. Destinations generate SACKs upon receiving packets, as in the SACK Scheme. The SACKs are updated with latest information about the receipt of packets each time a relay meets with another relay to form G-SACKs. A relay carrying a G-SACK meets with the destination and the destination updates the G-SACK by including its latest packet receipt information.  $\square$

We show the benefits of G-SACK over SACK by considering Topology 1 shown in Figure 4.3 (with Single Destination). In Section 4.5.4, we also show the benefits of G-SACK over SACK by considering Topology 5 shown in Figure 4.6 (with Multiple Destinations).

We observe the benefit of G-SACK over SACK with single destination in the return path which is due to the fact that in the G-SACK, the meeting between relays both carrying different SACK information from the same destination results in mixing/updating of SACK at both relays, as compared to SACK scheme where there is no mixing of SACKs inside the network. We show by the following example that why mixing of the SACKs from the destination inside the network is effective.

*Example 4.5.1.* Consider the following sequence of meetings in order to observe the benefit of G-SACK over SACK: (i) let a relay  $R1$  carrying a packet of Source 1 meets with the destination, it brings ACK 1, (ii) let another relay  $R2$  bringing packet of Source 2 at the destination brings a SACK for Source 1 and Source 2, (iii) let both relays meet with each other before coming in contact with any source(s), and (iv) let  $R1$  meets with Source 2. Thus, when there is no mixing of SACKs inside the network, then meeting of  $R1$  with

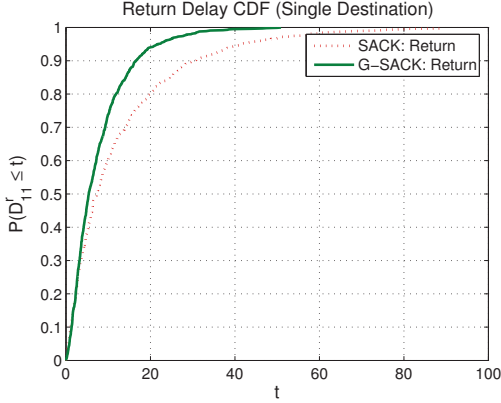


Figure 4.11: Comparison of pairwise return delay CDF of SACK versus G-SACK (single destination): Topology 1.

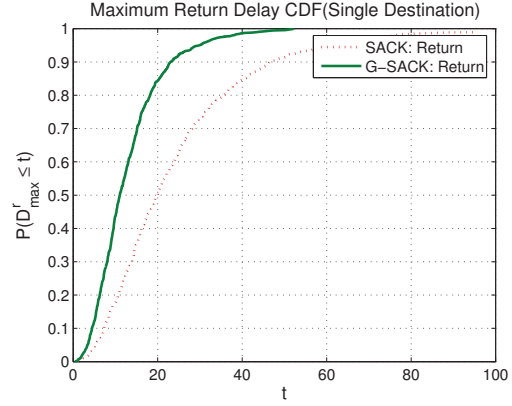


Figure 4.12: Comparison of network-wide maximum return delay CDF of SACK versus G-SACK (single destination): Topology 1.

Source 2 is not useful as it has ACK information only for Source 1. But mixing of SACK information makes this meeting beneficial as  $R1$  gets updated information from  $R2$  upon its meeting which is now useful for Source 2.  $\square$

In Figures 4.11 and 4.12, we compare the CDFs of the return delay (random variable) for one unicast flow and for maximum of all flows in the network respectively with SACK and G-SACK for Topology 1 with three sources. Mean delays, success probabilities and standard deviations alongwith improvement are given in Table 4.3.

Referring to Figures 4.11-4.12 and Table 4.3, we make the following observations:

- O5: We again observe similar benefits as stated in Section 4.5.1 and 4.5.2 as follow. We observe that the return delay CDFs with the G-SACK stay above the return delay CDFs with the SACK implying that the delay with the G-SACK is *stochastically smaller* than the delay with the SACK.
- O6: **Pairwise:** The gain due to G-SACK in return delay is small initially. As time progresses, the destination starts receiving packets from many sources. Then the relays carrying different SACKs start mixing their information resulting in increase in the number of relays carrying more recent SACK information. Now, when a relay carrying SACK information meets with a source node, with very high probability, the source receives back its intended ACK information, resulting in improvement in return path. For example, pairwise return success probability with G-SACK is 0.94 as compared to that of 0.80 with SACK at time  $t = 20$ .

Table 4.3: G-SACK over SACK (Single Destination)

Scenario	Forward			Return			Round Trip		
	$P_{11}^f$	$E[D_{11}^f]$	$\sigma[D_{11}^f]$	$P_{11}^r$	$E[D_{11}^r]$	$\sigma[D_{11}^r]$	$P_{11}^{rt}$	$E[D_{11}^{rt}]$	$\sigma[D_{11}^{rt}]$
SACK	0.985	2.2712	3.9634	0.9969	12.2160	13.8255	0.982	14.4364	14.8146
G-SACK	0.981	2.2204	4.7291	1.0	7.7169	7.2466	0.981	10.2697	8.9624
Improvement(%)	—	—	—	—	36.83	47.59	—	28.86	39.50
Scenario	$P^f$	$E[D_{avg}^f]$	$\sigma[D_{avg}^f]$	$P^r$	$E[D_{avg}^r]$	$\sigma[D_{avg}^r]$	$P^{rt}$	$E[D_{avg}^{rt}]$	$\sigma[D_{avg}^{rt}]$
SACK	0.945	2.2197	2.1982	0.9894	11.8276	6.9466	0.935	14.0360	7.4399
G-SACK	0.934	2.2778	2.4718	1.0	7.5037	4.2978	0.934	9.9727	5.5487
Improvement(%)	—	—	—	—	36.56	38.13	—	28.95	25.42
Scenario	$P^f$	$E[D_{max}^f]$	$\sigma[D_{max}^f]$	$P^r$	$E[D_{max}^r]$	$\sigma[D_{max}^r]$	$P^{rt}$	$E[D_{max}^{rt}]$	$\sigma[D_{max}^{rt}]$
SACK	0.945	4.4206	6.3285	0.9894	23.7265	16.2619	0.935	27.0219	17.4024
G-SACK	0.934	4.6034	7.2524	1.0	12.9835	8.3755	0.934	16.7482	11.0858
Improvement(%)	—	—	—	—	45.28	48.50	—	38.02	36.30

O7: **Network wide:** Improvement in network wide maximum mean return delay starts very quickly as compared to improvement in pairwise mean return delay. For example, network wide maximum return success probability with G-SACK is 0.83 as compared to that of 0.50 with SACK at time  $t = 20$ . The benefit due to G-SACK increases as the source degree of the destination increases (results not reported here).

O8: Mixing of SACK information inside the network (G-SACK) is more beneficial than SACK only in improving return delay and delay variance. For example, SACK improves pairwise return delay by 25.52% over the (plain) ACK Scheme (refer to Table 4.1), whereas G-SACK improves the same by 36.83% even over SACK (refer to Table 4.3). The variance also improves quite significantly due to mixing of SACK information inside the network.

The G-SACK improves delays remarkably even with single destination. We observe huge improvement with multiple destinations case detailed in next Section 4.5.4.

#### 4.5.4 Benefits of G-SACK over SACK with Multiple Destinations

As G-SACK can acknowledge *multiple sources* about the receipt of packets at *multiple destinations*, therefore, we consider the example network shown in Figure 4.6 (Topology 5: multiple destination) in order to show the benefits of G-SACK over SACK. In this Topology, there are three sources and three destinations. Each source node multicasts (actually broadcast) its packet to all three destination nodes. The average source/destination-degree is equal to 3.

In Figure 4.13, we compare CDFs of the return delays between a single source-destination pair of the G-SACK with the SACK for multiple destinations. Similarly, in Figure 4.14,

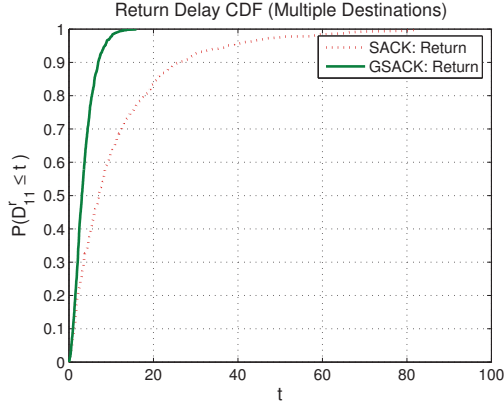


Figure 4.13: Comparison of pairwise return delay CDF of SACK versus GSACK (multiple destinations): Topology 5.

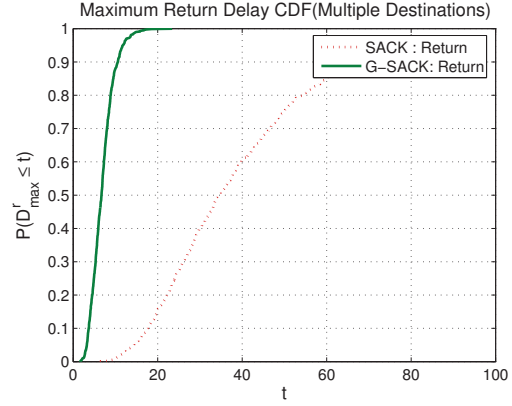


Figure 4.14: Comparison of network-wide maximum return delay CDF of SACK versus G-SACK (multiple destinations): Topology 5.

we compare the overall (for whole network) maximum return delay CDFs. Mean delays, success probabilities and corresponding standard deviations alongwith improvement are provided in Table 4.4.

Referring to Figures 4.13-4.14 and Table 4.4, we make the following observations:

- O9: Again, we observe that delays with G-SACK are stochastically smaller than with SACK. The gain of G-SACK is due to the fact that, unlike a SACK which carries information from a single destination that generates it, a G-SACK initially generated by a destination gathers packet receipt information from multiple destinations on its way back to the sources. Thus, the number of relays carrying packet receipt information from any specific destination increases inside the network. The improvement in delay performance due to G-SACKs over SACKs increases as the fraction of multicast sessions increases because a multicast session must continue until acknowledgements from all the destinations are received, and the G-SACKs help in this regard.

The G-SACK improves network wide maximum return delay significantly by increasing the source/destinations degree. The improvement in variance and success probabilities is also much better as compared to the SACK Scheme. For example, as compared to the SACK, the G-SACK improves pairwise mean return delay by 36.83% and 68.06% for single and multiple destinations case respectively. Similarly, the G-SACK improves maximum mean return delay as compared to the SACK by 45.28% and 81.67% for single destination and multiple destinations scenarios respectively.

Table 4.4: G-SACK over SACK (Multiple Destinations)

Scenario	Forward			Return			Round Trip		
	$P_{11}^f$	$E[D_{11}^f]$	$\sigma[D_{11}^f]$	$P_{11}^r$	$E[D_{11}^r]$	$\sigma[D_{11}^r]$	$P_{11}^{rt}$	$E[D_{11}^{rt}]$	$\sigma[D_{11}^{rt}]$
SACK	0.970	2.1481	3.6806	0.999	11.3114	12.9743	0.969	13.4591	14.0446
G-SACK	0.968	2.3295	4.2461	1.0	3.6124	2.4851	0.968	5.9419	5.1368
Improvement(%)	—	—	—	—	68.06	80.85	—	55.85	63.43
Scenario	$P^f$	$E[D_{avg}^f]$	$\sigma[D_{avg}^f]$	$P^r$	$E[D_{avg}^r]$	$\sigma[D_{avg}^r]$	$P^{rt}$	$E[D_{avg}^{rt}]$	$\sigma[D_{avg}^{rt}]$
SACK	0.835	2.0993	1.2617	0.9748	11.8657	4.0935	0.814	13.9587	4.4508
G-SACK	0.82	2.2526	1.3508	1.0	3.5580	1.3150	0.82	5.8106	2.0040
Improvement(%)	—	—	—	2.59	70.01	67.88	0.74	58.37	54.97
Scenario	$P^f$	$E[D_{max}^f]$	$\sigma[D_{max}^f]$	$P^r$	$E[D_{max}^r]$	$\sigma[D_{max}^r]$	$P^{rt}$	$E[D_{max}^{rt}]$	$\sigma[D_{max}^{rt}]$
SACK	0.835	6.7624	7.5608	0.9748	37.7574	18.1125	0.814	41.3973	18.9178
G-SACK	0.82	7.7116	8.6348	1.00	6.9207	2.7467	0.82	12.5310	8.7067
Improvement(%)	—	—	—	2.59	81.67	84.84	0.74	69.73	53.98

## 4.6 Improving the Forward Path

In this section, we quantify the benefits of inter-session network coding with the following Coding Scheme.

**The Coding Scheme:** Packet payload parts of different sources are combined at relays by forming random linear combinations over the Galois field  $\mathbb{F}_q$  [125, 126]. When relay  $j$ , which has a pure or a coded packet, meets with a source or with another relay that has a pure or a coded packet, relay  $j$  replaces its content with a new RLC. When an empty relay  $j$  meets with a source or with another relay that has a pure or a coded packet, relay  $j$  copies the pure or coded packet.  $\square$

To quantify the benefits of coding, only the forward path is relevant. We quantify the gain due to coding by comparing the forward success probabilities and the Cumulative Distribution Functions (CDFs) of the random forward delays obtained with the Coding Scheme and the Basic Scheme (in which the forward path does not involve coding (see Section 4.4)). Recall that we have taken  $q = 4$  (see Section 4.4).

In Figures 4.15, 4.16 and 4.17, we compare the forward delay CDFs of the source-destination pair (1, 1), the network-wide average and maximum forward delays, respectively, obtained with the Coding Scheme and the Basic Scheme (i.e., without coding) for the topology shown in Figure 4.2 consisting of 3 source-destination pairs. Each source unicasts its packet to one of the destination nodes and the average source/destination-degree is equal to 1. In Figures 4.18, 4.19 and 4.20, we provide results for a similar unicast topology with 10 source-destination pairs. Mean delays, success probabilities and standard deviations along with percentage improvements are provided in Tables 4.5 and 4.6, respectively.

Referring to Figures 4.15-4.20 and Tables 4.5 and 4.6, we make the following observa-

tions:

Table 4.5: Benefits of coding: 3 unicast flows.

Scenario	Forward			Return			Round Trip		
	$P_{11}^f$	$E[D_{11}^f]$	$\sigma[D_{11}^f]$	$P_{11}^r$	$E[D_{11}^r]$	$\sigma[D_{11}^r]$	$P_{11}^{rt}$	$E[D_{11}^{rt}]$	$\sigma[D_{11}^{rt}]$
w/o coding	0.981	2.2295	3.4012	0.994	16.0823	16.1756	0.975	18.3038	16.8181
coding	1.0	1.2229	0.5248	0.991	15.7965	16.7567	0.991	17.0185	16.8304
benefit (%)	1.94	45.15	84.57	—	—	—	—	—	—
Scenario	$P^f$	$E[D_{avg}^f]$	$\sigma[D_{avg}^f]$	$P^r$	$E[D_{avg}^r]$	$\sigma[D_{avg}^r]$	$P^{rt}$	$E[D_{avg}^{rt}]$	$\sigma[D_{avg}^{rt}]$
w/o coding	0.937	2.4574	2.5864	0.9785	16.8021	10.2819	0.917	19.2071	10.5719
coding	1.0	1.2272	0.2863	0.982	16.3564	9.5734	0.982	17.5824	9.6066
benefit (%)	6.72	50.06	88.93	—	—	—	—	—	—
Scenario	$P^f$	$E[D_{max}^f]$	$\sigma[D_{max}^f]$	$P^r$	$E[D_{max}^r]$	$\sigma[D_{max}^r]$	$P^{rt}$	$E[D_{max}^{rt}]$	$\sigma[D_{max}^{rt}]$
w/o coding	0.937	5.0969	7.3967	0.9785	31.3528	19.8724	0.917	34.5955	20.1253
coding	1.0	1.6805	0.4183	0.982	30.5181	19.2508	0.982	31.8224	19.2787
benefit (%)	6.72	67.03	94.34	—	—	—	—	—	—

Table 4.6: Benefits of coding: 10 unicast flows.

Scenario	Forward			Return			Round Trip		
	$P_{11}^f$	$E[D_{11}^f]$	$\sigma[D_{11}^f]$	$P_{11}^r$	$E[D_{11}^r]$	$\sigma[D_{11}^r]$	$P_{11}^{rt}$	$E[D_{11}^{rt}]$	$\sigma[D_{11}^{rt}]$
w/o coding	0.923	5.4488	9.3091	0.986	18.4437	17.0452	0.911	23.6602	19.0294
coding	1	3.8322	1.5971	0.995	14.6955	16.5397	0.995	18.5263	16.9059
benefit (%)	8.34	29.67	82.84	—	—	—	—	—	—
Scenario	$P^f$	$E[D_{avg}^f]$	$\sigma[D_{avg}^f]$	$P^r$	$E[D_{avg}^r]$	$\sigma[D_{avg}^r]$	$P^{rt}$	$E[D_{avg}^{rt}]$	$\sigma[D_{avg}^{rt}]$
w/o coding	0.431	5.7216	2.7548	0.925	18.1435	5.3114	0.431	23.7155	5.8263
coding	1	3.7930	0.5298	0.947	15.3277	5.1744	0.947	19.1169	5.2629
benefit (%)	132.02	33.71	80.77	—	—	—	—	—	—
Scenario	$P^f$	$E[D_{max}^f]$	$\sigma[D_{max}^f]$	$P^r$	$E[D_{max}^r]$	$\sigma[D_{max}^r]$	$P^{rt}$	$E[D_{max}^{rt}]$	$\sigma[D_{max}^{rt}]$
w/o coding	0.431	24.9669	18.2471	0.925	51.0408	18.3003	0.431	60.2884	17.0522
coding	1	5.8646	1.3054	0.947	48.0331	18.9829	0.947	52.3274	19.0535
benefit (%)	132.02	76.51	92.85	—	—	—	—	—	—

O10: For the case with 3 unicast flows, the pairwise forward success probability,  $P_{11}^f$ , without coding is observed to be 98.1% and the network-wide forward success probability  $P^f = 93.7\%$ . We emphasize that  $P_{11}^f$  and  $P^f$  are less than 100%, even with a load of only 3 unicast flows, primarily due to Type-II non-reachability (recall that Type-I and Type-II non-reachability issues are explained in Section 4.2). We have observed from simulations that even a five-fold increase of the simulation time from  $T = 100$  to  $T = 500$  does not improve the forward success probabilities any further. In fact, it can be observed in Figure 4.15 that the flow (1, 1) is complete within 40 units of time if it does. Similarly, it can be observed in Figure 4.17 that all flows are complete within 80 units of time if they do.

With coding, however,  $P_{11}^f$  becomes 100%. Indeed, coding rules out Type-II non-

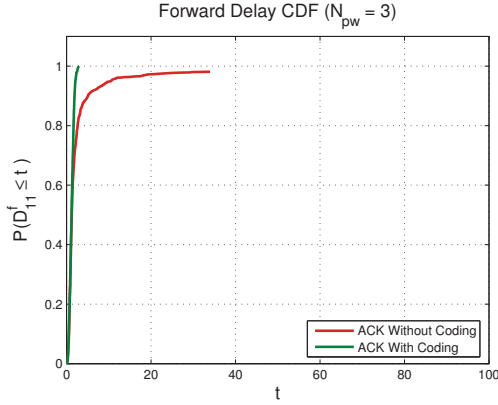


Figure 4.15: Comparison of pairwise forward delay CDFs with and without coding: Topology 1.

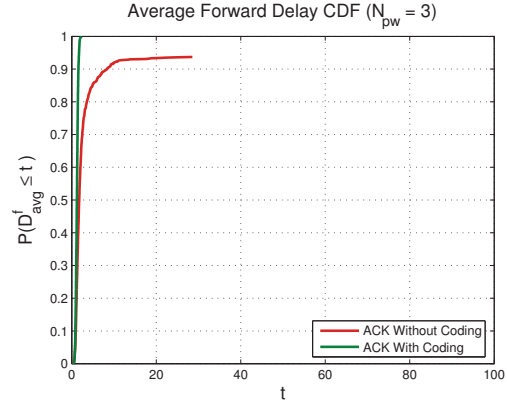


Figure 4.16: Comparison of network-wide average forward delay CDFs with and without coding: Topology 1.

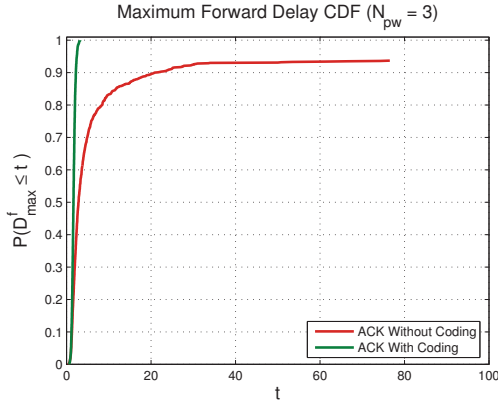


Figure 4.17: Comparison of network-wide maximum forward delay CDFs with and without coding: Topology 1.

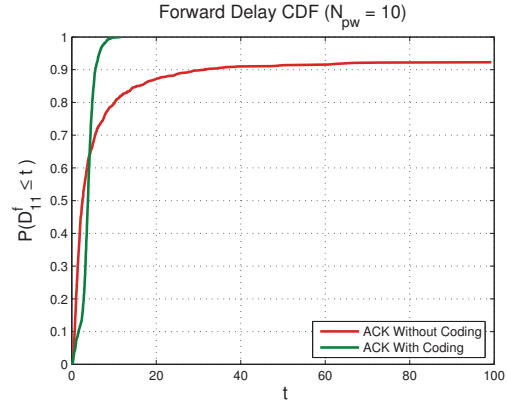


Figure 4.18: Comparison of pairwise forward delay CDFs with and without coding.

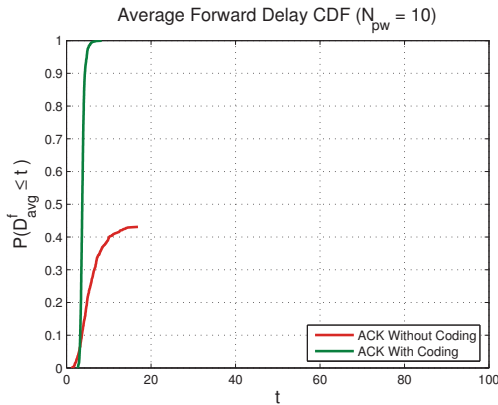


Figure 4.19: Comparison of network-wide average forward delay CDFs with and without coding.

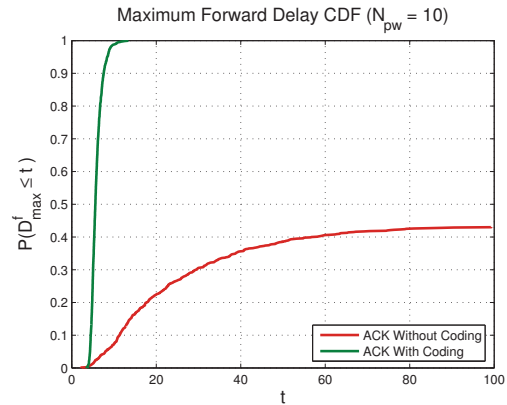


Figure 4.20: Comparison of network-wide maximum forward delay CDFs with and without coding.

reachability; sooner or later each source meets with some relay and its packet enters into the network and finally gets delivered (it may take a long delay, but finally it does get delivered).

O11: For a higher load of 10 unicast flows, for the case without coding, the simulation time  $T = 100$  is short enough to cause Type-I non-reachability as well; observe that the red curves in Figures 4.18 and 4.20 go until a delay value  $t = 100$ . However, with coding, the forward success probabilities (both pairwise and network-wide) are 100%. In fact, it can be observed in Figures 4.15-4.20 that, except for a negligible initial part, the forward delay CDFs with coding remain above the forward delay CDFs without coding. This implies that, except for the negligible initial part, *the forward delays with coding are stochastically smaller than the forward delays without coding* (this is also supported by significantly smaller expected delays and standard deviations with coding). Moreover, coding provides 100% successful completion within a much smaller time budget of  $\approx 10$  time units (see Figure 4.18). A 132% improvement (2.32 times increase) in network-wide success probability is indeed remarkable.

O12: The 80-94% improvement (5- to 18-fold reduction) in the standard deviation of forward delays is also remarkable, which implies that *coding makes the packet transfers smoother*. However, we observe that the forward delay CDFs with and without coding cross each other (visible only in Figure 4.18) and the forward delay CDF without coding stays below the forward delay CDF with coding for a negligible initial part. This can be explained as follows. Coding involves a small delay for the packets from various sources to get mixed. Also the destination needs, on the average, more than one coded packets to extract its required packet whereas exactly one (uncoded or pure) packet is required without coding.

Indeed, the benefits of coding come into play after a small delay, which we call the *mixing delay*, after which there is a proper mixture of coded and uncoded packets in the network. The mixing delay can be measured by the delay value on the  $t$ -axis (x-axis) when the two CDFs cross each other. When there is no coding, some packet might reach its destination very quickly due to randomness and this accounts for the CDF without coding staying below the CDF with coding before the mixing delay. However, this slows down the transfer of other packets (as discussed in Section 4.2), and, on the average, the case without coding performs worse.

O13: The ratio of maximum to average expected forward delay decreases from 2.07 (without coding) to 1.37 (with coding) in case of 3 unicast flows. Similarly, it decreases from 4.36 (without coding) to 1.55 (with coding) in case of 10 unicast flows. This



implies that *coding improves the fairness among the packet transfers* and that *the improvement increases with increase in the load on the network*.

O14: We also observe improvement in return delay (refer to Table 4.6) with coding as compared to without coding. In fact, with coding, when forward path completes, there are still enough number of empty relays in the network as compared to without coding. Therefore, ACK(s) is/are replicated to more relays and return delay with coding is less than that without coding.

## 4.7 Benefits of overall Proposed Scheme over Basic Scheme

In this section, we show the benefits of our overall proposal over the Basic Scheme without coding.

**The Proposed Scheme:** Our Proposed Scheme consists of (i) coding at relay to improve the forward path, and (ii) the G-SACK to improve the return path.  $\square$

Now, we consider three different topologies in order to show the benefits of our Proposed Scheme against the Basic Scheme (plain ACK without coding). The topologies considered are Topology 3, Topology 4 and Topology 5 shown in Figures 4.2, 4.5 and 4.6, respectively. Note that in Topology 4 shown in Figure 4.5, there are three sources and three destinations. One source node multicasts (actually broadcast) its packet to all destination nodes, one source node multicasts its packet to two destination nodes while one source node unicasts its packet to one destination node. The average source/destination-degree is equal to 2 in Topology 4.

In Figures 4.21, 4.22 and 4.23, we compare the pairwise round trip delay CDFs of the Basic Scheme and the Proposed Scheme for unicast, 50% mix of unicast and multicast, and multicast respectively.

In Figures 4.24, 4.25 and 4.26, we compare the maximum (of all flows) round trip delay CDFs of the Basic Scheme and the Proposed Scheme for unicast, 50% mix of unicast and multicast, and multicast respectively. Mean delays, success probabilities and corresponding standard deviations alongwith improvement are provided in Table 4.7, Table 4.8 and Table 4.9.

Next, we obtain and compare additional simulation results with  $N = 25, 50, 75$ . But, as suggested in [88], we keep the product  $N\beta$  as constant. In Figures 4.27 and 4.28, we compare the mean forward and round trip delay and the corresponding success probabilities, respectively, of the Basic Scheme and the Proposed Scheme by varying the number of relays  $N$ .

Refer to Figures 4.21-4.28) and Tables 4.7-4.9, we make the following observations.

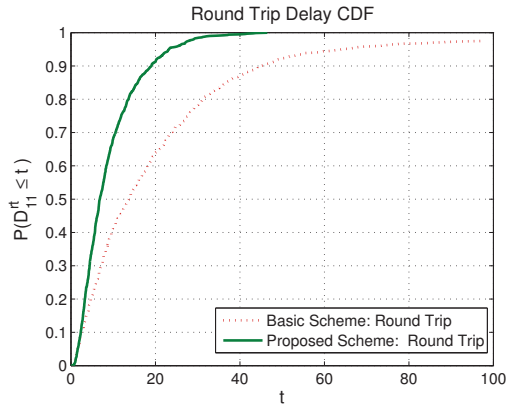


Figure 4.21: Comparison of pairwise round trip delay CDF of Basic Scheme versus Proposed Scheme: Topology 3.

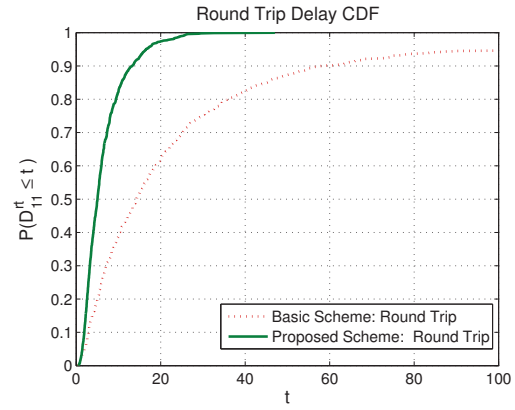


Figure 4.22: Comparison of pairwise round trip delay CDF of Basic Scheme versus Proposed Scheme: Topology 4.

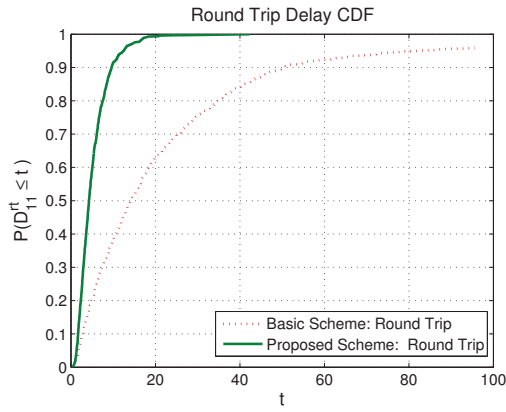


Figure 4.23: Comparison of pairwise round trip delay CDF of Basic Scheme versus Proposed Scheme: Topology 5.

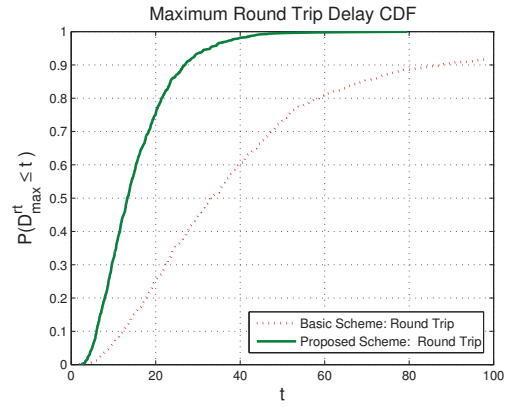


Figure 4.24: Comparison of network-wide maximum round trip delay CDF of Basic Scheme versus Proposed Scheme: Topology 3.

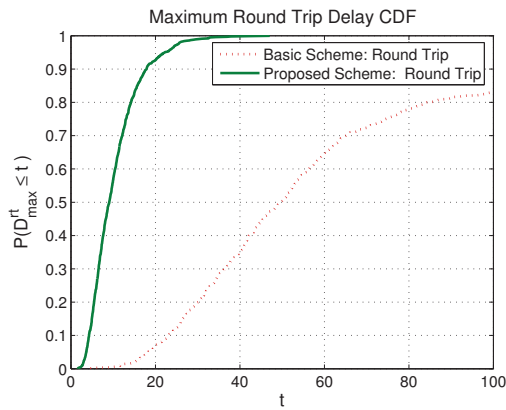


Figure 4.25: Comparison of network-wide maximum round trip delay CDF of Basic Scheme versus Proposed Scheme: Topology 4.

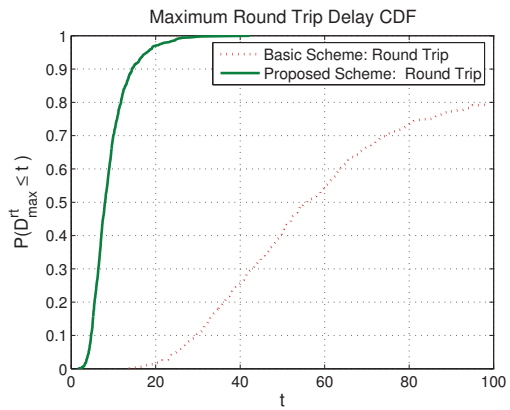


Figure 4.26: Comparison of network-wide maximum round trip delay CDF of Basic Scheme versus Proposed Scheme: Topology 5.

Table 4.7: Proposed over Basic Scheme: Topology 3

Scenario	Forward			Return			Round Trip		
	$P_{11}^f$	$E[D_{11}^f]$	$\sigma[D_{11}^f]$	$P_{11}^r$	$E[D_{11}^r]$	$\sigma[D_{11}^r]$	$P_{11}^{rt}$	$E[D_{11}^{rt}]$	$\sigma[D_{11}^{rt}]$
Basic	0.981	2.2295	3.4012	0.994	16.0823	16.1756	0.975	18.3038	16.8181
Proposed	1.0	1.2661	0.5103	1.0	7.7358	7.0283	1.0	9.0018	7.1703
Improvement(%)	1.94	43.21	85.0	0.62	51.90	56.55	2.56	50.82	57.37
Scenario	$P^f$	$E[D_{avg}^f]$	$\sigma[D_{avg}^f]$	$P^r$	$E[D_{avg}^r]$	$\sigma[D_{avg}^r]$	$P^{rt}$	$E[D_{avg}^{rt}]$	$\sigma[D_{avg}^{rt}]$
Basic	0.937	2.4574	2.5864	0.9785	16.8021	10.2819	0.917	19.2071	10.5719
Proposed	1.0	1.2639	0.2951	1.0	7.8367	4.5957	1.0	9.1006	4.6528
Improvement(%)	6.72	48.571	88.59	2.20	53.36	55.30	9.05	52.62	55.99
Scenario	$P^f$	$E[D_{max}^f]$	$\sigma[D_{max}^f]$	$P^r$	$E[D_{max}^r]$	$\sigma[D_{max}^r]$	$P^{rt}$	$E[D_{max}^{rt}]$	$\sigma[D_{max}^{rt}]$
Basic	0.937	5.0969	7.3967	0.9785	31.3528	19.8724	0.917	34.5955	20.1253
Proposed	1.0	1.7284	0.4293	1.0	13.8886	9.0026	1.0	15.3496	9.0492
Improvement(%)	6.72	66.09	94.20	2.20	55.70	54.70	9.05	55.63	55.04

Table 4.8: Proposed over Basic Scheme: Topology 4

Scenario	Forward			Return			Round Trip		
	$P_{11}^f$	$E[D_{11}^f]$	$\sigma[D_{11}^f]$	$P_{11}^r$	$E[D_{11}^r]$	$\sigma[D_{11}^r]$	$P_{11}^{rt}$	$E[D_{11}^{rt}]$	$\sigma[D_{11}^{rt}]$
Basic	0.952	2.4497	4.1897	0.9936	16.6622	17.2934	0.946	19.0762	18.1168
Proposed	1.0	1.2695	0.5459	1.0	5.1643	4.9591	1.0	6.4338	5.1240
Improvement(%)	5.01	48.18	86.97	0.64	69.01	71.32	5.71	66.27	71.72
Scenario	$P^f$	$E[D_{avg}^f]$	$\sigma[D_{avg}^f]$	$P^r$	$E[D_{avg}^r]$	$\sigma[D_{avg}^r]$	$P^{rt}$	$E[D_{avg}^{rt}]$	$\sigma[D_{avg}^{rt}]$
Basic	0.867	2.2018	1.5887	0.9584	17.1848	7.0010	0.831	19.3647	7.2744
Proposed	1.0	1.2588	0.2648	1.0	4.1086	2.5586	1.0	5.2783	2.6651
Improvement(%)	15.34	42.83	83.33	4.34	76.09	63.45	20.34	72.74	63.36
Scenario	$P^f$	$E[D_{max}^f]$	$\sigma[D_{max}^f]$	$P^r$	$E[D_{max}^r]$	$\sigma[D_{max}^r]$	$P^{rt}$	$E[D_{max}^{rt}]$	$\sigma[D_{max}^{rt}]$
Basic	0.867	6.0504	7.3123	0.9584	42.8873	19.2729	0.831	45.7103	19.6193
Proposed	1.0	1.8318	0.4293	1.0	8.9410	5.8355	1.0	10.3393	5.9502
Improvement(%)	15.34	69.72	94.13	4.34	79.15	69.72	20.34	77.38	69.67

Table 4.9: Proposed over Basic Scheme: Topology 5

Scenario	Forward			Return			Round Trip		
	$P_{11}^f$	$E[D_{11}^f]$	$\sigma[D_{11}^f]$	$P_{11}^r$	$E[D_{11}^r]$	$\sigma[D_{11}^r]$	$P_{11}^{rt}$	$E[D_{11}^{rt}]$	$\sigma[D_{11}^{rt}]$
Basic	0.963	2.3487	3.7792	0.9958	16.6338	17.0530	0.959	18.9828	17.5904
Proposed	1.0	1.2933	0.5621	1.0	3.9447	3.6940	1.0	5.2380	3.8859
Improvement(%)	3.84	44.94	85.13	0.42	76.29	78.34	4.28	72.41	77.91
Scenario	$P^f$	$E[D_{avg}^f]$	$\sigma[D_{avg}^f]$	$P^r$	$E[D_{avg}^r]$	$\sigma[D_{avg}^r]$	$P^{rt}$	$E[D_{avg}^{rt}]$	$\sigma[D_{avg}^{rt}]$
Basic	0.829	2.3312	1.6827	0.9565	16.4299	5.5921	0.793	18.7245	5.8749
Proposed	1.0	1.2746	0.2493	1.0	3.9598	2.1756	1.0	5.1647	2.2658
Improvement(%)	20.63	45.32	85.18	4.55	75.90	61.10	26.10	72.42	61.43
Scenario	$P^f$	$E[D_{max}^f]$	$\sigma[D_{max}^f]$	$P^r$	$E[D_{max}^r]$	$\sigma[D_{max}^r]$	$P^{rt}$	$E[D_{max}^{rt}]$	$\sigma[D_{max}^{rt}]$
Basic	0.829	8.5356	11.4188	0.9565	47.1655	18.0528	0.793	50.7366	18.4644
Proposed	1.0	1.9503	0.4802	1.0	8.0163	4.4875	1.0	9.0674	4.5419
Improvement(%)	20.63	77.15	95.79	4.55	83.0	75.14	26.10	82.13	75.40

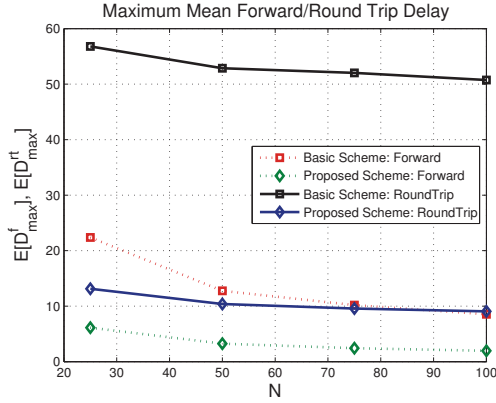


Figure 4.27: Comparing network-wide mean forward and mean round trip delay of Basic Scheme with Proposed Scheme: Topology 5.

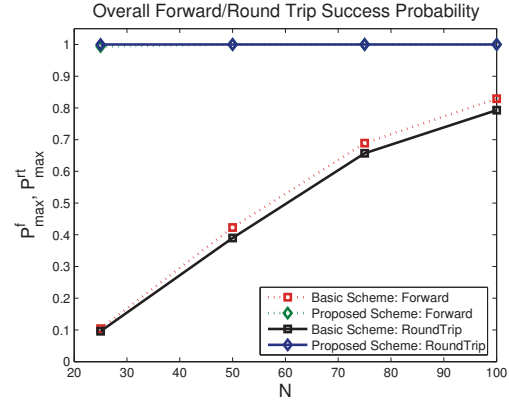


Figure 4.28: Comparing network-wide forward and round trip success probability of Basic Scheme with Proposed Scheme: Topology 5.

- O15: The pairwise as well as the network wide maximum mean round trip delays with our Proposed Scheme are stochastically significantly smaller as compared to the Basic Scheme. This is true for any topology and we provide results with Topologies 3,4 and 5 (refer to Figures 4.21-4.26). The decrease in the mean round trip delay with our Proposed Scheme as compared to the Basic Scheme is indeed remarkable. It is also worth noticing that the overall success probability with our Proposed Scheme for all considered topologies is equal to 1. It is evident that network wide maximum round trip delay CDFs reach 1 very quickly. This happens even with as small as 50 nodes.
- O16: The performance of the Basic Scheme goes on decreasing with increase of average source-degree and destination-degree, while it impacts very little the performance of the Proposed Scheme. In fact, the percentage improvement in overall mean delay with our Proposed Scheme as compared to the Basic Scheme goes on increasing with increasing source/destination degree.
- O17: The decrease in the mean forward and round trip delays with our overall Proposed Scheme as compared to the Basic Scheme are indeed remarkable for all values of  $N = 25, 50, 75, 100$  (refer to Figures 4.27-4.28). It is also worth noticing that the forward as well as round trip success probabilities with our Proposed Scheme are equal to 1 for  $N = 50, 75, 100$ . For  $N = 25$ , overall forward success probability is 0.994 while overall round trip success probability is still 1. In Figure 4.27, each point is calculated against successful experiments only out of 1000 experiments and the corresponding success probabilities are shown in Figure 4.28.

**remark:** It is worth mentioning that our Proposed Scheme is quite efficient and achieves overall success probability equal to 1 even with 10 different source nodes each multicasting its packet to 10 destination nodes (results not reported here) and even with number of relays equal to 50 by keeping the product  $N\beta$  as constant as suggested in [88]. Actually, network coding enables to complete overall forward transmission very quickly (after sufficient mixing of packets of all sources occurs inside the network) and G-SACK is major contributor for overall success probability.

## 4.8 Conclusion of the Chapter

We have proposed and studied via extensive simulations a set of enhancements to improve reliable transport in DTNs covering both unicast and multicast flows. We have observed that random linear coding makes the forward component more reliable (*i.e.*, increases the forward success probability) and the G-SACK makes the return path more reliable.

We have quantified the gains achieved due to each feature of our proposal as well as the overall gain achieved by all of them. We have learnt the important lesson that mixing inside the network is extremely beneficial in (1) significantly decreasing the delay variance, and (2) significantly decreasing the network wide maximum delay. In particular, we have observed that coding makes the forward path smoother, *i.e.*, it decreases delay jitter.

# Tracking Message-Spread in DTNs

## Contents

---

<b>5.1</b>	Network Model . . . . .	103
<b>5.2</b>	Characterization of the Process and the Observation . . . . .	106
<b>5.3</b>	Objective and Approach . . . . .	107
<b>5.4</b>	Fluid and Diffusion Approximations . . . . .	108
5.4.1	Fluid Approximation for the Spreading Process . . . . .	109
5.4.2	Diffusion Approximation for the Spreading Process . . . . .	111
5.4.3	Diffusion Approximation for the Observation . . . . .	112
<b>5.5</b>	Discrete Time Kalman Filtering . . . . .	114
<b>5.6</b>	Linear Minimum Mean Squared Error Estimator . . . . .	118
5.6.1	Derivation of the Means and (Co)variances . . . . .	119
<b>5.7</b>	Performance of Analytical Prediction and Estimation . . . . .	121
5.7.1	Direct Delivery . . . . .	121
5.7.2	Epidemic Routing . . . . .	125
<b>5.8</b>	Conclusion of the Chapter . . . . .	125
<b>5.9</b>	Appendices . . . . .	127
5.9.1	Fluid and Diffusion Models . . . . .	127
5.9.2	Brownian Motion . . . . .	128
5.9.3	Density-Dependent Markov Chains . . . . .	129
5.9.4	Continuous Mapping Approach . . . . .	132

---

In this chapter, our objective is to estimate and track the degree of spread of a message/file in the network. Indeed, having such real-time information is critical for on-line control of routing and energy expenditure. It also benefits the multi-casting application. In fact, since multiple copies of the same packet are allowed to spread in the network, it is important to track the number of copies so as to have an on-line adaptive replication policy. In this chapter, we address the problem of tracking the number of copies both in the case of direct delivery [16] and epidemic routing [17]. The problem of adaptively controlling the spreading process is out of the scope of this thesis.

It is important to realize that, with only probabilistic knowledge available on the network dynamics, it would be impossible to infer the spreading of messages/copies of packets in *specific realizations* or the so-called *sample paths*. Since the problem of interest is of tracking a specific realization, one must resort to taking measurements and combine it with the probabilistic characterization of the network dynamics to extract valuable information about specific sample paths. For the purpose of taking measurements, we assume that there are several *observer* nodes in the network which count the number of copies of the message. Our approach is to keep the counting process *anonymous* (i.e., user IDs are not revealed to the observers) and *lightweight* (i.e., the history of meetings with specific users are not maintained) so that a sufficiently large number of (possibly third-party) observers can be employed for tracking purpose in order to achieve high accuracy.

**Our Contributions:** Under the assumption of exponential inter-meeting times of mobile nodes, we solve the following problems. First, we obtain the fluid approximation of the spreading process which, in the absence of measurements, is the only meaningful information one can derive from the probabilistic characterization of the network dynamics with relatively large number of nodes. Second, we derive the diffusion approximations for the spreading and observation processes, and combine them with actual observations in order to track the process using Kalman filters. Third, we evaluate the accuracy of tracking by computing the average error in predicting certain level-crossing times (i.e., the time at which the process and the estimation cross certain pre-specified thresholds). Fourth, in the case of direct delivery, we derive the exact expressions for the *instantaneous*<sup>1</sup> linear Minimum-Mean-Squared-Error (MMSE) estimator and compare its performance with that of Kalman filter in terms of accuracy in predicting level-crossing times.

**Comparison with Related Work:** Mean-field approximations have been used to estimate the mean number of infected nodes under various spreading policies [88]. Such approximations are accurate when the number of nodes is sufficiently large. Our approach of using measurements with MMSE estimator and Kalman filter allows us to track the discrepancy between the mean-field approximations and the actual sample path of the pro-

---

<sup>1</sup>We call it *instantaneous* since the estimate at time  $t$  is based only on the measurement at time  $t$ .

cess. Furthermore, our Kalman filter estimation is based on a second-order approximation whereas the mean-field approximations are only first-order descriptions.

A related estimation problem in wire-line networks has been considered in [131], where the number of participants to a multicast session is tracked over time thanks to measurements taken by polling the users. In [131], the authors assume an *infinite* population from which arrivals occur and apply the diffusion approximation of the well-known  $M/M/\infty$  queueing model. We, however, consider a realistic *finite* population from which arrivals occur. In [131], the Kalman filter is developed to track the fluctuations in the *stationary regime* of the  $M/M/\infty$  queue. We, however, track the fluctuations in the *transient phase*. Furthermore, in [131], the delay in measurements (i.e., of the poll messages and the returning acknowledgments) is ignored. We, however, explicitly characterize the measurement process which complicates the derivation of the measurement equation.

**Applications:** The main application of interest we have in mind is **advertising in mobile social networks (MSNs)**. In such networks, the source (operator) could be interested to stop the advertisement after it reaches to a certain percentage of users. In such a case, it is necessary to track the number of users that have been advertised. The framework of intermittent connectivity and epidemic routing also applies to two more applications, namely, (i) **P2P file sharing**, and (ii) **Internet worm detection**.

## 5.1 Network Model

We consider a DTN consisting of  $S_0$  sources,  $N_0$  users, and  $H_0$  observers. The users are mobile, but the sources and the observers are *static* (see Remark 5.1.3 for the mobile case). The static sources and the observers are connected by a wired network. One can think of the sources and the observers as base stations (BSs) and/or WiFi access points (APs), and the users as mobile terminals (MTs) and/or vehicles with wireless devices. Figure 5.1 depicts such a DTN which represents a sparse deployment of BSs and a sparse population of MTs.

Two nodes are said to “meet” when they come within the communication range of each other (due to mobility of one or both of the nodes). Let  $T_{ij}^{SU}(k)$  denote the time at which source  $i$  and user  $j$ ,  $i = 1, \dots, S_0$ ,  $j = 1, \dots, N_0$ , meet for the  $k$ -th time. Let

$$\tau_{ij}^{SU}(k) := T_{ij}^{SU}(k) - T_{ij}^{SU}(k-1)$$

denote the  $k$ -th inter-meeting time of source  $i$  and user  $j$ , where  $T_{ij}^{SU}(0) := 0$  for all  $i = 1, \dots, S_0$ ,  $j = 1, \dots, N_0$ . Similarly, by  $\tau_{lj}^{HU}(k)$  (resp.  $\tau_{jj'}^{UU}(k)$ ) we denote the  $k$ -th inter-meeting time of observer  $l$  and user  $j$  (resp. users  $j$  and  $j'$ ),  $l = 1, \dots, H_0$ ,  $j, j' = 1, \dots, N_0$ . We make the following independence assumptions regarding the inter-meeting times:



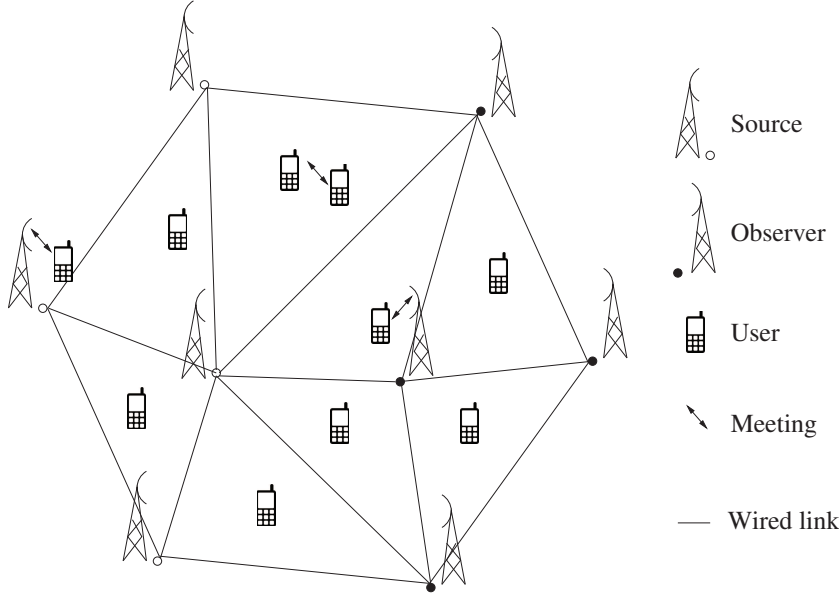


Figure 5.1: A DTN consisting of  $S_0$  sources,  $H_0$  observers and  $N_0$  users.

- (A1) For each pair  $(i, j)$ ,  $i = 1, \dots, S_0$ ,  $j = 1, \dots, N_0$ , the sequence of inter-meeting times  $\{\tau_{ij}^{SU}(k)\}$ ,  $k = 1, 2, \dots$ , consists of independent *exponential* random variables with parameter  $\beta_S$ .
- (A2) For each pair  $(l, j)$ ,  $l = 1, \dots, H_0$ ,  $j = 1, \dots, N_0$ , the sequence of inter-meeting times  $\{\tau_{lj}^{HU}(k)\}$ ,  $k = 1, 2, \dots$ , consists of independent *exponential* random variables with parameter  $\beta_H$ .
- (A3) For each pair  $(j, j')$ ,  $j, j' = 1, \dots, N_0$ , the sequence of inter-meeting times  $\{\tau_{jj'}^{UU}(k)\}$ ,  $k = 1, 2, \dots$ , consists of independent *exponential* random variables with parameter  $\beta_U$ .
- (A4) The inter-meeting times of any pair of nodes is independent of the inter-meeting times of any other pair of nodes. In particular, we have that:

*“The observer-user inter-meeting times are independent of the source-user as well as the user-user inter-meeting times.”*

We focus on tracking the spread of *one* given message. At time  $t = 0$ , the message is only available at the  $S_0$  sources. When a user meets with a source, the user gets a copy of the message with probability  $p_S$  (if the user does not already have the message). The probability  $1 - p_S$  models the link-layer error. Depending on whether (or not) users spread copies of the message to other users, we study two basic paradigms of message forwarding,

namely, *direct delivery* [132] and *epidemic routing* [17]. In the case of direct delivery, users do not spread the message to other users. In the case of epidemic routing, whenever a user  $j$  having the message meets with another user  $j'$  which does not have the message, user  $j'$  gets a copy of the message with probability  $p_U$ , where  $1 - p_U$  models the link-layer error for user-user communication.

It is clear that a source will not have perfect knowledge of the number of users that have already received the message. This is due to one of the following reasons:

- (i) The source keeps transmitting the message as a *broadcast* message, and a user receives the message (with probability  $p_S$ ), but no acknowledgment (ACK) is sent by the user because the source is broadcasting.
- (ii) The source transmits the message as a unicast message upon meeting with a user which receives the message (with probability  $p_S$ ) and sends an ACK, but the source does not receive the ACK (probably, because the user has already moved out of the range of the source).
- (iii) In the case of epidemic routing, the users also spread the message to other users (with probability  $p_U$ ), and the source cannot count such forwarding events.

Since the sources do not know the exact number of users that have received the message, they take the help of *external* observers to estimate and track the degree of spread of the message over time (see Remark 5.1.2). At each contact with a user, an observer gets to know if the user has the message (or not), and simply increments a counter if the user is found to have the message. The observers send their observations (counts) to a central entity using the wired network. The sum of the observers' counts is used as the measurement.

We assume that:

- (A5) The contact duration of an observer and a user is long enough for the observer to know *exactly* whether the user has (or does not have) the message.

This is a reasonable assumption since two short control packets (one from the observer to the user and one from the user to the observer) suffice for this purpose. On the other hand, the actual message can be long enough so that the contact duration of a source and a user may not suffice for the user to successfully receive the message and the source to successfully receive the ACK, if any.

*Remark 5.1.1.* In our setting, an observer does not (need to) know the identity of a user it meets, and does not (need to) keep the history of its meetings with specific users. Hence, one copy at one user might be counted multiple times by the observer(s). However, this *anonymous* and *light-weight* counting scheme allows one to take the help of third-party observers.  $\square$

*Remark 5.1.2.* We assume that all the  $H_0$  observers are external, i.e., they are different from the sources and the users. However, it is possible to extend to the case where each of the  $S_0$  source nodes can also act as an observer for a certain fraction of time. However, to keep the source-user meetings independent of the observer-user meetings (see A4), a node should not act as a source and an observer at the same time, i.e., a node should not give a copy of the message to a user and count the copy at the same time.  $\square$

*Remark 5.1.3.* We assume the sources and the observers to be static. However, the sources and the observers can be allowed to be mobile in the model and our analysis methodology developed in subsequent sections would still apply in the following situations as well. The sources can be allowed to be mobile in the model if, whenever an observer meets with another node, it can distinguish between whether the node is a source or a user. The observers can be allowed to be mobile in the model if the observers can send their observation (counts) to a central entity without delay, possibly, using an alternative technology which delivers the observations with negligible delay.  $\square$

## 5.2 Characterization of the Process and the Observation

Let  $X(t)$  denote the number of *users* that have the message at time  $t$ . Note that  $X(t)$  does not include the sources. Thus,  $X(0) = 0$ . Let  $Y(t)$  denote the *total count of all the observers* up to time  $t$ , with  $Y(0) := 0$ . Henceforth, we refer to  $\{X(t), t \geq 0\}$  as “the process” and to  $\{Y(t), t \geq 0\}$  as “the observation” or “the measurement”. In this section (as well as in the subsequent sections) we analyze the process  $\{X(t)\}$  and the observation  $\{Y(t)\}$  both in the case of direct delivery and epidemic routing in a unified manner.

From the network model described in Section 5.1, it is clear that the process  $\{X(t), t \geq 0\}$  is a continuous time Markov chain (CTMC) with state space  $\mathcal{S} = \{0, 1, \dots, N_0\}$  and transition rate matrix  $\mathcal{Q} = [q(X, X')]_{X, X' \in \mathcal{S}}$  given by

**Direct Delivery:**

$$q(X, X + l) = \begin{cases} p_S \beta_S S_0 (N_0 - X), & \text{if } l = +1, \\ -p_S \beta_S S_0 (N_0 - X), & \text{if } l = 0, \\ 0, & \text{otherwise.} \end{cases}$$

**Epidemic Routing:**

$$q(X, X + l) = \begin{cases} (p_S \beta_S S_0 + p_U \beta_U X) (N_0 - X), & \text{if } l = +1, \\ -(p_S \beta_S S_0 + p_U \beta_U X) (N_0 - X), & \text{if } l = 0, \\ 0, & \text{otherwise.} \end{cases}$$

Given the process  $\{X(t), t \geq 0\}$ , the observation  $\{Y(t), t \geq 0\}$  is a *doubly-stochastic*

Poisson process (see [133, Chapter II]) with (stochastic) intensity function<sup>2</sup>  $\beta_H H_0 X(t)$ , i.e.,

$$Y(t) = \mathcal{P} \left( \beta_H H_0 \int_0^t X(u) du \right), \quad (5.1)$$

where  $\{\mathcal{P}(t), t \geq 0\}$  denotes a unit rate Poisson process which is independent of  $\{X(t), t \geq 0\}$ . The independence of the processes  $\{\mathcal{P}(t), t \geq 0\}$  and  $\{X(t), t \geq 0\}$  is due to the fact that  $\{\mathcal{P}(\beta_H H_0 t), t \geq 0\}$  is determined by observer-user meetings that are independent of source-user and user-user meetings (see A4) which determine  $\{X(t), t \geq 0\}$ .

Note that, starting at State-0, the process  $\{X(t)\}$  ultimately gets absorbed in State- $N_0$ , which is the only absorbing state. Thus, the stationary behavior of the process  $\{X(t)\}$  is not of much interest. We are mainly interested in the transient behavior of the process  $\{X(t), t \geq 0\}$ . In particular, by using a model for the transient behavior of the process  $\{X(t), t \geq 0\}$  and taking measurements, *we want to track a specific sample path in a specific realization of the process  $\{X(t), t \geq 0\}$ .*

### 5.3 Objective and Approach

Our objective is to solve the following problems:

- P1: Analytically predict the average trajectory of the process  $\{X(t), t \geq 0\}$ .
- P2: Track specific sample paths of  $\{X(t), t \geq 0\}$  using the history of observation,  $\{Y(u), u \in U, U \subseteq [0, t]\}$ .
- P3: Estimate the time  $t_L$  when the process crosses a certain level  $X_L$ .

Next, we outline our approach to solve P1-P3.

**Problem P1:** To solve P1, we obtain the fluid approximation for the process  $\{X(t), t \geq 0\}$  (Section 5.4.1). An informal background on fluid and diffusion approximations has been provided in Appendix 5.9.1.

**Problem P2:** To solve P2, we obtain the diffusion approximations for the process (Section 5.4.2) as well as for the observation (Section 5.4.3). In Section 5.5, we sample the diffusion approximation of the process (resp. of the observation) to derive a discrete time stochastic linear difference equation describing the system dynamics (resp. describing the dependence of measurements  $Y$  on the process  $X$ ). Then, using the *system dynamics equation* and the *measurement equation*, and applying Kalman filtering we track the trajectory of the process.

---

<sup>2</sup>A *homogeneous* Poisson process over time is characterized by a “constant” or time-independent intensity. A *non-homogeneous* Poisson process is characterized by an intensity function which is a deterministic function of time. A doubly-stochastic Poisson process has an intensity function which is itself a stochastic process.

**Problem P3:** P3 is solved by using the solutions of P1 and P2. Notice that, P3, is motivated by the applications discussed in at the beginning of this chapter. P3, in turn, provides motivation for P1 and P2. However, P1 and P2 are also important in their own right.

Our idea of resorting to fluid and diffusion approximations instead of carrying out a transient analysis of the Markov chain  $\{X(t), t \geq 0\}$  is motivated by the following. A transient analysis of the Markov chain  $\{X(t), t \geq 0\}$  would be intractable and computationally intensive for realistic values of  $N_0$  at which the network would normally operate, say  $N_0 \geq 50$ . On the other hand, the accuracy of the fluid and the diffusion approximations would improve with the increase in the number of users,  $N_0$ . Furthermore, the diffusion approximations in terms of Brownian motions provide us with a linear system dynamics equation with Gaussian process noise and a linear measurement equation with Gaussian measurement noise, under which Kalman filtering is known to be optimal (in several different ways) [134, 135]. A brief background on Brownian motion has been provided in Appendix 5.9.2.

## 5.4 Fluid and Diffusion Approximations

We obtain the fluid and diffusion approximations for the process  $\{X(t), t \geq 0\}$  by invoking limit theorems for *density-dependent Markov chains* (see [136, Chapter 8], [137, Chapter 11], and also see Appendix 5.9.3 for the definition and a summary of main results). To that end, consider a sequence of networks, indexed by the parameter  $N = 1, 2, \dots$ , where the  $N$ -th network represents a DTN with  $N$  mobile users,  $S^{(N)}$  static sources and  $H^{(N)}$  static observers. Initially, the message is only available at the  $S^{(N)}$  sources. The source-user, observer-user and user-user inter-meeting times are independent and exponentially distributed random variables with parameters  $\beta_S^{(N)}$ ,  $\beta_H^{(N)}$ , and  $\beta_U^{(N)}$ , respectively. For the  $N$ -th network,  $X^{(N)}(t)$  represents the number of users that have the message at time  $t$ , with  $X^{(N)}(0) = 0$ , and  $Y^{(N)}(t)$  represents the total count of all the observers at time  $t$ , with  $Y^{(N)}(0) := 0$ . The link-layer error is modeled by the probabilities  $1 - p_S$  and  $1 - p_U$ , as before.

We obtain the fluid and diffusion approximations for the process  $\{X(t), t \geq 0\}$  by recognizing that the sequence  $\{X^{(N)}(t), t \geq 0\}$  represents a family of density-dependent Markov chains (see Definition 5.9.2 in Appendix 5.9.3). Notice that, *the network corresponding to  $N = N_0$  represents our DTN of interest containing  $N_0$  mobile users*. Thus, we have  $X^{(N_0)}(t) = X(t)$ ,  $Y^{(N_0)}(t) = Y(t)$ ,  $S^{(N_0)} = S_0$ ,  $H^{(N_0)} = H_0$ ,  $\beta_S^{(N_0)} = \beta_S$  and so on.

Since the observation  $\{Y(t)\}$  is driven by the process  $\{X(t)\}$ , it is natural to expect that the fluid and diffusion approximations of the observation  $\{Y(t)\}$  should be related to that of

the process  $\{X(t)\}$ . We obtain the diffusion approximations of the observation  $\{Y(t)\}$  from that of the process  $\{X(t)\}$  by the *Continuous Mapping Approach* (see Appendix 5.9.4).

#### 5.4.1 Fluid Approximation for the Spreading Process

**Direct Delivery:** First consider the sequence  $\{X^{(N)}(t), t \geq 0\}$  of Markov chains in the case of direct delivery, where  $\{X^{(N)}(t)\}$  has state space  $\mathcal{S}^{(N)} = \{0, 1, \dots, N\}$  and transition rate matrix  $\mathcal{Q}^{(N)} = [q^{(N)}(X, X')]_{X, X' \in \mathcal{S}^{(N)}}$  given by

$$q^{(N)}(X, X+l) = \begin{cases} p_S \beta_S^{(N)} S^{(N)}(N-X), & \text{if } l = +1, \\ -p_S \beta_S^{(N)} S^{(N)}(N-X), & \text{if } l = 0, \\ 0, & \text{otherwise.} \end{cases}$$

The only positive transition rate of  $\{X^{(N)}(t)\}$ , which is associated with a jump of +1, can be rewritten as

$$q^{(N)}(X, X+1) = N \left( p_S \cdot N \beta_S^{(N)} \cdot \frac{S^{(N)}}{N} \right) \left( 1 - \frac{X}{N} \right).$$

To be able to apply Theorems 5.9.1 and 5.9.2 (see Appendix 5.9.3), we stipulate that, for all  $N$ ,

$$N \beta_S^{(N)} = N_0 \beta_S =: \lambda'_S, \quad \text{a constant, and}$$

$$\frac{S^{(N)}}{N} = \frac{S_0}{N_0} =: s_0, \quad \text{a constant, so that}$$

$$p_S \beta_S^{(N)} S^{(N)} = p_S \beta_S S_0 = p_S \lambda'_S s_0 =: \lambda_S, \quad \text{a constant.}$$

This amounts to saying that, in the sequence of networks indexed by  $N$ , the total rate at which a source meets with all users remains a constant and that the ratio of the number of sources to the number of users remains a constant.

Then, the transition rate associated with a jump of +1, can be written in a density-dependent form [136, 137] as

$$q^{(N)}(X, X+1) = N f_{+1} \left( \frac{X}{N} \right),$$

where

$$f_{+1}(w) = \lambda_S (1 - w). \tag{5.2}$$

Defining

$$x^{(N)}(t) = \frac{X^{(N)}(t)}{N}, \tag{5.3}$$

and applying Theorem 5.9.1 (see Appendix 5.9.3), we observe that, as  $N \rightarrow \infty$ , the sequence  $\{x^{(N)}(t), t \geq 0\}$ ,  $N = 1, 2, \dots$ , converges uniformly over compact sets, almost surely, to the solution  $x(t)$  of the ordinary differential equation (ODE)

$$\frac{dx(t)}{dt} = F_S(x(t)), \quad (5.4)$$

with initial condition  $x(0) = 0$ , where  $F_S(w)$  denotes the drift function corresponding to the “spreading” process  $\{X(t)\}$ . Since there is only one positive transition rate of  $\{X^{(N)}(t)\}$ , which is associated with a jump of  $+1$ , we have

$$F_S(w) = f_{+1}(w), \quad (5.5)$$

where  $f_{+1}(w)$  is given by (5.2). Solving (5.4) with initial condition  $x(0) = 0$ , we obtain

$$x(t) = 1 - e^{-\lambda_S t}, \quad \forall t, 0 \leq t < \infty. \quad (5.6)$$

The deterministic fluid approximation for the process  $\{X(t)\}$  is given by

$$X(t) = X^{(N_0)}(t) \approx N_0 x(t), \quad (5.7)$$

where  $x(t)$  is given by (5.6). When  $N_0$  is large, the fluid approximation provides very good first-order approximation for the actual process  $\{X(t)\}$ .

Note that, when measurements are not available or are difficult/expensive to collect, then the fluid approximation is the best that one can rely on.

**Epidemic Routing:** Next, consider the sequence  $\{X^{(N)}(t), t \geq 0\}$  of Markov chains in the case of epidemic routing, where  $\{X^{(N)}(t)\}$  has state space  $\mathcal{S}^{(N)} = \{0, 1, \dots, N\}$  and transition rate matrix  $\mathcal{Q}^{(N)} = [q^{(N)}(X, X')]_{X, X' \in \mathcal{S}^{(N)}}$  given by

$$q^{(N)}(X, X+l) = \begin{cases} (p_S \beta_S^{(N)} S^{(N)} + p_U \beta_U^{(N)} X) (N - X), & \text{if } l = +1, \\ - (p_S \beta_S^{(N)} S^{(N)} + p_U \beta_U^{(N)} X) (N - X), & \text{if } l = 0, \\ 0, & \text{otherwise.} \end{cases}$$

The only positive transition rate of  $\{X^{(N)}(t)\}$ , which is associated with a jump of  $+1$ , can be rewritten as

$$q^{(N)}(X, X+1) = N \left( p_S \cdot N \beta_S^{(N)} \cdot \frac{S^{(N)}}{N} + p_U \cdot N \beta_U^{(N)} \cdot \frac{X}{N} \right) \left( 1 - \frac{X}{N} \right).$$

As before, we stipulate that, for all  $N$ ,

$$N \beta_S^{(N)} = N_0 \beta_S =: \lambda'_S, \quad \text{a constant, and}$$

$$\begin{aligned}
N\beta_U^{(N)} &= N_0\beta_U =: \lambda'_U, \quad \text{a constant, and} \\
\frac{S^{(N)}}{N} &= \frac{S_0}{N_0} =: s_0, \quad \text{a constant, so that} \\
p_S\beta_S^{(N)}S^{(N)} &= p_S\beta_SS_0 = p_S\lambda'_Ss_0 =: \lambda_S, \quad \text{a constant, and} \\
p_UN\beta_U^{(N)} &= p_UN_0\beta_U = p_U\lambda'_U =: \lambda_U, \quad \text{a constant.}
\end{aligned}$$

Then, the transition rate associated with a jump of +1, can be written in a density-dependent form [136, 137] as

$$q^{(N)}(X, X+1) = Nf_{+1}\left(\frac{X}{N}\right),$$

where

$$f_{+1}(w) = (\lambda_S + \lambda_U w)(1 - w). \quad (5.8)$$

Equations (5.3), (5.4), (5.5) and (5.7) apply in this case as well with  $f_+(\cdot)$  given by (5.8) and  $x(t)$  given by

$$x(t) = 1 - \frac{\lambda_U + \lambda_S}{\lambda_U + \lambda_S e^{(\lambda_U + \lambda_S)t}}, \quad \forall t, 0 \leq t < \infty. \quad (5.9)$$

#### 5.4.2 Diffusion Approximation for the Spreading Process

Defining

$$v_x^{(N)}(t) = \sqrt{N} \left( x^{(N)}(t) - x(t) \right), \quad (5.10)$$

where  $x^{(N)}(t)$  is given by (5.3), and  $x(t)$  is given by (5.6) in the case of direct delivery and by (5.9) in the case of epidemic routing, respectively, we observe that,  $v_x^{(N)}(0) = 0$ , for all  $N$ , since  $x^{(N)}(0) = x(0) = 0$ . Then, applying Theorem 5.9.2 (see Appendix 5.9.3), we observe that, as  $N \rightarrow \infty$ , the sequence  $\{v_x^{(N)}(t), t \geq 0\}$ ,  $N = 1, 2, \dots$ , converges in distribution to the solution  $v_x(t)$  of the equation

$$v_x(t) = B_S \left( \int_0^t f_{+1}(x(u)) du \right) + \int_0^t F'_S(x(u)) v_x(u) du, \quad (5.11)$$

with initial condition  $v_x(0) = 0$ , where  $B_S(\cdot)$  is a standard Brownian motion that captures the randomness of the “spreading” process,  $f_+(\cdot)$  is given by (5.2) in the case of direct delivery and by (5.8) in the case of epidemic routing, and

$$F'_S(w) = \frac{dF_S(w)}{dw} = \begin{cases} -\lambda_S & \text{(Direct Delivery)} \\ \lambda_U - \lambda_S - 2\lambda_U w & \text{(Epidemic Routing)} \end{cases} \quad (5.12)$$

represents the derivative of the drift function  $F_S$ .



To solve (5.11), we consider the equivalent (in distribution) stochastic differential equation (SDE)

$$dv_x(t) = F'_S(x(t))v_x(t)dt + \sqrt{f_{+1}(x(t))}dB_S(t), \quad (5.13)$$

then solve it with initial condition  $v_x(0) = 0$ , and obtain (see [138, Page 354])

$$v_x(t) = \Phi(t) \int_0^t \Phi^{-1}(u) \sqrt{f_{+1}(x(u))} dB_S(u), \quad (5.14)$$

where  $\Phi(t)$  is the solution to the ODE

$$\frac{d\Phi(t)}{dt} = F'_S(x(t))\Phi(t), \quad \Phi(0) = 1,$$

and hence,  $\Phi(t)$  is given by

$$\Phi(t) = \exp\left(\int_0^t F'_S(x(u))du\right) = \begin{cases} e^{-\lambda_S t} & \text{(Direct Delivery)} \\ \frac{e^{(\lambda_U + \lambda_S)t}(\lambda_U + \lambda_S)^2}{(\lambda_U + \lambda_S e^{(\lambda_U + \lambda_S)t})^2} & \text{(Epidemic Routing)}. \end{cases} \quad (5.15)$$

The diffusion approximation for the process  $\{X(t)\}$  is given by

$$X(t) \approx N_0 x(t) + \sqrt{N_0} v_x(t). \quad (5.16)$$

### 5.4.3 Diffusion Approximation for the Observation

Since  $Y^{(N)}(t)$  denotes the total observer count in the  $N$ -th network, we write

$$\begin{aligned} Y^{(N)}(t) &= \mathcal{P}\left(\beta_H^{(N)} H^{(N)} \int_0^t X^{(N)}(u) du\right) \\ &= \mathcal{P}\left(\lambda_H \int_0^t X^{(N)}(u) du\right), \end{aligned} \quad (5.17)$$

where we stipulate that, for all  $N \geq 1$ ,

$$N\beta_H^{(N)} = N_0\beta_H =: \lambda'_H, \quad \text{a constant, and}$$

$$\frac{H^{(N)}}{N} = \frac{H_0}{N_0} =: h_0, \quad \text{a constant, so that}$$

$$\beta_H^{(N)} H^{(N)} = \beta_H H_0 = \lambda'_H h_0 =: \lambda_H, \quad \text{a constant.}$$

This amounts to saying that, in the sequence of networks indexed by  $N$ , the total rate at which an observer meets with all users remains a constant and that the ratio of the number of observers to the number of users remains a constant.

We define the following:

$$y(t) := \lambda_H \int_0^t x(u) du, \quad (5.18)$$

$$y^{(N)}(t) := \frac{Y^{(N)}(t)}{N}, \quad \text{and} \quad (5.19)$$

$$v_y^{(N)}(t) := \sqrt{N} \left( y^{(N)}(t) - y(t) \right) = \left( \frac{Y^{(N)}(t)}{\sqrt{N}} - \sqrt{N} y(t) \right). \quad (5.20)$$

It can be shown that

$$y(t) = \begin{cases} \lambda_H \left( t - \frac{1}{\lambda_S} (1 - e^{-\lambda_S t}) \right) & \text{(Direct Delivery)} \\ \frac{\lambda_H}{\lambda_U} \left( \ln \left( \frac{\lambda_U + \lambda_S e^{(\lambda_U + \lambda_S)t}}{\lambda_U + \lambda_S} \right) - \lambda_S t \right) & \text{(Epidemic Routing).} \end{cases} \quad (5.21)$$

We observe that  $v_y^{(N)}(t)$  can be rewritten as

$$v_y^{(N)}(t) = (\phi_1^{(N)} \circ \phi_2^{(N)})(t) - c_N y(t),$$

where ‘ $\circ$ ’ denotes the composition operator,  $c_N = \sqrt{N}$ ,  $\phi_1^{(N)}(t) = \frac{\mathcal{P}(Nt)}{\sqrt{N}}$ , and  $\phi_2^{(N)}(t) = \lambda_H \int_0^t x^{(N)}(u) du$ , where  $x^{(N)}(t)$  is given by (5.3). Next, we observe that, as  $N \rightarrow \infty$ , we have,

$$(\phi_1^{(N)} - c_N \iota)(t) = \frac{\mathcal{P}(Nt) - Nt}{\sqrt{N}} \Rightarrow B_H(t),$$

where ‘ $\iota$ ’ denotes the identity operator,  $\Rightarrow$  denotes convergence in distribution, and  $B_H(\cdot)$  is a standard Brownian motion that captures the randomness of the observer-user meetings. Note that  $B_H(\cdot)$  is independent of  $B_S(\cdot)$  (which captures the randomness in source-user and user-user meetings).

By the continuity of the integration operator, as  $N \rightarrow \infty$ , we also have,

$$c_N (\phi_2^{(N)} - y)(t) = \lambda_H \int_0^t v_x^{(N)}(u) du \Rightarrow \lambda_H \int_0^t v_x(u) du,$$

where  $v_x^{(N)}(t)$  and  $v_x(t)$  are given by (5.10) and (5.11), respectively.

Recalling the independence of  $\{\mathcal{P}(t)\}$  and  $\{X(t)\}$ , we observe that  $\phi_1^{(N)}(t)$  and  $\phi_2^{(N)}(t)$  are independent processes for all  $N \geq 1$ . Then, applying the joint convergence theorem (see Theorem 5.9.4 in Appendix 5.9.4), we obtain, as  $N \rightarrow \infty$ ,

$$\left( (\phi_1^{(N)} - c_N \iota), c_N (\phi_2^{(N)} - y) \right) \Rightarrow \left( B_H, \lambda_H \int v_x(u) du \right).$$

Next, we observe that,  $x^{(N)}(t) \rightarrow x(t)$ , almost surely, implies that  $x^{(N)}(t) \Rightarrow x(t)$ , since almost sure convergence implies convergence in distribution. Then, by continuous mapping, as  $N \rightarrow \infty$ , we have

$$\phi_2^{(N)}(t) \Rightarrow y(t).$$

Observing that  $y(t)$  is a deterministic function, we apply Theorem 5.9.5 (see Appendix 5.9.4), to conclude that

$$\left( \left( \phi_1^{(N)} - c_N t \right), \phi_2^{(N)}, c_N \left( \phi_2^{(N)} - y \right) \right) \Rightarrow \left( B_H, y, \lambda_H \int v_x(u) du \right).$$

Finally, observing that  $y(t)$  is strictly increasing, we apply Theorem 5.9.3 (see Appendix 5.9.4), and obtain,  $v_y^{(N)}(t) \Rightarrow v_y(t)$ , as  $N \rightarrow \infty$ , where

$$\begin{aligned} v_y(t) &= B_H(y(t)) + \lambda_H \int_0^t v_x(u) du \\ &= \int_0^t \sqrt{\lambda_H x(u)} dB_H(u) + \lambda_H \tilde{\Phi}(t) \Phi^{-1}(t) v_x(t) \\ &\quad - \lambda_H \int_0^t \tilde{\Phi}(u) \Phi^{-1}(u) \sqrt{f_{+1}(x(u))} dB_S(u), \end{aligned} \quad (5.22)$$

where

$$\tilde{\Phi}(t) := \int_0^t \Phi(u) du = \begin{cases} \frac{1 - e^{-\lambda_S t}}{\lambda_S} & \text{(Direct Delivery)} \\ \frac{1}{\lambda_S} \left( 1 - \frac{\lambda_U + \lambda_S}{\lambda_U + \lambda_S e^{(\lambda_U + \lambda_S)t}} \right) & \text{(Epidemic Routing).} \end{cases} \quad (5.23)$$

The diffusion approximation for the observation  $\{Y(t)\}$  is given by

$$Y(t) \approx N_0 y(t) + \sqrt{N_0} v_y(t). \quad (5.24)$$

## 5.5 Discrete Time Kalman Filtering

In this Section, we derive discrete time dynamical equations for the fluctuation processes  $\{v_x(t)\}$  and  $\{v_y(t)\}$  by sampling at discrete times given by an increasing sequence  $\{T_k, k \geq 0\}$ , with  $T_0 = 0$ . Defining

$$v_{x,k} := v_x(T_k), \quad k = 0, 1, 2, \dots,$$

where  $0 < T_1 < T_2 < \dots$ , denote the times at which we want to track the process  $X(t)$ , we obtain the *system dynamic equation* from (5.14) as:

$$v_{x,k+1} = \alpha_k v_{x,k} + w_k, \quad k = 0, 1, 2, \dots, \quad (5.25)$$

where

$$\begin{aligned} \alpha_k &= \Phi(T_{k+1}) \Phi^{-1}(T_k) \\ &= \begin{cases} e^{-\lambda_S(T_{k+1}-T_k)} & \text{(Direct Delivery)} \\ \frac{e^{(\lambda_U + \lambda_S)T_{k+1}} (\lambda_U + \lambda_S e^{(\lambda_U + \lambda_S)T_k})^2}{e^{(\lambda_U + \lambda_S)T_k} (\lambda_U + \lambda_S e^{(\lambda_U + \lambda_S)T_{k+1}})^2} & \text{(Epidemic Routing)} \end{cases} \end{aligned} \quad (5.26)$$

and

$$w_k = \Phi(T_{k+1}) \int_{T_k}^{T_{k+1}} \Phi^{-1}(u) \sqrt{f_{+1}(x(u))} dB_S(u). \quad (5.27)$$

Defining

$$v_{y,k} := v_y(T_k), \quad k = 0, 1, 2, \dots,$$

we obtain from (5.22) the *measurement equation* as:

$$v_{y,k} = \gamma_k v_{x,k} + z_k, \quad k = 0, 1, 2, \dots, \quad (5.28)$$

where

$$\begin{aligned} \gamma_k &= \lambda_H \tilde{\Phi}(T_k) \Phi^{-1}(T_k) \\ &= \begin{cases} \frac{\lambda_H}{\lambda_S} (e^{\lambda_S T_k} - 1) & \text{(Direct Delivery)} \\ \frac{\lambda_H (\lambda_U + \lambda_S e^{(\lambda_U + \lambda_S) T_k})^2 (1 - \frac{\lambda_U + \lambda_S}{\lambda_U + \lambda_S e^{(\lambda_U + \lambda_S) T_k}})}{\lambda_S (\lambda_U + \lambda_S)^2 e^{(\lambda_U + \lambda_S) T_k}} & \text{(Epidemic Routing)} \end{cases} \end{aligned} \quad (5.29)$$

and  $z_k = r_k + s_k$ , where

$$\begin{aligned} r_k &= -\lambda_H \int_0^{T_k} \tilde{\Phi}(u) \Phi^{-1}(u) \sqrt{f_{+1}(x(u))} dB_S(u), \quad \text{and} \\ s_k &= \int_0^{T_k} \sqrt{\lambda_H x(u)} dB_H(u). \end{aligned}$$

Defining,  $n_k := n_{1,k} + n_{2,k}$ , where

$$\begin{aligned} n_{1,k} &= -\lambda_H \int_{T_k}^{T_{k+1}} \tilde{\Phi}(u) \Phi^{-1}(u) \sqrt{f_{+1}(x(u))} dB_S(u), \\ n_{2,k} &= \int_{T_k}^{T_{k+1}} \sqrt{\lambda_H x(u)} dB_H(u), \end{aligned}$$

we obtain,  $r_{k+1} = r_k + n_{1,k}$ ,  $s_{k+1} = s_k + n_{2,k}$  and

$$z_{k+1} = z_k + n_k. \quad (5.30)$$

Notice that, the process noise sequence  $\{w_k\}$  is white, but the measurement noise sequence  $\{z_k\}$  is *colored*, or, more precisely, *sequentially correlated* [139]. With such measurement noise, we adopt the *measurement differencing approach* [134, 139]. We define, for  $k \geq 1$ ,

$$v'_{y,k} := v_{y,k} - v_{y,k-1} \quad (5.31)$$

with  $v_y(0) = 0$ , and derive the *modified measurement equation* as:

$$\begin{aligned}
v'_{y,k+1} &= v_{y,k+1} - v_{y,k} \\
&= \gamma_{k+1}v_{x,k+1} + z_{k+1} - \gamma_k v_{x,k} - z_k \\
&= \gamma_{k+1}(\alpha_k v_{x,k} + w_k) + z_{k+1} - \gamma_k v_{x,k} - z_k \\
&= \gamma'_k v_{x,k} + z'_k,
\end{aligned} \tag{5.32}$$

where

$$\begin{aligned}
\gamma'_k &= \gamma_{k+1}\alpha_k - \gamma_k \\
&= \begin{cases} \frac{\lambda_H}{\lambda_S} (1 - e^{-\lambda_S(T_{k+1}-T_k)}) & \text{(Direct Delivery)} \\ \frac{\lambda_H(\lambda_U + \lambda_S e^{(\lambda_U + \lambda_S)T_k})((\lambda_U + \lambda_S e^{(\lambda_U + \lambda_S)T_{k+1}}) - (\lambda_U + \lambda_S e^{(\lambda_U + \lambda_S)T_k}))}{\lambda_S(\lambda_U + \lambda_S)e^{(\lambda_U + \lambda_S)T_k}(\lambda_U + \lambda_S e^{(\lambda_U + \lambda_S)T_{k+1}})} & \text{(Epidemic Routing)} \end{cases} \tag{5.33}
\end{aligned}$$

and

$$z'_k = \gamma_{k+1}w_k + n_k. \tag{5.34}$$

Notice that the process noise sequence  $\{w_k\}$  and the modified measurement noise sequence  $\{z'_k\}$  are white with

$$w_k \sim \mathcal{N}(0, Q_k), \quad \text{and} \quad z'_k \sim \mathcal{N}(0, R_k),$$

where  $Q_k := E[w_k^2]$ , and  $R_k := E[(z'_k)^2]$ . However, the modified measurement noise is correlated with the process noise. We have  $E[w_k z'_l] = C_k \delta_{kl}$ , where  $\delta_{kl}$  denotes the Kronecker delta function which is equal to 1 if  $k = l$ , and equal to 0 otherwise. It is easy to see that

$$E[w_k z'_k] = \gamma_{k+1}E[w_k^2] + E[w_k n_k] \tag{5.35}$$

and

$$E[(z'_k)^2] = \gamma_{k+1}^2 E[w_k^2] + 2\gamma_{k+1}E[w_k n_k] + E[n_k^2]. \tag{5.36}$$

Defining  $E_k := e^{(\lambda_U + \lambda_S s_0)T_k}$ , it can be shown that

$$\begin{aligned}
E[w_k^2] &= \Phi^2(T_{k+1}) \int_{T_k}^{T_{k+1}} \Phi^{-2}(u) f_{+1}(x(u)) du \\
&= \begin{cases} e^{-2\lambda_S T_{k+1}} (e^{\lambda_S T_{k+1}} - e^{\lambda_S T_k}) & \text{(Direct Delivery)} \\ \left[ \lambda_S^2 (E_{k+1} - E_k) + \lambda_U^2 (E_k^{-1} - E_{k+1}^{-1}) + 2\lambda_U \lambda_S (\lambda_U + \lambda_S) (T_{k+1} - T_k) \right] \\ \times \frac{\lambda_S (\lambda_U + \lambda_S) E_{k+1}^2}{(\lambda_U + \lambda_S E_{k+1})^4} & \text{(Epidemic Routing)} \end{cases} \tag{5.37}
\end{aligned}$$

$$\begin{aligned}
E[w_k n_k] &= -\lambda_H \Phi(T_{k+1}) \int_{T_k}^{T_{k+1}} \tilde{\Phi}(u) \Phi^{-2}(u) f_{+1}(x(u)) du \\
&= \begin{cases} \lambda_H e^{-\lambda_S T_{k+1}} (T_{k+1} - T_k) - \frac{\lambda_H}{\lambda_S} e^{\lambda_S T_{k+1}} (e^{\lambda_S T_k} - e^{-\lambda_S T_k}) & \text{(Direct Delivery)} \\ \left[ \frac{\lambda_U}{(\lambda_U + \lambda_S)^2} \left( \frac{\lambda_U}{\lambda_U + \lambda_S} - 1 \right) (E_k^{-1} - E_{k+1}^{-1}) + \frac{\lambda_S}{\lambda_U + \lambda_S} \left( \frac{2\lambda_U}{\lambda_U + \lambda_S} - 1 \right) (T_{k+1} - T_k) \right. \\ \left. + \frac{(\lambda_S)^2}{(\lambda_U + \lambda_S)^3} (E_{k+1} - E_k) \right] \times \frac{-\lambda_H (\lambda_U + \lambda_S)^2 E_{k+1}}{(\lambda_U + \lambda_S E_{k+1})^2} & \text{(Epidemic Routing)} \end{cases}
\end{aligned} \tag{5.38}$$

and

$$\begin{aligned}
E[n_k^2] &= \lambda_H^2 \int_{T_k}^{T_{k+1}} \tilde{\Phi}^2(u) \Phi^{-2}(u) f_{+1}(x(u)) du + \lambda_H \int_{T_k}^{T_{k+1}} x(u) du \\
&= \begin{cases} \frac{\lambda_H^2}{\lambda_S^2} (e^{\lambda_S T_{k+1}} - e^{\lambda_S T_k}) + \lambda_H \left( 1 - \frac{2\lambda_H}{\lambda_S} \right) (T_{k+1} - T_k) \\ + \frac{\lambda_H}{\lambda_S} \left( 1 - \frac{\lambda_H}{\lambda_S} \right) (e^{-\lambda_S T_{k+1}} - e^{-\lambda_S T_k}) & \text{(Direct Delivery)} \\ \frac{\lambda_H^2}{\lambda_U + \lambda_S} \left[ \frac{1}{\lambda_S} \left( \frac{\lambda_U}{\lambda_U + \lambda_S} - 1 \right)^2 (E_k^{-1} - E_{k+1}^{-1}) + \left( \frac{2\lambda_U}{\lambda_U + \lambda_S} - 2 \right) (T_{k+1} - T_k) \right. \\ \left. + \frac{\lambda_S}{(\lambda_U + \lambda_S)^2} (E_{k+1} - E_k) \right] + \frac{\lambda_H}{\lambda_U} [\log(\lambda_U + \lambda_S E_{k+1}) - \log(\lambda_U + \lambda_S E_k)] \\ - \frac{\lambda_H}{\lambda_U} \lambda_S (T_{k+1} - T_k) & \text{(Epidemic Routing)} \end{cases}
\end{aligned} \tag{5.39}$$

Let  $\hat{v}_{x,k}^-$  and  $\hat{v}_{x,k}^+$  denote the estimates of  $v_{x,k}$  at time  $k$  before and after, respectively, taking into account the measurement  $v_{y,k}$ , i.e.,

$$\hat{v}_{x,k}^- = E[v_{x,k} | v_{y,1}, \dots, v_{y,k-1}],$$

$$\hat{v}_{x,k}^+ = E[v_{x,k} | v_{y,1}, \dots, v_{y,k}].$$

Let  $P_k^-$  and  $P_k^+$  denote the covariances of the corresponding estimation errors, i.e.,

$$P_k^- = E[(v_{x,k} - \hat{v}_{x,k}^-)^2], \quad P_k^+ = E[(v_{x,k} - \hat{v}_{x,k}^+)^2].$$

Let

$$\hat{v}_{y,k} := \sqrt{N_0}((\hat{Y}(T_k)/N_0) - y(T_k)),$$

where  $\hat{Y}(T_k)$  and  $y(T_k)$  denote the actual measurement (i.e., observers' total count) and the value of  $y(t)$ , respectively, at time  $T_k$ . Defining

$$\hat{v}'_{y,k} := \hat{v}_{y,k} - \hat{v}_{y,k-1},$$

we apply Kalman filtering with sequentially correlated measurement noise as follows:

**Step 1:** Start with  $\hat{v}_{x,0}^+ = 0$ ,  $P_0^+ = 0$  and  $v_y(0) = 0$  (so that  $v'_{y,k}$  can be computed for  $k \geq 1$  upon the availability of the measurement  $v_{y,k}$  for  $k \geq 1$ ).

**Step 2:** For  $k \geq 1$ , apply the measurement  $\hat{v}'_{y,k}$  to compute  $\hat{v}_{x,k}^+$  and  $P_k^+$  as (see [134, Equation 7.78])

$$\begin{aligned}\hat{v}_{x,k}^+ &= \alpha_{k-1}\hat{v}_{x,k-1}^+ + \left[\alpha_{k-1}P_{k-1}^+\gamma'_{k-1} + C_{k-1}\right] \left[(\gamma'_{k-1})^2P_{k-1}^+ + R_{k-1}\right]^{-1} \left(\hat{v}'_{y,k} - \gamma'_{k-1}\hat{v}_{x,k-1}^+\right) \\ P_k^+ &= \alpha_{k-1}^2P_{k-1}^+ + Q_{k-1} - \left[\alpha_{k-1}P_{k-1}^+\gamma'_{k-1} + C_{k-1}\right]^2 \left[(\gamma'_{k-1})^2P_{k-1}^+ + R_{k-1}\right]^{-1}\end{aligned}$$

We obtain the estimates for the process as

$$\hat{X}(T_k) = N_0x(T_k) + \sqrt{N_0}\hat{v}_{x,k}^+,$$

where  $\sqrt{N_0}\hat{v}_{x,k}^+$  provides an estimate of the fluctuation of the process about its mean, at time  $T_k$ .

**Choosing the Sequence of Discrete Times:** Our framework can be applied with any strictly increasing sequence  $0 < T_1 < T_2 < \dots < T_k < \dots$ . We consider two possibilities:

1. **Constant Intervals:** In this case,  $T_k = kT$  for some constant  $T > 0$ .
2. **Triggered by Observations:** One can choose to update the estimates at the observer-user meetings.

## 5.6 Linear Minimum Mean Squared Error Estimator

For the case of direct delivery, we derive the linear MMSE estimator for comparison purpose. Consider the correlated random variables  $X$  and  $Y$ , with their mean vector and covariance matrix given by

$$\begin{pmatrix} m_x \\ m_y \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} V_{xx} & V_{xy} \\ V_{yx} & V_{yy} \end{pmatrix},$$

respectively. The linear MMSE estimator is obtained by applying Proposition 5.6.1.

**Proposition 5.6.1.** *The linear estimator of  $X$  given  $Y$  which minimizes the expected square estimation error is given by*

$$E[X|Y] = m_x + V_{xy}V_{yy}^{-1}(Y - m_y) . \quad \square$$

Let  $T_\lambda^j$  denote the time at which user  $j$  receives a copy of the file. Note that  $T_\lambda^j$  is exponentially distributed with parameter  $\lambda_S = p_S S_0 \beta_S$ . Then, the probability  $p(t)$  that a user has a copy of the file at time  $t$  is given by

$$p(t) = P(T_\lambda^j \leq t) = 1 - \exp(-\lambda_S t) . \quad (5.40)$$

Let  $\xi_j(t)$  denote the indicator variable that takes the value 1 if user  $j$ ,  $j = 1, \dots, N_0$ , has a copy of the file at time  $t$ , and 0 otherwise. Then, we have

$$X(t) = \sum_{j=1}^{N_0} \xi_j(t) . \quad (5.41)$$

Clearly, for each  $j$ ,  $j = 1, \dots, N_0$ , and for each  $t \geq 0$ ,  $\xi_j(t)$  is a *Bernoulli* random variable with  $P(\xi_j(t) = 1) = p(t)$  and  $P(\xi_j(t) = 0) = 1 - p(t)$ . By independence of source-relay meeting events, for each  $t \geq 0$ , the random variables  $\xi_j(t)$ ,  $j = 1, \dots, N_0$ , are i.i.d., and we conclude that  $X(t)$  has a *Binomial distribution with parameters  $N_0$  and  $p(t)$* , i.e.,

$$P(X(t) = k) = \binom{N_0}{k} p(t)^k (1 - p(t))^{N_0 - k} . \quad (5.42)$$

Given the process  $\{X(t), t \geq 0\}$ , the measurement,  $\{Y(t), t \geq 0\}$ , is a *doubly-stochastic Poisson process* [133] with (stochastic) intensity function  $\lambda_H X(t)$ . Thus, the count of the observer,  $Y(t)$ , has a Poisson distribution with parameter

$$\theta_y(t) = \lambda_H \int_0^t X(u) du . \quad (5.43)$$

We emphasize that,  $\forall t \geq 0$ ,  $\theta_y(t)$  is a random variable, since  $X(t)$  is stochastic.

Next in Section 5.6.1, guided by Proposition 5.6.1, we derive the quantities  $m_x$ ,  $m_y$ ,  $V_{xx}$ , and  $V_{xy}$  as functions of time.

### 5.6.1 Derivation of the Means and (Co)variances

**Lemma 5.6.1.** (i)  $X(t)$  has mean  $m_x(t)$  and variance  $V_{xx}(t)$  given by

$$m_x(t) = N_0 p(t) , \quad V_{xx}(t) = N_0 p(t)(1 - p(t)) = m_x(t)(1 - p(t)) .$$

(ii)  $Y(t)$  has mean  $m_y(t)$  and variance  $V_{yy}(t)$  given by

$$m_y(t) = \lambda_H m_x(t) E[T_x(t)] ,$$

and

$$V_{yy}(t) = m_y(t) + m_x(t) \lambda_H^2 \left( E[T_x^2(t)] - (E[T_x(t)])^2 \right) + \lambda_H^2 E[T_x^2(t)] V_{xx}(t) ,$$



where

$$E[T_x(t)] = \frac{t}{1 - \exp(-\lambda_S t)} - \frac{1}{\lambda_S} \quad , \quad E[T_x^2(t)] = \frac{\exp(\lambda_S t)}{\lambda_S^3} (\lambda_S^2 t^2 - 2\lambda_S t + 2) - \frac{2}{\lambda_S^3} \quad .$$

(iii) The covariance between  $X(t)$  and  $Y(t)$  is given by

$$V_{yx}(t) = m_y(t)(1 - p(t)) \quad .$$

*Proof.* (i) Follows directly from (5.42).

(ii) Substituting (5.41) in (5.43), and interchanging the order of summation and integration, we obtain

$$\theta_y(t) = \sum_{i=1}^{N_0} \lambda_H \int_0^t \xi_i(u) du = \sum_{i=1}^{N_0} \lambda_H \max(t - T_\lambda^i, 0) = \sum_{i=1}^{X(t)} \lambda_H T_x^i(t) \quad ,$$

where  $T_x^i(t)$  are i.i.d. random variables, each distributed like the truncated random variable  $T_x(t)$  (truncated at  $t$ ) with the following distribution:

$$P(T_x(t) > a) = P(t - T_\lambda^i > a | T_\lambda^i \leq t) = \frac{1 - \exp(-\lambda_S(t - a))}{1 - \exp(-\lambda_S t)} \quad \text{for } 0 \leq a \leq t \quad . \quad (5.44)$$

$E[T_x(t)]$  and  $E[T_x^2(t)]$  as given above follow from (5.44). Then,

$$m_y(t) = E[Y(t)] = E_X[E_Y[Y(t)|X(t)]] = E[\theta_y(t)] = \lambda_H m_x(t) E[T_x(t)] \quad . \quad (5.45)$$

$$\begin{aligned} V_{yy}(t) &= \text{Var}(Y(t)) = E_X[\text{Var}_Y(Y(t)|X(t))] + \text{Var}_X(E_Y[Y(t)|X(t)]) \\ &= E[\theta_y(t)] + \text{Var}(\theta_y(t)) \quad , \end{aligned} \quad (5.46)$$

since the variance of a Poisson random variable is equal to its mean. As before,  $E[\theta_y(t)] = m_y(t)$  and  $\text{Var}(\theta_y(t))$  is obtained as follows:

$$\begin{aligned} \text{Var}(\theta_y(t)) &= E[X(t)] \text{Var}(\lambda_H T_x(t)) + E[\lambda_H^2 T_x(t)^2] \text{Var}(X(t)) \\ &= m_x(t) \lambda_H^2 \left( E[T_x^2(t)] - (E[T_x(t)])^2 \right) + \lambda_H^2 E[T_x^2(t)] V_{xx}(t) \quad . \end{aligned} \quad (5.47)$$

(iii) We have

$$\begin{aligned} V_{yx}(t) &= V_{xy}(t) = E[X(t)Y(t)] - E[X(t)]E[Y(t)] \\ &= E_X[E_Y[X(t)Y(t)|X(t)]] - m_x(t)m_y(t) \\ &= E[X(t)\theta_y(t)] - m_x(t)m_y(t) = E[X^2(t)]\lambda_H E[T_x(t)] - m_x(t)m_y(t) \\ &= \left( V_{xx}(t) + (m_x(t))^2 \right) \lambda_H E[T_x(t)] - m_x(t)m_y(t) = m_y(t)(1 - p(t)) \end{aligned} \quad (5.48)$$

□

## 5.7 Performance of Analytical Prediction and Estimation

In this section, we evaluate: (i) the quality of estimation provided by the Kalman filter, (ii) compare the quality of estimation with the linear MMSE estimator (in the direct delivery case), and (iii) the accuracy of the predictions about the level-crossing times based on the estimation. We also comment on the usefulness of the fluid model of the process.

We track the process  $\{X(t)\}$  at observer-user meetings.

### 5.7.1 Direct Delivery

We simulate the DTN in the case of direct delivery for the following scenarios:

<b>Scenario 1:</b>	$S_0 = 1,$	$N_0 = 50,$	$\beta_S = 0.002,$	$\lambda_H = \lambda_S.$
<b>Scenario 2:</b>	$S_0 = 1,$	$N_0 = 50,$	$\beta_S = 0.002,$	$\lambda_H = 10\lambda_S.$
<b>Scenario 3:</b>	$S_0 = 1,$	$N_0 = 50,$	$\beta_S = 0.002,$	$\lambda_H = 100\lambda_S.$
<b>Scenario 4:</b>	$S_0 = 1,$	$N_0 = 50,$	$\beta_S = 0.002,$	$\lambda_H = 1000\lambda_S.$
<b>Scenario 5:</b>	$S_0 = 1,$	$N_0 = 200,$	$\beta_S = 0.002,$	$\lambda_H = \lambda_S.$
<b>Scenario 6:</b>	$S_0 = 1,$	$N_0 = 1000,$	$\beta_S = 0.002,$	$\lambda_H = \lambda_S.$

In Figure 5.2 we depict the performance of the MMSE estimator and the Kalman filter for Scenario 1. We note that the estimations by both the MMSE estimator and the Kalman filter are very close to each other and indeed close to the fluid approximation  $N_0x(t)$  of  $X(t)$ . In Figure 5.3, we show the estimations of the fluctuations about the fluid limit for Scenario 1, and notice that neither the MMSE estimator nor the Kalman filter is able to successfully track the fluctuations in this scenario. We suspect that the inability to track the fluctuations in Scenario 1 is primarily due to the insufficiency of measurement data.

To verify if the inability to track the fluctuations in Scenario 1 is indeed due to the insufficiency of measurement data, we examine Scenario 2 (Figures 5.4 and 5.5), Scenario 3 (Figures 5.6 and 5.7) and Scenario 4 (Figures 5.8 and 5.9). In Scenarios 2, 3, and 4 we increase the rate  $\lambda_H$  at which measurements are taken by the observer by a factor 10, 100, and 1000, respectively. This increase can be achieved by increasing the number of observers or increasing the observer-user meeting rate or both. We observe that the performance of the Kalman filter is much improved in Scenario 2 with faster measurements. Further increase in  $\lambda_H$  results in better tracking of the fluctuations by the Kalman filter, and hence, extremely accurate tracking of the process itself.

In Figures 5.2-5.9, we observe that the MMSE estimator fails to make use of faster measurements. In fact, it stays very close to the fluid approximation. This can be explained as follows. The MMSE estimation differs from the fluid approximation  $N_0x(t)$  (which is equal to  $m_x(t)$ ) by the term  $V_{xy}(t)V_{yy}(t)^{-1}(Y(t) - m_y(t))$ . From the expressions for  $V_{xy}(t)$  and  $V_{yy}(t)$  in Sect. 5.6.1, it can be seen that  $V_{xy}(t)$  initially increases sub-linearly, but then

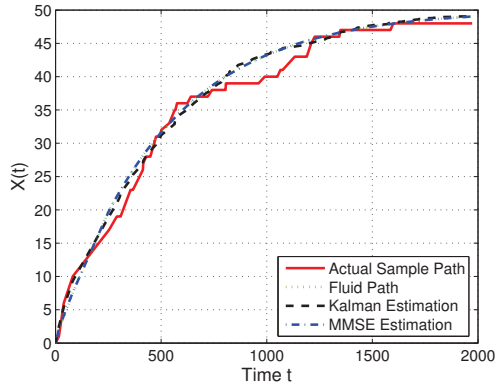


Figure 5.2: Performance of MMSE estimation and Kalman filter estimation of the process for Scenario 1.

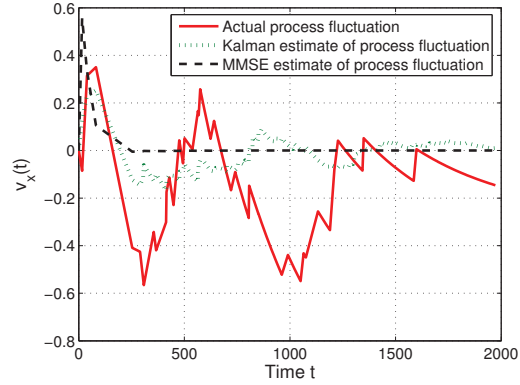


Figure 5.3: MMSE estimation and Kalman filter estimation of the process fluctuations for Scenario 1.

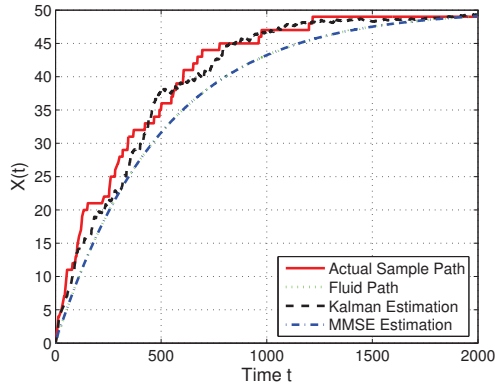


Figure 5.4: Performance of MMSE estimation and Kalman filter estimation of the process for Scenario 2.

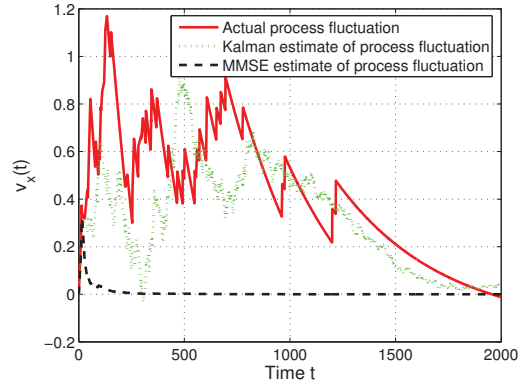


Figure 5.5: MMSE estimation and Kalman filter estimation of the process fluctuations for Scenario 2.

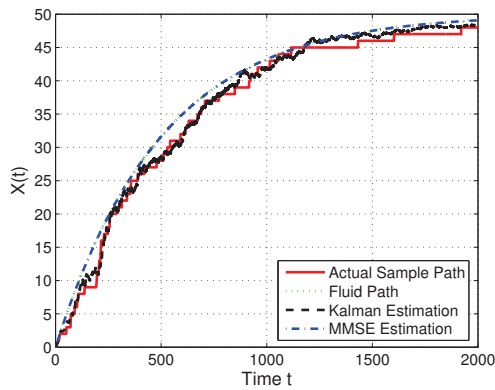


Figure 5.6: Performance of MMSE estimation and Kalman filter estimation of the process for Scenario 3.

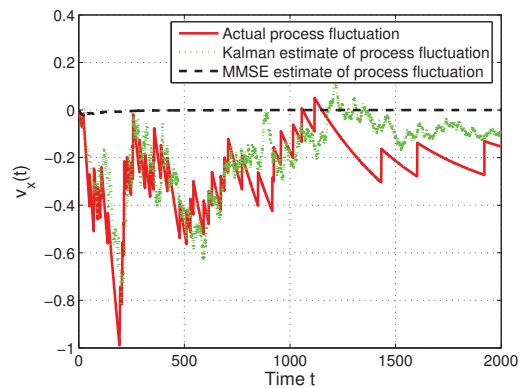


Figure 5.7: MMSE estimation and Kalman filter estimation of the process fluctuations for Scenario 3.

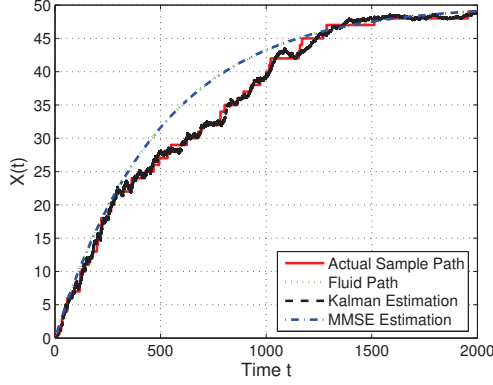


Figure 5.8: Performance of MMSE estimation and Kalman filter estimation of the process for Scenario 4.

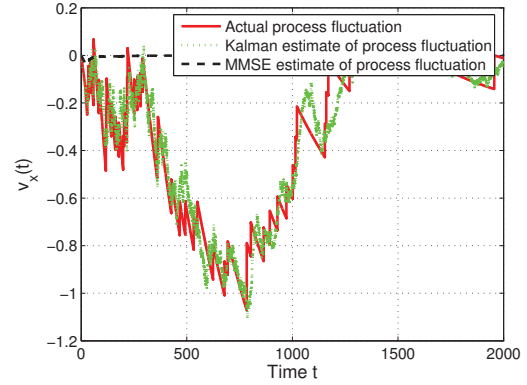


Figure 5.9: MMSE estimation and Kalman filter estimation of the process fluctuations for Scenario 4.

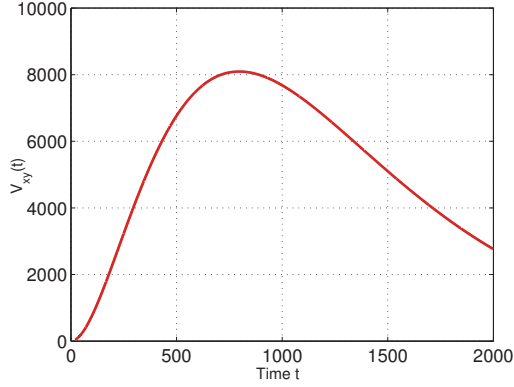


Figure 5.10:  $V_{xy}(t)$  as a function of time.

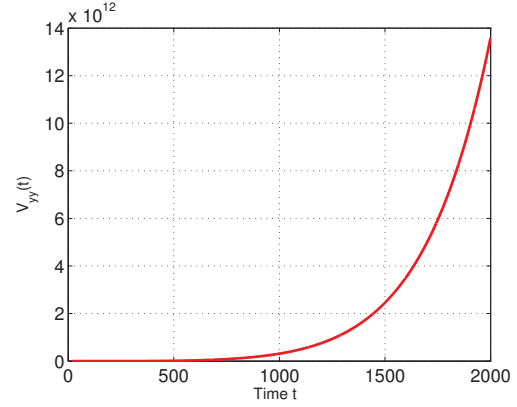


Figure 5.11:  $V_{yy}(t)$  as a function of time.

quickly decreases exponentially with  $t$  (see Fig. 5.10), and  $V_{yy}(t)$  increases super-linearly with  $t$  (see Fig. 5.11). Thus, except for an initial phase, the effect of the measurement  $(Y(t) - m_y(t))$  is diminished by the factor  $V_{xy}(t)V_{yy}(t)^{-1}$ . Increasing  $\lambda_H$  by a factor  $K$  increases  $V_{xy}(t)$  by a factor  $K$ , but also increases  $V_{yy}(t)$  by a factor  $K^2$ . Thus, increasing  $\lambda_H$  by a factor  $K$  results in an overall attenuation of the measurement  $(Y(t) - m_y(t))$  by a factor  $K$  (see Sect. 5.6.1). Furthermore, the difference between the measurement  $Y(t)$  and its mean  $m_y(t)$  also decreases with  $t$ . In summary, we can expect the performance of the MMSE estimator to get worse with time.

Next, we examine the situations in which the fluid approximation itself can be used as a good predictor. Suppose that we increase the area of the network by a factor  $K$  keeping the density of nodes constant. Thus, we increase both the number of sources and the

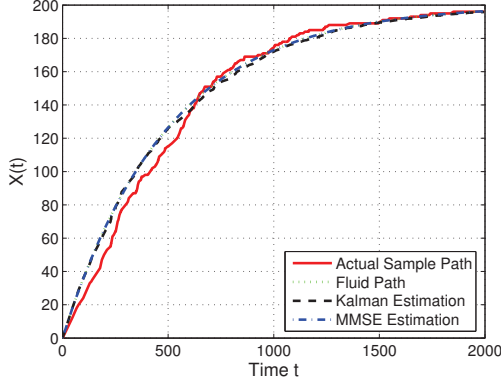


Figure 5.12: Performance of MMSE estimation and Kalman filter estimation of the process with Scenario 5.

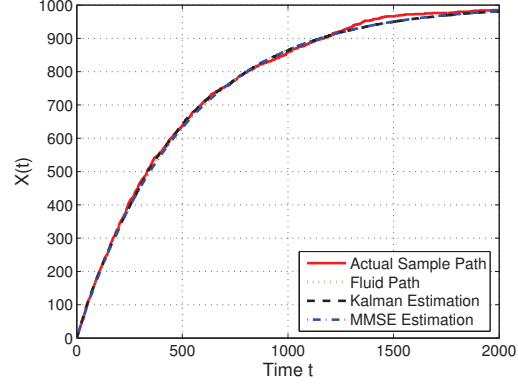


Figure 5.13: Performance of MMSE estimation and Kalman filter estimation of the process with Scenario 6.

number of users by  $K$ . However, as described in Section 5.4.1, we decrease the source-user meeting rate by  $K$ . Then, the net rate at which meetings occur in the network increases from  $p_S S_0 N_0 \beta_S = \lambda_S N_0$  to  $p_S (K S_0) (K N_0) (\beta_S / K) = \lambda_S (K N_0)$ . This scaling is equivalent to increasing only the number of users by a factor  $K$  keeping the number of sources and the source-user meeting rate constant. Thus, if the area of the network is large so that the number of users is large, then the fluid model can be a good predictor. We demonstrate this by Figures 5.12 and 5.13 which correspond to Scenarios 5 and 6, respectively. Note that Scenarios 5 and 6 are derived from Scenario 1 by scaling as above with a scaling factor  $K = 4$  and  $K = 20$ , respectively. Comparing Figures 5.2, 5.12 and 5.13, we observe that the process becomes smoother and closer to the fluid approximation with increase in the number of users.

**Level-Crossing Times:** Next, we compare the accuracy of the MMSE and the Kalman estimators in estimating the level-crossing times by computing the percentage error w.r.t. the level-crossing times of the actual process and averaging over 100 runs. Fixing the threshold levels at  $X_L = 0.15N_0, 0.25N_0, 0.50N_0, 0.75N_0$ , and  $0.90N_0$ , we obtained average percentage errors for estimates of level-crossing times by the MMSE and the Kalman estimators for Scenario 4. We summarize the results as follows:

$X_L = 0.15N_0$ ,	$e(MMSE) = 27.17\%$ ,	$e(Kalman) = 9.19\%$ .
$X_L = 0.25N_0$ ,	$e(MMSE) = 23.43\%$ ,	$e(Kalman) = 6.77\%$ .
$X_L = 0.50N_0$ ,	$e(MMSE) = 17.72\%$ ,	$e(Kalman) = 5.53\%$ .
$X_L = 0.75N_0$ ,	$e(MMSE) = 14.73\%$ ,	$e(Kalman) = 4.96\%$ .
$X_L = 0.90N_0$ ,	$e(MMSE) = 14.39\%$ ,	$e(Kalman) = 7.83\%$ .

We conducted similar experiments (results not reported here) with different parameter settings and observed similar trends. Note that the Kalman filter outperforms the MMSE estimator for all threshold levels because it takes into account all previous sample measures.

### 5.7.2 Epidemic Routing

We simulate the DTN in the case of epidemic routing for the following scenarios:

- Scenario 1':**  $S_0 = 1$ ,  $N_0 = 50$ ,  $\beta_S = 0.002$ ,  $\beta_U = 0.0002$ ,  $\lambda_H = \lambda_S$  .  
**Scenario 2':**  $S_0 = 1$ ,  $N_0 = 50$ ,  $\beta_S = 0.002$ ,  $\beta_U = 0.0002$ ,  $\lambda_H = 10\lambda_S$  .  
**Scenario 3':**  $S_0 = 1$ ,  $N_0 = 50$ ,  $\beta_S = 0.002$ ,  $\beta_U = 0.0002$ ,  $\lambda_H = 100\lambda_S$  .  
**Scenario 4':**  $S_0 = 1$ ,  $N_0 = 50$ ,  $\beta_S = 0.002$ ,  $\beta_U = 0.0002$ ,  $\lambda_H = 1000\lambda_S$  .

The corresponding plots are shown by Figures 5.14-5.21. As before, it can be observed that the tracking performance of the Kalman filter improves with increase in  $\lambda_H$  which can be achieved by increasing the number of observers or increasing the observer-user meeting rate or both.

**Level-Crossing Times:** Next, we demonstrate the accuracy of the Kalman estimators in estimating the level-crossing times by computing the percentage error w.r.t. the level-crossing times of the actual process and averaging over 100 runs. Again, fixing the threshold levels at  $X_L = 0.15N_0, 0.25N_0, 0.50N_0, 0.75N_0$ , and  $0.90N_0$ , we obtain average percentage errors for estimates of level-crossing times by the Kalman estimators for Scenario 4. We summarize the results as follows:

$$\begin{aligned} X_L = 0.15N_0 &, & e(Kalman) &= 10.22\% . \\ X_L = 0.25N_0 &, & e(Kalman) &= 9.68\% . \\ X_L = 0.50N_0 &, & e(Kalman) &= 5.24\% . \\ X_L = 0.75N_0 &, & e(Kalman) &= 10.86\% . \\ X_L = 0.90N_0 &, & e(Kalman) &= 10.10\% . \end{aligned}$$

We conducted similar experiments (results not reported here) with different parameter settings and observed similar trends. We observe from computed average percentage error w.r.t. the level-crossing times of the actual process that Kalman filter perform quite nicely w.r.t. the actual process.

## 5.8 Conclusion of the Chapter

In this chapter, we have tackled the problem of estimating file-spread in DTNs under direct delivery and epidemic routing. We have provided solid analytical basis to our estimation framework. We have also provided insightful conclusions validated with simulations. We

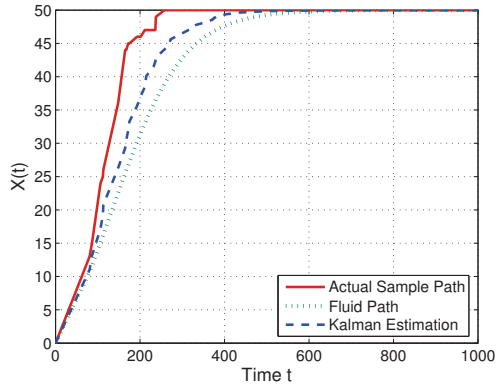


Figure 5.14: Performance of Kalman filter estimation of the process for Scenario 1'.

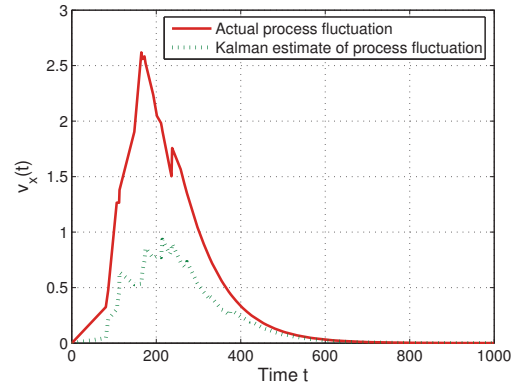


Figure 5.15: Kalman filter estimation of the process fluctuations for Scenario 1'.

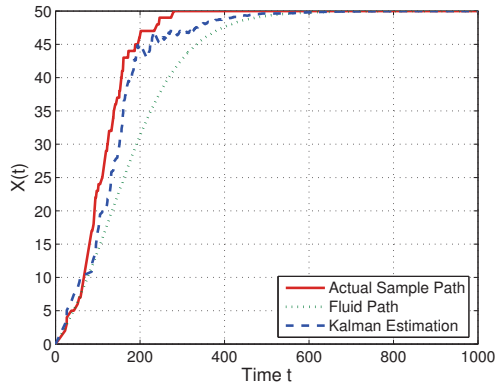


Figure 5.16: Performance of Kalman filter estimation of the process for Scenario 2'.

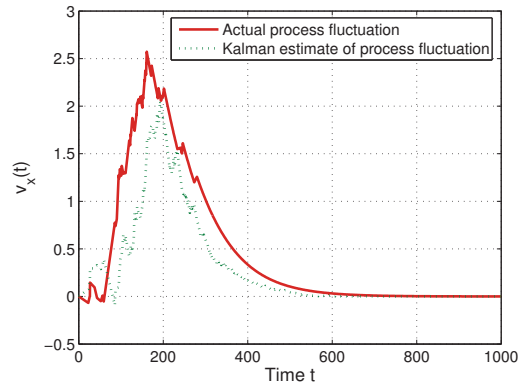


Figure 5.17: Kalman filter estimation of the process fluctuations for Scenario 2'.

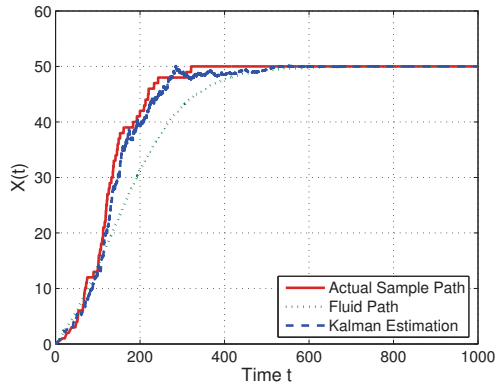


Figure 5.18: Performance of Kalman filter estimation of the process for Scenario 3'.

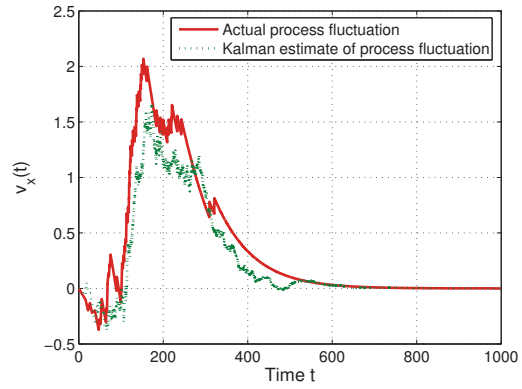


Figure 5.19: Kalman filter estimation of the process fluctuations for Scenario 3'.

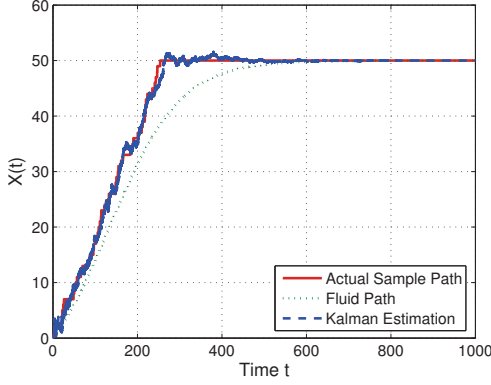


Figure 5.20: Performance of Kalman filter estimation of the process for Scenario 4'.

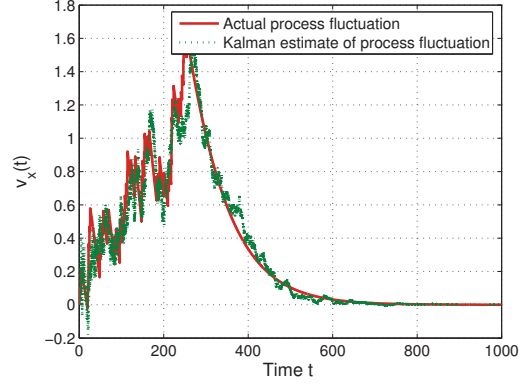


Figure 5.21: Kalman filter estimation of the process fluctuations for Scenario 4'.

have some important insights: (i) we have obtained the fluid approximation of the spreading process which, in the absence of measurements, is the only meaningful information one can derive from the probabilistic characterization of the network dynamics with relatively large number of nodes, (ii) we have derived the diffusion approximations for the spreading and observation processes, and combine them with actual observations in order to track the process using Kalman filters, (iii) we have evaluated the accuracy of tracking by computing the average error in predicting certain level-crossing times (i.e., the time at which the process and the estimation cross certain pre-specified thresholds) and (iv) in the case of direct delivery, we have derived the exact expressions for the instantaneous linear MMSE estimator and have compared its performance with that of Kalman filter in terms of accuracy in predicting level-crossing times.

## 5.9 Appendices

### 5.9.1 Fluid and Diffusion Models

In this appendix, we provide a brief informal background on fluid and diffusion limits and approximations. Please refer to [140], [141] and [142] for more details.

Intuitively speaking, the fluid approximation provides the first-order deterministic approximation to a stochastic process and represents its *average* behavior. The diffusion approximation provides the second-order approximation to a stochastic process representing its average behavior added with *random fluctuations about the average* (usually, in terms of a Brownian motion).

Consider a sequence  $\{Z^{(n)}(t), t \geq 0\}$ ,  $n = 1, 2, \dots$ , of stochastic processes. Index  $n$



represents some quantity which is *scaled up* to infinity in order to study the sequence of processes at the limit, as  $n \uparrow \infty$ . For queueing systems,  $n$  might represent “the number of servers” (as in infinite server approximations) or “a multiplying factor of one or more transition rates” (as in heavy-traffic approximations) or some other quantity w.r.t. which the scaling is performed.

Consider the Strong Law of Large Numbers (SLLN) type rescaling  $z^{(n)}(t) := Z^{(n)}(t)/n$ . Under certain conditions, as  $n \uparrow \infty$ , the sequence of rescaled processes  $\{z^{(n)}(t), t \geq 0\}$ ,  $n = 1, 2, \dots$ , converges almost surely (or sometimes in probability) to a deterministic process  $\{z(t), t \geq 0\}$  (see, for example, Theorem 4.1 of [141]). Then, the limit  $\{z(t), t \geq 0\}$  is called the *fluid limit* associated with the sequence  $\{Z^{(n)}(t), t \geq 0\}$ ,  $n = 1, 2, \dots$ , and the approximation

$$Z^{(n)}(t) \approx nz(t) , \quad \forall t \geq 0 , \quad (5.49)$$

is called the *fluid approximation* for the  $n$ -th system.

Consider now the Central Limit Theorem (CLT) type rescaling  $v_z^{(n)}(t) = \sqrt{n}(z^{(n)}(t) - z(t))$ , which amplifies the deviation of the rescaled process  $\{z^{(n)}(t), t \geq 0\}$  from the fluid limit  $\{z(t), t \geq 0\}$ . Under certain conditions, as  $n \uparrow \infty$ , the sequence of rescaled processes  $\{v_z^{(n)}(t), t \geq 0\}$ ,  $n = 1, 2, \dots$ , converges weakly to a *diffusion process* (or a continuous-time Markov process with continuous sample paths)  $\{v_z(t), t \geq 0\}$  (see, for example, Theorem 4.2 of [141]). Then,  $v_z(t)$  is called the *diffusion limit* associated with the sequence  $\{Z^{(n)}(t), t \geq 0\}$ ,  $n = 1, 2, \dots$ , and the approximation

$$Z^{(n)}(t) \stackrel{d}{\approx} nz(t) + \sqrt{n}v_z(t) , \quad \forall t \geq 0 , \quad (5.50)$$

is called the *diffusion approximation* for the  $n$ -th system, where  $\stackrel{d}{\approx}$  means “approximately distributed as”. In particular, if  $v_z(0)$  is a Gaussian random variable, then  $\{v_z(t), t \geq 0\}$  is a Gaussian process and it is completely characterized by its mean and auto-covariance functions.

### 5.9.2 Brownian Motion

In this appendix we provide the definition of the standard Brownian motion  $\{B(t), t \geq 0\}$  and summarize some useful results that we have used.

**Definition 5.9.1** (Brownian Motion). A stochastic process  $B(t, \omega)$  is called a Brownian motion if it satisfies the following conditions:

1.  $P\{\omega : B(0, \omega) = 0\} = 1$ .
2. For any  $0 \leq s < t$ , the random variable  $B(t) - B(s)$  is a Gaussian random variable with mean 0 and variance  $t - s$ .

3.  $B(t, \omega)$  has independent increments, i.e., for any  $0 \leq t_1 < t_2 < \dots < t_n$ , the random variables  $B(t_1)$ ,  $B(t_2) - B(t_1)$ ,  $\dots$ ,  $B(t_n) - B(t_{n-1})$  are independent.
4. Almost all sample paths of  $B(t, \omega)$  are continuous functions, i.e.,

$$P\{\omega : B(\cdot, \omega) \text{ is continuous}\} = 1.$$

Some important results are summarized below.

- **Mean and Covariance**

$$E[B(t)] = 0 \quad ; \quad \text{Cov}[B(s)B(t)] = \min\{s, t\}$$

- **Translation and Scaling Invariance:** For any positive constants  $t_0$  and  $c$ ,

$$\tilde{B}(t) := B(t + t_0) - B(t_0) \quad \text{and} \quad \hat{B}(t) := B(ct)/\sqrt{c}$$

are standard Brownian motions.

- **Integration w.r.t.  $B(t)$ :** If  $f(t)$  and  $g(t)$  are *square integrable* deterministic functions, then

$$\begin{aligned} E \left[ \int_a^b f(t) dB(t) \right] &= 0, \\ E \left[ \int_a^b f(t) dB(t) \int_a^b g(t) dB(t) \right] &= \int_a^b f(t)g(t)dt \\ \text{Var} \left[ \int_a^b f(t) dB(t) \right] &= \int_a^b (f(t))^2 dt \end{aligned}$$

- **Time Change**

$$B \left( \int_0^t f(u) du \right) \stackrel{d}{=} \int_0^t \sqrt{f(u)} dB(u)$$

### 5.9.3 Density-Dependent Markov Chains

First, we define the following notations used in this appendix. The set of integers (resp. real numbers) is denoted by  $\mathbb{Z}$  (resp.  $\mathbb{R}$ ). The space of  $d$ -dimensional vectors with integer (resp. real) components is denoted by  $\mathbb{Z}^d$  (resp.  $\mathbb{R}^d$ ). The absolute value of a scalar  $b$  is denoted by  $|b|$ . The norm of a vector  $z$  is denoted by  $\|z\|$ . The transpose of a vector  $z$  (resp. a matrix  $G$ ) is denoted by  $z^T$  (resp.  $G^T$ ).

Consider a one-parameter family of continuous time Markov chains  $\{Z^{(N)}(t), t \geq 0\}$ , indexed by  $N = 1, 2, \dots$ , where  $\{Z^{(N)}(t)\}$  has state space  $\mathcal{S}^{(N)} \subset \mathbb{Z}^d$  and transition rate matrix  $Q^{(N)} = [q^{(N)}(Z, Z')]$ ,  $Z, Z' \in \mathcal{S}^{(N)}$ .

**Definition 5.9.2** (Density Dependent Markov Chains [127, 137]). The family of Markov chains  $\{Z^{(N)}(t), t \geq 0\}$ ,  $N = 1, 2, \dots$ , is called density-dependent if there exist a subset  $\mathcal{R}$  of  $\mathbb{R}^d$  and continuous functions  $f_l$ ,  $l \in \mathbb{Z}^d$ , with  $f_l : \mathcal{R} \rightarrow \mathbb{R}$ , such that

$$q^{(N)}(Z, Z + l) = N f_l \left( \frac{Z}{N} \right), \quad l \neq 0.$$

In practice, instead of considering all possible  $l \in \mathbb{Z}^d$ , one only needs to consider the much smaller set

$$\mathcal{L} = \{l \in \mathbb{Z}^d : l \neq 0, q^{(N)}(Z, Z + l) \neq 0 \text{ for some } Z \in \mathcal{S}^{(N)}\},$$

whose elements correspond to (actual) transitions of positive rate. For all  $l \notin \mathcal{L}$ , one can set  $f_l$  identically equal to zero. Henceforth, we only consider transitions with positive rates and denote the total number of such transitions by  $|\mathcal{L}|$ .

Define the *drift function*  $F(\cdot)$  by

$$F(w) := \sum_l l f_l(w), \quad w \in \mathcal{R}. \quad (5.51)$$

Note that

$$F(w) = (F_1(w), \dots, F_d(w))^T$$

is a  $d$ -dimensional column vector of functions, because  $l$  is a  $d$ -dimensional column vector.

Define the *Jacobian matrix* of  $F$  by

$$J_F(w) := (\nabla F_1(w), \dots, \nabla F_d(w))^T, \quad w \in \mathcal{R}, \quad (5.52)$$

where

$$\nabla F_i(w) = \left( \frac{\partial F_i(w)}{\partial w_1}, \dots, \frac{\partial F_i(w)}{\partial w_d} \right)$$

denotes the *gradient* of the function  $F_i$  and  $w = (w_1, \dots, w_d)$ .

Defining the *density process*  $\{z^{(N)}(\cdot)\}$  by

$$z^{(N)}(t) = \frac{Z^{(N)}(t)}{N},$$

we recall the Functional Strong Law of Large Numbers (FSLLN) for density-dependent Markov chains (see [137, Chapter 11, Theorem 2.1]).

**Theorem 5.9.1** (Ethier and Kurtz [137]). Suppose that for each compact set  $K \subset \mathcal{R}$ ,

$$\sum_l \|l\| \sup_{w \in K} f_l(w) < \infty,$$

and there exists  $M_K > 0$  such that

$$\|F(w) - F(w')\| \leq M_K \|w - w'\|, \quad \forall w, w' \in K.$$

Suppose also that

$$\lim_{N \rightarrow \infty} z^{(N)}(0) = z_0,$$

and  $z(\cdot)$  satisfies

$$z(t) = z_0 + \int_0^t F(z(u)) du, \quad t \geq 0. \quad (5.53)$$

Then, for every  $t$ ,  $0 \leq t < \infty$ ,

$$\lim_{N \rightarrow \infty} \sup_{0 \leq s \leq t} \|z^{(N)}(s) - z(s)\| = 0,$$

almost surely.

*Remark 5.9.1.* Theorem 5.9.1 says that, when the drift function  $F(\cdot)$  is uniformly bounded and Lipschitz continuous over compact subsets of  $\mathcal{R}$ , then, as  $N \rightarrow \infty$ , the density process  $\{z^{(N)}\}$  converges uniformly over compact subsets (u.o.c.) to a deterministic function  $z(\cdot)$ , almost surely (a.s.).

Defining the *deviation process*  $\{v_z^{(N)}(\cdot)\}$  by

$$v_z^{(N)}(t) = \sqrt{N}(z^{(N)}(t) - z(t)),$$

where  $z^{(N)}(t)$  is the density process and  $z(\cdot)$  denotes the deterministic limit given by (5.53), we recall the Functional Central Limit Theorem (FCLT) for density-dependent Markov chains (see [137, Chapter 11, Theorem 2.3]).

**Theorem 5.9.2** (Ethier and Kurtz [137]). *Suppose that for each compact set  $K \subset \mathcal{R}$ ,*

$$\sum_l \|l\|^2 \sup_{w \in K} f_l(w) < \infty,$$

*and that the  $f_l$ ,  $l \in \mathcal{L}$ , and  $J_F$  are continuous. Suppose also that*

$$\lim_{N \rightarrow \infty} v_z^{(N)}(0) = v_{z_0},$$

*where  $v_{z_0}$  is a constant. Then, as  $N \rightarrow \infty$ ,  $\{v_z^{(N)}(\cdot)\}$  converges in distribution to  $v_z(\cdot)$ , where  $v_z(\cdot)$  is the solution to the stochastic integral equation*

$$\begin{aligned} v_z(t) &= v_{z_0} + \sum_l l B_l \left( \int_0^t f_l(z(u)) du \right) \\ &\quad + \int_0^t J_F(z(u)) v_z(u) du, \end{aligned} \quad (5.54)$$

*where  $B_l(\cdot)$ ,  $l \in \mathcal{L}$ , are independent standard Brownian motions (each corresponding to a transition with positive rate) and  $z(t)$  is given by (5.53).*

### 5.9.4 Continuous Mapping Approach

First, we define the following notations used in this appendix. The identity mapping is denoted by  $\iota$ . The empty set is denoted by  $\emptyset$ . The composition of two functions  $f$  and  $g$  is denoted by  $f \circ g$ . The set of discontinuities of a function  $g$  is denoted by  $\text{Disc}(g)$ .

Let  $(E, m)$  denote a metric space with metric  $m$ . The space of functions  $\phi : [0, \infty) \rightarrow E$  that are continuous are denoted by  $\mathbb{C}_E[0, \infty)$ . The space of functions  $\phi : [0, \infty) \rightarrow E$  that are right-continuous and have left limits are denoted by  $\mathbb{D}_E[0, \infty)$ . We are particularly interested in the case  $E = \mathbb{R}^d$ , and use the simple notation  $\mathbb{C}^d$  (resp.  $\mathbb{D}^d$ ) to denote  $\mathbb{C}_{\mathbb{R}^d}[0, \infty)$  (resp.  $\mathbb{D}_{\mathbb{R}^d}[0, \infty)$ ). The subset of functions in  $\mathbb{C}^1$  that are nondecreasing (resp. strictly increasing) is denoted by  $\mathbb{C}_{\uparrow}^1$  (resp.  $\mathbb{C}_{\uparrow\uparrow}^1$ ). The subset of functions in  $\mathbb{D}^1$  that are nondecreasing (resp. strictly increasing) is denoted by  $\mathbb{D}_{\uparrow}^1$  (resp.  $\mathbb{D}_{\uparrow\uparrow}^1$ ). The cartesian product of two spaces  $S$  and  $S'$  is denoted by  $S \times S'$ . The space of  $d$ -dimensional elements, where each component of an element takes values in  $S$ , is denoted by  $S^d$ .

Convergence in distribution is denoted by  $\Rightarrow$ .

The continuous mapping approach exploits previously established stochastic-process limits and the Continuous Mapping Theorem to obtain new stochastic-process limits [142]. In this appendix, we recall a particular version of the Continuous Mapping Theorem (CMT) which specifies the conditions under which convergence is preserved for composition plus addition (see [142, Theorem 13.3.1]).

**Theorem 5.9.3** (Convergence preservation for composition plus addition [142]). *Let  $\psi_1, \psi_3$  and  $\psi_1^{(N)}$ ,  $N \geq 1$ , be random elements of  $\mathbb{D}^d$ ; let  $\psi_2, \psi_2^{(N)}$  and  $\psi_3^{(N)}$ ,  $N \geq 1$ , be random elements of  $\mathbb{D}_{\uparrow}^1$ ; and let  $c_N \in \mathbb{R}^d$  for  $N \geq 1$ . If*

$$\left( \psi_1^{(N)} - c_N \iota, \psi_2^{(N)}, c_N(\psi_2^{(N)} - \psi_3^{(N)}) \right) \Rightarrow (\psi_1, \psi_2, \psi_3)$$

*in  $\mathbb{D}^d \times \mathbb{D}_{\uparrow}^1 \times \mathbb{D}^d$ , and  $\psi_2 \in \mathbb{C}_{\uparrow\uparrow}^1$  and*

$$P(\text{Disc}(\psi_1 \circ \psi_2) \cap \text{Disc}(\psi_3) = \emptyset) = 1,$$

*then*

$$\psi_1^{(N)} \circ \psi_2^{(N)} - c_N \psi_3^{(N)} \Rightarrow \psi_1 \circ \psi_2 + \psi_3$$

*in  $\mathbb{D}^d$ .*

When applying Theorem 5.9.3, we shall establish the requirement of joint convergence in distribution by invoking independence (see [142, Theorem 11.4.4]) and convergence in distribution to a deterministic limit of one of the component sequence of random elements (see [142, Theorem 11.4.5]).

**Theorem 5.9.4** (Joint convergence for independent random elements [142]). *Let  $\psi_1^{(N)}$  and  $\psi_2^{(N)}$  be **independent** random elements of separable metric spaces  $(S', m')$  and  $(S'', m'')$ , respectively, for each  $N \geq 1$ . Then there is joint convergence in distribution  $(\psi_1^{(N)}, \psi_2^{(N)}) \Rightarrow (\psi_1, \psi_2)$  in  $S' \times S''$  if and only if  $\psi_1^{(N)} \Rightarrow \psi_1$  in  $S'$  and  $\psi_2^{(N)} \Rightarrow \psi_2$  in  $S''$ .*

**Theorem 5.9.5** (Joint convergence when one limit is deterministic [142]). *Suppose that  $\psi_1^{(N)} \Rightarrow \psi_1$  in a separable metric space  $S'$  and  $\psi_2^{(N)} \Rightarrow \psi_2$  in a separable metric space  $S''$ , where  $\psi_2$  is deterministic. Then*

$$(\psi_1^{(N)}, \psi_2^{(N)}) \Rightarrow (\psi_1, \psi_2).$$



# Conclusions and Perspectives

## Contents

---

<b>6.1</b>	<b>Conclusions . . . . .</b>	<b>135</b>
<b>6.2</b>	<b>Perspectives . . . . .</b>	<b>136</b>

---

## 6.1 Conclusions

In this thesis, we have first proposed a new reliable transport scheme for DTNs. Our proposed scheme is based on the use of ACKs and coding. We have modeled the evolution of the network under our scheme using a fluid-approximation approach. We have optimized, using differential evolution approach, the number of outstanding RLCs to be sent back to back before the expiration of a so called cycle time-out as well as the value of the time-out itself, so as to minimize the file transfer time. We have analytically obtained mean file transfer times and showed, by comparing numerical analysis versus simulations, the validity of our models. We have accounted for the buffer expiry time-out and quantified its impact on the optimal values of our protocol parameters. We have also demonstrated the adaptability of our optimal procedure to variations in expiry time-out. We have also proposed two approximations to overcome the computational complexity of the proposed model and observed that these approximations work very close to our initial proposal.

Secondly, we have proposed and studied via extensive simulations a new set of enhancement features to improve reliable transport in DTNs. Our proposed enhancements cover both unicast and multicast flows. We have enabled intersession random linear network coding at relays. We have observed that the use of coding not only helps in fast delivery



of packet(s) at the destination(s) but also increases the probability of successful data delivery. We have also observed significant improvement in variance due to network coding. We have relied on the use of a smart ACK mechanism, which we call G-SACK, to improve reliability. The G-SACK potentially contains global information on receipt of packets at all destinations. We have quantified the gains achieved due to various enhancement features to improve reliable transport individually as well as the overall gain achieved by all of them together.

Finally, we have tackled the problem of estimating message-spread in DTNs under direct delivery and epidemic routing. We have provided analytical basis to our estimation framework. We have also provided insights validated with simulations. Some of the important insights are: (i) we have obtained the fluid approximation of the spreading process which, in the absence of measurements, is the only meaningful information one can derive from the probabilistic characterization of the network dynamics with relatively large number of nodes, (ii) we have derived the diffusion approximations for the spreading and observation processes, and combine them with actual observations in order to track the process using Kalman filters, (iii) we have evaluated the accuracy of tracking by computing the average error in predicting the time at which the process and the estimation cross certain pre-specified thresholds and (iv) we have derived, in the case of direct delivery, the exact expressions for the instantaneous linear MMSE estimator and we have compared its performance in terms of accuracy with that of Kalman filter with respect to prediction of level-crossing times.

## 6.2 Perspectives

Since the mobile nodes in DTNs operate on battery power, minimization of energy consumption by limiting the number of copies of RLCs and ACKs is one of the primary issues in designing protocols for DTNs. Our transport scheme has been designed to minimize the mean round-trip delay in case of successful transfer without accounting for energy constraints. Although, our reliable transport proposal does not focus on energy issues (related to RLC and ACK replication), it is worth mentioning that the parameters  $\tau_{i,S}$  and  $\tau_{i,W}$  of our proposed protocol can be tuned to trade between delay and energy. Therefore, it would be of interest to investigate the trade-off between reliability and energy consumption due to the replication of packets and ACKs in the relays with our proposed protocol.

Next, of immense practical interest, are the scenarios where: (1) multiple packets per flow are allowed in the relay buffers, and (2) the number of information packets per flow is of the same order of magnitude as the number of nodes in the network. Our reliable transport scheme may be extended to cover such scenarios by combining with suitable

buffer management policies such as [63, 64].

It would also be interesting to design a reliable transport proposal considering the coding at intermediate nodes. We have studied, through simulations, the gains due to random linear network coding of packets of different sources at relays, but not in an optimal way.

It would be quite interesting to evaluate enhancement schemes to improve reliable transport with multiple packet buffers and multiple packet transfers between source-destination pairs.

The problem of adaptively controlling the spreading process under direct delivery would also be quite important to investigate. Using our estimation scheme, it would be interesting to study controls of spreading process for broadcast/multicast applications. For the unicast case, we have studied the optimal forwarding problem with the help of explicit feedback provided by the unicast destination.



# Bibliography

- [1] P. Pereira, A. Casaca, J. Rodrigues, V. Soares, J. Triay, and C. Cervelló-Pastor, “From delay-tolerant networks to vehicular delay-tolerant networks,” *Communications Surveys & Tutorials, IEEE*, no. 99, pp. 1–17, 2011.
- [2] C. Caini, H. Cruickshank, S. Farrell, and M. Marchese, “Delay-and disruption-tolerant networking (DTN): An alternative solution for future satellite networking applications,” *Proceedings of the IEEE*, no. 99, pp. 1–18, 2011.
- [3] P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, and C. Diot, “Pocket switched networks and human mobility in conference environments,” in *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*. ACM, 2005, pp. 244–251.
- [4] T. Jonson, J. Pezeshki, V. Chao, K. Smith, and J. Fazio, “Application of delay tolerant networking (DTN) in airborne networks,” in *Military Communications Conference, 2008. MILCOM 2008. IEEE*. IEEE, 2008, pp. 1–7.
- [5] N. Kayastha, D. Niyato, P. Wang, and E. Hossain, “Applications, architectures, and protocol design issues for mobile social networks: A survey,” *Proceedings of the IEEE*, vol. 99, no. 12, pp. 2130–2158, 2011.
- [6] T. Small and Z. Haas, “The shared wireless infostation model: a new ad hoc networking paradigm (or where there is a whale, there is a way),” in *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*. ACM, 2003, pp. 233–244.
- [7] A. Pentland, R. Fletcher, and A. Hasson, “Daknet: rethinking connectivity in developing nations,” *Computer*, vol. 37, no. 1, pp. 78–83, 2004.

- [8] S. Burleigh, A. Hooke, L. Torgerson, K. Fall, V. Cerf, B. Durst, K. Scott, and H. Weiss, "Delay-tolerant networking: an approach to interplanetary internet," *Communications Magazine, IEEE*, vol. 41, no. 6, pp. 128 – 136, june 2003.
- [9] "Internet Research Task Force Delay-Tolerant Networking Research Group." [Online]. Available: <http://www.dtnrg.org>
- [10] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc On-demand Distance Vector (AODV) routing (rfc 3561)," *IETF MANET Working Group (August. 2003)*, 2003.
- [11] D. Johnson and D. Maltz, "Dynamic source routing in ad hoc wireless networks," *Mobile computing*, pp. 153–181, 1996.
- [12] S. Biswas and R. Morris, "ExOR: Opportunistic Multi-Hop Routing for Wireless Networks," in *Sigcomm*, 2005.
- [13] T. Spyropoulos, K. Psounis, and C. Raghavendra, "Efficient routing in intermittently connected mobile networks: the multiple-copy case," *Networking, IEEE/ACM Transactions on*, vol. 16, no. 1, pp. 77–90, 2008.
- [14] A. Lindgren, A. Doria, and O. Schelen, "Probabilistic Routing in Intermittently Connected Networks," in *ACM Mobicom*, 2003.
- [15] E. Bulut, Z. Wang, and B. Szymanski, "Cost-effective multiperiod spraying for routing in delay-tolerant networks," *IEEE/ACM Transactions on Networking (TON)*, vol. 18, no. 5, pp. 1530–1543, 2010.
- [16] M. Grossglauser and D. Tse, "Mobility increases the capacity of ad hoc wireless networks," *Networking, IEEE/ACM Transactions on*, vol. 10, no. 4, pp. 477–486, 2002.
- [17] A. Vahdat and D. Becker, "Epidemic Routing for Partially-Connected Ad Hoc Networks," in *Duke University, Tech. Rep. CS-200006*, 2000.
- [18] J. Postel, "User Datagram Protocol," in *RFC-768*, 1980.
- [19] —, "Transmission Control Protocol," in *RFC-793*, 1981.
- [20] D. Macedo, A. dos Santos, and G. Pujolle, "From tcp/ip to convergent networks: challenges and taxonomy," *Communications Surveys Tutorials, IEEE*, vol. 10, no. 4, pp. 40 –55, quarter 2008.
- [21] H. Balakrishnan *et al.*, *Challenges to reliable data transport over heterogeneous wireless networks*. University of California, Berkeley, 1998.

- [22] H. Balakrishnan, S. Seshan, E. Amir, and R. Katz, "Improving TCP/IP performance over wireless networks," in *Proceedings of the 1st annual international conference on Mobile computing and networking*. ACM, 1995, pp. 2–11.
- [23] T. Mahmoodi, V. Friderikos, O. Holland, and A. Hamid Aghvami, "Cross-layer design to improve wireless tcp performance with link-layer adaptation," in *Vehicular Technology Conference, 2007. VTC-2007 Fall. 2007 IEEE 66th*. IEEE, 2007, pp. 1504–1508.
- [24] G. Holland and N. Vaidya, "Analysis of TCP Performance Over Mobile Ad Hoc Networks," *ACM Wireless Networks*, vol. 8, no. 2, pp. 275–288, March 2002.
- [25] M. Ramadas, S. Burleigh, and S. Farrell, "Licklider Transmission Protocol - Motivation," in *RFC-5325*, September 2008.
- [26] Q. Zhang, Z. Jin, Z. Zhang, and Y. Shu, "Network coding for applications in the delay tolerant network (dtn)," in *Mobile Ad-hoc and Sensor Networks, 2009. MSN '09. 5th International Conference on*, dec. 2009, pp. 376 –380.
- [27] K. Fall, "A delay-tolerant network architecture for challenged internets," in *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, ser. SIGCOMM '03. New York, NY, USA: ACM, 2003, pp. 27–34. [Online]. Available: <http://doi.acm.org/10.1145/863955.863960>
- [28] E. Altman, F. D. Pellegrini, and L. Sassatelli, "Dynamic Control of Coding in Delay Tolerant Networks," in *IEEE Infocom*, March 2010.
- [29] Y. Lin, B. Liang, and B. Li, "Performance modeling of network coding in epidemic routing," in *Proceedings of the 1st international MobiSys workshop on Mobile opportunistic networking*. ACM, 2007, pp. 67–74.
- [30] X. Zhang, G. Neglia, J. Kurose, and D. Towsley, "On the benefits of random linear coding for unicast applications in disruption tolerant networks," in *Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks, 2006 4th International Symposium on*. IEEE, 2006, pp. 1–7.
- [31] E. Altman, "An epidemic game between contents in a wireless network," 2012.
- [32] M. Ramadas, S. Burleigh, and S. Farrell, "Licklider Transmission Protocol Specification," in *RFC-5326*, September 2008.
- [33] S. Farrell, M. Ramadas, and S. Burleigh, "Licklider transmission protocol-security extensions," 2008.

- [34] S. Farrell and V. Cahill, "Evaluating LTP-T: A DTN-Friendly Transport Protocol," in *Satellite and Space Communications, 2007. IWSSC '07. International Workshop on*, sept. 2007, pp. 178–181.
- [35] I. Psaras, G. Papastergiou, V. Tsaoussidis, and N. Peccia, "DS-TP: Deep-Space Transport Protocol," in *Aerospace Conference, 2008 IEEE*. IEEE, 2008, pp. 1–13.
- [36] G. Papastergiou, I. Psaras, and V. Tsaoussidis, "Deep-Space Transport Protocol: A Novel Transport Scheme for Space DTNs," *Computer Communications, Special Issue of Computer Communications on Delay and Disruption Tolerant Networking*, vol. 32, no. 16, pp. 1757–1767, October 2009.
- [37] K. Fall and S. Farrell, "DTN: An Architectural Retrospective," *Selected Areas in Communications, IEEE Journal on*, vol. 26, no. 5, pp. 828–836, 2008.
- [38] K. A. Harras and K. C. Almeroth, "Transport Layer Issues in Delay Tolerant Mobile Networks," in *IFIP NETWORKING*, 2006.
- [39] O. B. Akan, J. Fang, and I. F. Akyildiz, "TP-Planet: A Reliable Transport Protocol for InterPlanetary Internet," *IEEE Journal on Selected Areas in Communications(JSAC)*, vol. 22, no. 2, pp. 348–361, 2004.
- [40] K. Fall and S. McCanne, "You don't know jack about network performance," *Queue*, vol. 3, no. 4, pp. 54–59, 2005.
- [41] I. Akyildiz, Ö. Akan, C. Chen, J. Fang, and W. Su, "Interplanetary internet: state-of-the-art and research challenges," *Computer Networks*, vol. 43, no. 2, pp. 75–112, 2003.
- [42] K. Scott and S. Burleigh, "Bundle Protocol Specification," in *RFC-5050*, November 2007.
- [43] C. C. for Space Data Systems, "Space Communications Protocol Standards (SCPS) - Transport Protocol (SCPS-TP)," in *CCSDS 714.0-B-2, Blue Book*, October 2006.
- [44] CCSDS, "CCSDS File Delivery Protocol (CFDP)," in *CCSDS 727.0-B-4, Blue Book*, January 2007.
- [45] J. Fang and I. F. Akyildiz, "RCP-Planet: A Rate Control Protocol for Interplanetary Internet," *International Journal of Satellite Communications and Networking*, vol. 25, no. 2, pp. 167–194, 2007. [Online]. Available: <http://dx.doi.org/10.1002/sat.873>

- [46] L. Wood, J. McKim, W. Eddy, W. Ivancic, and C. Jackson, "Saratoga: A Scalable File Transfer Protocol," November 13, 2009.
- [47] C. Samaras and V. Tsaoussidis, "DTTP: A Delay-Tolerant Transport Protocol for Space Internetworks," in *ERCIM*, 2008.
- [48] G. Papastergiou, C. Samaras, and V. Tsaoussidis, "Where does transport layer fit into space dtn architecture?" in *Advanced satellite multimedia systems conference (asma) and the 11th signal processing for space communications workshop (spsc), 2010 5th*. IEEE, 2010, pp. 81–88.
- [49] L. Wood, W. Eddy, and P. Holliday, "A bundle of problems," in *Aerospace conference, 2009 IEEE*. IEEE, 2009, pp. 1–17.
- [50] K. Fall, W. Hong, and S. Madden, "Custody transfer for reliable delivery in delay tolerant networks," *IRB-TR-03-030*, July, 2003.
- [51] L. Brakmo and L. Peterson, "TCP Vegas: End to end congestion avoidance on a global internet," *Selected Areas in Communications, IEEE Journal on*, vol. 13, no. 8, pp. 1465–1480, 1995.
- [52] F. Muhammad, L. Franck, and S. Farrell, "Transmission protocols for challenging networks: Ltp and ltp-t," in *Satellite and Space Communications, 2007. IWSSC'07. International Workshop on*. IEEE, 2007, pp. 145–149.
- [53] S. Farrell and V. Cahill, "LTP-T: A Generic Delay Tolerant Transport Protocol," Technical Report TCD-CS-2005-69, Computer Science, Trinity College Dublin, Tech. Rep., 2005.
- [54] "SSTL: Surrey Satellite Technology Ltd." [Online]. Available: <http://www.sstl.co.uk>
- [55] M. Seligman, K. Fall, and P. Mundur, "Storage routing for dtn congestion control," *Wireless Communications and Mobile Computing*, vol. 7, no. 10, pp. 1183–1196, 2007.
- [56] J. Ott and D. Kutscher, "A disconnection-tolerant transport for drive-thru internet environments," in *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, vol. 3. IEEE, 2005, pp. 1849–1862.
- [57] M. Seligman, K. Fall, and P. Mundur, "Alternative custodians for congestion control in delay tolerant networks," in *Proceedings of the 2006 SIGCOMM workshop on Challenged networks*. ACM, 2006, pp. 229–236.



- [58] S. Schütz, L. Eggert, S. Schmid, and M. Brunner, "Protocol enhancements for intermittently connected hosts," *ACM SIGCOMM Computer Communication Review*, vol. 35, no. 3, pp. 5–18, 2005.
- [59] J. Ott and D. Kutscher, "Drive-thru internet: Ieee 802.11 b for," in *INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 1. IEEE, 2004.
- [60] P. Nikander and R. Moskowitz, "Host identity protocol (hip) architecture," 2006.
- [61] L. Eggert and F. Gont, "Tcp user timeout option," 2009.
- [62] L. Eggert, S. Schütz, and S. Schmid, "Tcp extensions for immediate retransmissions," *draft-eggert-tcpm-tcp-retransmit-now-02 (work in progress)*, 2005.
- [63] A. Krifa, C. Barakat, and T. Spyropoulos, "Mobitrade: trading content in disruption tolerant networks," in *Proceedings of the 6th ACM workshop on Challenged networks*. ACM, 2011, pp. 31–36.
- [64] A. Krifa, C. Barakat, T. Spyropoulos *et al.*, "Message drop and scheduling in DTNs: Theory and practice," 2010.
- [65] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot, "Optimized link state routing protocol for ad hoc networks," in *Multi Topic Conference, 2001. IEEE INMIC 2001. Technology for the 21st Century. Proceedings. IEEE International*. IEEE, 2001, pp. 62–68.
- [66] Y. Ko and N. Vaidya, "Location-Aided Routing (LAR) in mobile ad hoc networks," *Wireless Networks*, vol. 6, no. 4, pp. 307–321, 2000.
- [67] S. Lee, W. Su, and M. Gerla, "Wireless ad hoc multicast routing with mobility prediction," *Mobile Networks and Applications*, vol. 6, no. 4, pp. 351–360, 2001.
- [68] Y. Ge, T. Kunz, and L. Lamont, "Quality of service routing in ad-hoc networks using olsr," in *System Sciences, 2003. Proceedings of the 36th Annual Hawaii International Conference on*. IEEE, 2003, pp. 9–pp.
- [69] H. Dubois-Ferriere, M. Grossglauser, and M. Vetterli, "Age matters: efficient route discovery in mobile ad hoc networks using encounter ages," in *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*. ACM, 2003, pp. 257–266.

- [70] E. Jones and P. Ward, "Routing strategies for delay-tolerant networks," *Submitted to ACM Computer Communication Review (CCR)*, 2006.
- [71] J. Shen, S. Moh, and I. Chung, "Routing protocols in delay tolerant networks: A comparative survey," in *The 23rd International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC 2008)*, 2008, pp. 6–9.
- [72] M. Liu, Y. Yang, and Z. Qin, "A survey of routing protocols and simulations in delay-tolerant networks," *Wireless Algorithms, Systems, and Applications*, pp. 243–253, 2011.
- [73] L. Pelusi, A. Passarella, and M. Conti, "Opportunistic Networking: Data Forwarding in Disconnected Mobile Ad Hoc Networks," *IEEE Communications Magazine*, vol. 44, no. 11, pp. 134–141, November 2006.
- [74] Z. Zhang, "Routing in intermittently connected mobile ad hoc networks and delay tolerant networks: overview and challenges," *Communications Surveys Tutorials, IEEE*, vol. 8, no. 1, pp. 24–37, quarter 2006.
- [75] M. Khabbaz, C. Assi, and W. Fawaz, "Disruption-tolerant networking: A comprehensive survey on recent developments and persisting challenges," *Communications Surveys Tutorials, IEEE*, vol. PP, no. 99, pp. 1–34, 2011.
- [76] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. S. Peh, and D. Rubenstein, "Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with zebranet," *SIGOPS Oper. Syst. Rev.*, vol. 36, no. 5, pp. 96–107, Oct. 2002. [Online]. Available: <http://doi.acm.org/10.1145/635508.605408>
- [77] A. Doria, "Providing connectivity to the saami nomadic community," in *In Proc. 2nd Int. Conf. on Open Collaborative Design for Sustainable Innovation*, 2002.
- [78] D. Goodman, J. Borras, N. Mandayam, and R. Yates, "Infostations: A new system model for data and messaging services," in *Vehicular Technology Conference, 1997 IEEE 47th*, vol. 2. IEEE, 1997, pp. 969–973.
- [79] W. Zhao, M. Ammar, and E. Zegura, "A message ferrying approach for data delivery in sparse mobile ad hoc networks," in *Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing*. ACM, 2004, pp. 187–198.
- [80] S. Jain, R. Shah, W. Brunette, G. Borriello, and S. Roy, "Exploiting mobility for energy efficient data collection in wireless sensor networks," *Mobile Networks and Applications*, vol. 11, no. 3, pp. 327–339, 2006.

- [81] A. Vahdat, D. Becker *et al.*, “Epidemic Routing for Partially Connected Ad-Hoc Networks,” Technical Report CS-200006, Duke University, Tech. Rep., 2000.
- [82] B. Burns, O. Brock, and B. Levine, “MV routing and capacity building in disruption tolerant networks,” in *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, vol. 1, march 2005, pp. 398 – 408 vol. 1.
- [83] A. Lindgren, A. Doria, and O. Schelén, “Probabilistic routing in intermittently connected networks,” *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 7, no. 3, pp. 19–20, 2003.
- [84] J. Widmer and J.-Y. Le Boudec, “Network coding for efficient communication in extreme networks,” in *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, ser. WDTN ’05. New York, NY, USA: ACM, 2005, pp. 284–291. [Online]. Available: <http://doi.acm.org/10.1145/1080139.1080147>
- [85] M. Musolesi, S. Hailes, and C. Mascolo, “Adaptive routing for intermittently connected mobile ad hoc networks,” in *World of Wireless Mobile and Multimedia Networks, 2005. WoWMoM 2005. Sixth IEEE International Symposium on a*, june 2005, pp. 183 – 189.
- [86] J. Leguay, T. Friedman, and V. Conan, “Evaluating mobility pattern space routing for dtns,” in *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings.* IEEE, 2006, pp. 1–10.
- [87] E. Jones, L. Li, J. Schmidtke, and P. Ward, “Practical routing in delay-tolerant networks,” *Mobile Computing, IEEE Transactions on*, vol. 6, no. 8, pp. 943 –959, aug. 2007.
- [88] X. Zhang, G. Neglia, J. Kurose, and D. Towsley, “Performance Modeling of Epidemic Routing,” *Computer Networks*, vol. 51, pp. 2867–2891, 2007.
- [89] A. Jindal and K. Psounis, “Performance analysis of epidemic routing under contention,” in *Proceedings of the 2006 international conference on Wireless communications and mobile computing.* ACM, 2006, pp. 539–544.
- [90] Y. Lin, B. Li, and B. Liang, “Stochastic analysis of network coding in epidemic routing,” 2007.
- [91] T. Spyropoulos, K. Psounis, and C. Raghavendra, “Spray and wait: an efficient routing scheme for intermittently connected mobile networks,” in *Proceedings of the*

- 2005 ACM SIGCOMM workshop on Delay-tolerant networking. ACM, 2005, pp. 252–259.
- [92] S. Jain, K. Fall, and R. Patra, “Routing in a delay tolerant network,” *SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 4, pp. 145–158, Aug. 2004. [Online]. Available: <http://doi.acm.org/10.1145/1030194.1015484>
- [93] T. Small and Z. J. Haas, “Resource and performance tradeoffs in delay-tolerant wireless networks,” in *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, ser. WDTN ’05. New York, NY, USA: ACM, 2005, pp. 260–267. [Online]. Available: <http://doi.acm.org/10.1145/1080139.1080144>
- [94] T. Spyropoulos, K. Psounis, and C. Raghavendra, “Efficient routing in intermittently connected mobile networks: The single-copy case,” *IEEE/ACM Transactions on Networking (TON)*, vol. 16, no. 1, pp. 63–76, 2008.
- [95] R. Ahlswede, N. Cai, S.-Y. Li, and R. Yeung, “Network information flow,” *Information Theory, IEEE Transactions on*, vol. 46, no. 4, pp. 1204–1216, jul 2000.
- [96] S.-Y. Li, R. Yeung, and N. Cai, “Linear network coding,” *Information Theory, IEEE Transactions on*, vol. 49, no. 2, pp. 371–381, feb. 2003.
- [97] R. Koetter and M. Medard, “An algebraic approach to network coding,” *Networking, IEEE/ACM Transactions on*, vol. 11, no. 5, pp. 782–795, oct. 2003.
- [98] T. Ho, M. Medard, R. Koetter, D. Karger, M. Effros, J. Shi, and B. Leong, “A random linear network coding approach to multicast,” *Information Theory, IEEE Transactions on*, vol. 52, no. 10, pp. 4413–4430, oct. 2006.
- [99] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, “Xors in the air: practical wireless network coding,” in *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 4. ACM, 2006, pp. 243–254.
- [100] A. Fujimura, S. Oh, and M. Gerla, “Network coding vs. erasure coding: Reliable multicast in ad hoc networks,” in *Military Communications Conference, 2008. MILCOM 2008. IEEE*. Ieee, 2008, pp. 1–7.
- [101] S. Oh, M. Gerla, and A. Tiwari, “Robust manet routing using adaptive path redundancy and coding,” in *Communication Systems and Networks and Workshops, 2009. COMSNETS 2009. First International*. IEEE, 2009, pp. 1–10.
- [102] M. Luby, “LT Codes,” in *IEEE FOCS*, 2002, pp. 271–282.

- [103] M. A. Shokrollahi, "Raptor Codes," in *IEEE International Symposium on Information Theory*, July 2003.
- [104] D. S. Lun, M. Médard, and M. Effros, "On Coding for Reliable Communication Over Packet Networks," in *Proc. 42nd Annual Allerton Conference on Communication, Control, and Computing*, September 2004, pp. 20–29.
- [105] Y. Wang, S. Jain, M. Martonosi, and K. Fall, "Erasure-coding based routing for opportunistic networks," in *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*. ACM, 2005, pp. 229–236.
- [106] L. Chen, C. Yu, T. Sun, Y. Chen, and H. Chu, "A hybrid routing approach for opportunistic networks," in *Proceedings of the 2006 SIGCOMM workshop on Challenged networks*. ACM, 2006, pp. 213–220.
- [107] B. Vellambi, R. Subramanian, F. Fekri, and M. Ammar, "Reliable and efficient message delivery in delay tolerant networks using rateless codes," in *Proceedings of the 1st international MobiSys workshop on Mobile opportunistic networking*. ACM, 2007, pp. 91–98.
- [108] E. Bulut, Z. Wang, and B. Szymanski, "Cost efficient erasure coding based routing in delay tolerant networks," in *Communications (ICC), 2010 IEEE International Conference on*. IEEE, 2010, pp. 1–5.
- [109] Y. Wang and H. Wu, "Delay/Fault-Tolerant Mobile Sensor Network (DFT-MSN): A new paradigm for pervasive information gathering," *Mobile Computing, IEEE Transactions on*, vol. 6, no. 9, pp. 1021–1034, 2007.
- [110] Y. Lin, B. Li, and B. Liang, "Efficient network coded data transmissions in disruption tolerant networks," 2008, pp. 1508–1516.
- [111] S. Jain, M. Demmer, R. Patra, and K. Fall, "Using redundancy to cope with failures in a delay tolerant network," in *ACM SIGCOMM Computer Communication Review*, vol. 35, no. 4. ACM, 2005, pp. 109–120.
- [112] L. Chen, C. Yu, C. Tseng, H. Chu, and C. Chou, "A content-centric framework for effective data dissemination in opportunistic networks," *Selected Areas in Communications, IEEE Journal on*, vol. 26, no. 5, pp. 761–772, 2008.
- [113] Y. Dai, P. Yang, G. Chen, and J. Wu, "Cfp: Integration of fountain codes and optimal probabilistic forwarding in dtns," in *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*. IEEE, 2010, pp. 1–5.

- [114] K.-C. Chung, Y.-C. Li, and W. Liao, "Exploiting network coding for data forwarding in delay tolerant networks," in *Vehicular Technology Conference (VTC 2010-Spring), 2010 IEEE 71st*, may 2010, pp. 1–5.
- [115] J. Widmer and J. Le Boudec, "Network coding for efficient communication in extreme networks," in *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*. ACM, 2005, pp. 284–291.
- [116] Y. Cao and Z. Sun, "Routing in delay/disruption tolerant networks: A taxonomy, survey and challenges."
- [117] Z. Haas, J. Halpern, and L. Li, "Gossip-based ad hoc routing," in *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 3. IEEE, 2002, pp. 1707–1716.
- [118] S. Ahmed and S. Kanhere, "Hubcode: message forwarding using hub-based network coding in delay tolerant networks," in *Proceedings of the 12th ACM international conference on Modeling, analysis and simulation of wireless and mobile systems*. ACM, 2009, pp. 288–296.
- [119] R. Storn and K. Price, "Differential evolution-a simple and efficient adaptive scheme for global optimization over continuous spaces," *International computer science institute-publications-TR*, 1995.
- [120] ———, "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces," *Journal of global optimization*, vol. 11, no. 4, pp. 341–359, 1997.
- [121] J. Lampinen and I. Zelinka, "Mixed integer-discrete-continuous optimization by differential evolution," in *Proceedings of the 5th International Conference on Soft Computing*. Citeseer, 1999, pp. 71–76.
- [122] D. Lucani, M. Stojanovic, and M. Médard, "Random Linear Network Coding for Time Division Duplexing: When to Stop Talking and Start Listening," in *Infocom*, April 2009.
- [123] R. Groenvelt, P. Nain, and G. Koole, "The Message Delay in Mobile Ad Hoc Networks," *Performance Evaluation*, vol. 62, pp. 210–228, 2005.
- [124] M. Ibrahim, A. A. Hanbali, and P. Nain, "Delay and Resource Analysis in MANETs in Presence of Throwboxes," *Performance Evaluation*, vol. 24, no. 9–12, pp. 933–945, October 2007.

- [125] P. Morandi, *Field and Galois theory*. Springer Verlag, 1996, vol. 167.
- [126] J. Milne, "Fields and galois theory," *Courses Notes, Version*, vol. 4, 2003.
- [127] T. G. Kurtz, "Solutions of Ordinary Differential Equations as Limits of Pure Jump Markov Processes," *Journal of Applied Probability*, vol. 7, no. 1, pp. 49–58, 1970.
- [128] A. Ali, E. Altman, T. Chahed, M. Panda, and L. Sassatelli, "A new reliable transport scheme in delay tolerant networks based on acknowledgments and random linear coding," in *Teletraffic Congress (ITC), 2011 23rd International*. IEEE, 2011, pp. 214–221.
- [129] K. Price and R. Storn, "Differential Evolution: A Simple and Efficient Heuristic for Global Optimization Over Continuous Spaces," *J. Global Optimiz.*, vol. 11, pp. 341–359, 1997.
- [130] M. Liu, Y. Yang, and Z. Qin, "A survey of routing protocols and simulations in delay-tolerant networks," in *Wireless Algorithms, Systems, and Applications*, ser. Lecture Notes in Computer Science, Y. Cheng, D. Eun, Z. Qin, M. Song, and K. Xing, Eds. Springer Berlin / Heidelberg, 2011, vol. 6843, pp. 243–253.
- [131] S. Alouf, E. Altman, and P. Nain, "Optimal on-line estimation of the size of a dynamic multicast group," in *Infocom'02*, vol. 2, June 2002, pp. 1109–1118.
- [132] M. Grossglauser and D. Tse, "Mobility increases the capacity of ad hoc wireless networks," *Networking, IEEE/ACM Transactions on*, vol. 10, no. 4, pp. 477–486, aug 2002.
- [133] P. Bremaud, *Point Processes and Queues: Martingale Dynamics*. Springer, 1981.
- [134] A. Jazwinski, *Stochastic Processes and Filtering Theory*, ser. Mathematics in Science and Engineering. Academic Press, 1970, no. v. 63.
- [135] D. Simon, *Optimal State Estimation*. John Wiley & Sons, Inc., 2006.
- [136] T. G. Kurtz, *Approximation of Population Processes*. Society for Industrial and Applied Mathematics, 1981.
- [137] S. Ethier and T. Kurtz, *Markov Processes: Characterization And Convergence*, ser. Wiley Series in Probability And Statistics. Wiley Interscience, 2005.
- [138] I. Karatzas and S. E. Shreve, *Brownian Motion and Stochastic Calculus*. Springer-Verlag, New York, 1988.

- 
- [139] A. E. Bryson and L. J. Henrikson, “Estimation using sampled data containing sequentially correlated noise,” *Journal of Spacecraft and Rockets*, vol. 5, pp. 662–665, 1968.
  - [140] A. Mandelbaum, W. Massey, and M. I. Reiman, “Strong approximations for markovian service networks,” *Queueing Systems*, vol. 30, pp. 149–201, 1998.
  - [141] A. Mandelbaum and G. Pats, “State-dependent queues: approximations and applications,” in *IMA Volumes in Mathematics and Its Applications*, F. Kelly and R. J. Williams, Eds. Springer-Verlag, Berlin, 1995, vol. 71, pp. 239–282.
  - [142] W. Whitt, *Stochastic-Process Limits*. Springer, 2002.





# Thesis Publications

- [C1] **Arshad Ali**, Eitan Altman, Tijani Chahed, Dieter Fiems, Manoj Panda, Lucile Sassatelli, “Estimating File-Spread in a Delay Tolerant Network with Two-hop Routing,” *NETWORKING 2012*, Lecture Notes in Computer Science, Volume 7290, pp. 277-290.
- [C2] **Arshad Ali**, Eitan Altman, Tijani Chahed, Manoj Panda, Lucile Sassatelli, “A New Reliable Transport Scheme in Delay Tolerant Networks Based on Acknowledgments and Random Linear Coding,” *ITC 2011*.
- [C3] **Arshad Ali**, Eitan Altman, Tijani Chahed, Manoj Panda, Lucile Sassatelli, “A New Proposal for Reliable Unicast and Multicast Transport in Delay Tolerant Networks,” *PIMRC 2011, Local and Personal Area Networks Track*.
- [S1] **Arshad Ali**, Manoj Panda, Tijani Chahed, Eitan Altman, “Improving the Transport Performance in Delay Tolerant Networks by Random Linear Network Coding and Global Acknowledgments,” *submitted (Journal)*”.
- [S2] **Arshad Ali**, Manoj Panda, Lucile Sassatelli, Tijani Chahed, Eitan Altman, “Reliable Transport in Delay Tolerant Networks,” *submitted (A Book chapter)*.
- [S3] Manoj Panda, **Arshad Ali**, Lucile Sassatelli, Tijani Chahed, Eitan Altman, Dieter Fiems, “Tracking Message Spread in Delay Tolerant Networks,” *submitted (Conference)*”.
- [US1] “Reliable Transport Proposal in Delay Tolerant Networks Based on Acknowledgments and Random Linear Coding,” *in preparation*”.



# Version Française

Les réseaux mobiles ad hoc (MANET) visent à rendre la communication entre les nœuds mobiles possible sans le soutien de l'infrastructure. Si la densité spatiale des nœuds mobiles dans un MANET est faible, alors un chemin de bout en bout entre une source et une destination n'existe presque jamais; deux nœuds mobiles peuvent communiquer uniquement quand ils entrent dans la zone de couverture radio l'un de l'autre. Grâce à la connectivité intermittente, les nœuds adoptant le paradigme de routage Store-Carry-and-Forward dans lequel une source dépend de la mobilité des autres nœuds, qui agissent comme des "relais", des paquets de données et les accusés de réception sont transférés entre une source et une destination à travers un ou plusieurs relais. Il peut y avoir plusieurs raisons pour une connectivité intermittente tels que la portée radio sans fil limitée, la faible densité de nœuds mobiles, les ressources énergétiques limitées, les attaques, et le bruit.

La motivation pour le développement des MANETs a augmenté au cours des dernières années en raison de l'entrée de dispositifs intelligents avec une capacité de communication sans fil à courte portée. Les MANETs dispersées appartiennent à la classe des DTNs (Delay Tolerant Networks) qui sont des réseaux ad hoc sans fil dispersées et fortement mobiles et où les occasions de transmission sont intermittentes. Dans les DTNs, à n'importe quel moment donné, aucun chemin continu contemporain ne peut jamais exister [8]. Les exemples de DTNs incluent les Réseaux Ad hoc Automobiles dispersés (VANETs) [1], les Réseaux Interplanétaires (IPNs) [2], les Réseaux de Poche Commutés (PSNs) [3], les Réseaux Aéroportés (ANs) [4], les Réseaux Sociaux Mobiles (MSNs) [5] et les Réseaux Sous-marins (UWNs) [6]. Les IPNs sont l'application de communication spatiale de DTNs. Les applications terrestres de DTNs incluent UWNs, PSNs, VANETs, ANs et les réseaux pour développer des régions rurales [7].

## B.1 Motivation pour la Thèse

Dans cette section, nous comparons tout d'abord les DTNs avec les réseaux traditionnels filaires et sans fil en termes de solutions de transport et de routage. Une telle comparaison nous permettra d'identifier certains problèmes importants qui doivent être traités en priorité.

Les protocoles UDP (User Datagram Protocol) [18] et TCP (Transmission Control Protocol) [19] sont les protocoles de transport les plus utilisés dans l'Internet aujourd'hui. Le premier offre un service sans connexion, sans garantie de livraison en raison de la non-confirmation de la livraison des données, le deuxième fournit un service orienté connexion et assure une livraison fiable des données avec accusé de réception. De loin, TCP est le *standard de facto* pour la transmission fiable de données.

Dans le transfert fiable des données, la source de données tient à s'assurer que toutes les informations sont

envoyées correctement et arrivent "en ordre" à la destination. Avec TCP, les paquets de données se sont donnés des numéros de séquence uniques de telle sorte que la destination peut détecter si des paquets sont manquants.

Pour assurer la fiabilité, la destination envoie un accusé de réception (ACK) en retour à la source pour chaque paquet (ou pour chaque deux paquets, en cas de ACKs retardés) reçus sans erreur. Un paquet  $i$  est considéré comme perdu s'il n'est pas accepté dans une certaine période de temps,  $T_0$ , ou si les accusés de réception pour les trois paquets envoyés plus récemment n'arrivent à la source avant l'ACK du paquet  $i$  (dit triple ACK dupliqués). En plus de cela, TCP implémente un mécanisme de contrôle de flux / congestion dans lequel les paquets sont envoyés à l'intérieur de la fenêtre congestion, dont la taille augmente avec la réception des accusés de réception et diminue après l'expiration d'un time-out ou en cas de triple ACK dupliqués.

TCP fonctionne généralement bien dans les réseaux constitués de liens avec de faibles taux d'erreur binaire. Dans les réseaux avec un plus grand taux d'erreur, telles que les réseaux sans fil et mobiles, la plupart des hypothèses formulées par TCP sont violées, ce qui entraîne une dégradation des performances de bout en bout dans de tels réseaux. Par exemple, TCP suppose que la perte de paquets est due à la congestion du réseau et s'adapte bien à la congestion du réseau. Cependant, il interprète mal les pertes de paquets sur un lien sans fil comme étant due à la congestion alors que celles-ci sont dues à la corruption pour cause d'interférences et de bruit. Le mécanisme de contrôle de la congestion de TCP ne peut que faire empirer les choses dans de telles situations. En outre, le protocole TCP nécessite au moins une trajectoire stable entre la source et la destination pour établir la connexion de données utilisé dans la livraison. Toutefois, un tel chemin ne peut pas exister dans les réseaux sans fil. Un compte rendu détaillé sur les limites de TCP dans les réseaux sans fil peut être trouvé dans [20].

Une solution aux problèmes mentionnés ci-dessus ont été fournies dans [21] où les auteurs suggèrent de "signaler" les pertes dues à la corruption des paquets (plutôt qu'à la congestion) de la couche liaison à la couche de transport. Balakrishnan et al. [22] proposent également un protocole "snoop" visant à améliorer les performances de TCP dans les réseaux sans fil. Mahmoudi et al. [23] ont proposé un TCP-aware dynamique algorithme *demande de automatique répétition*, qui est une conception multi-couche qui améliore les performances de TCP dans les réseaux sans fil avec l'adaptation de la couche de liaison.

En plus des taux élevés d'erreurs sur les bits, les MANETs pourraient également souffrir de défaillances de liaison. TCP s'avère être très inefficace pour le transport fiable dans les MANETs, parce qu'il interprète mal les pertes dues aux défaillances de liaison avec les pertes dues à la congestion [24]. C'est encore pire dans le cas de DTNs qui souffrent de ruptures de liens fréquents [25]. Les DTNs sont généralement déconnectés, avec une absence de connexion entre la source et la destination. La plupart du temps, il est impossible d'établir une connexion de données et de confirmer la livraison [26]. Par conséquent, le protocole TCP n'est pas fonctionnel dans les DTNs [27].

Les protocoles de routage traditionnels conçus pour les MANETs comme le protocole Ad-hoc On-demand Distance Vector (AODV) [10] ou le protocole Source Routage Dynamique (DSR) [11] ne donnent de bons résultats que si un lien de bout-en-bout de connectivité existe. En présence de contraintes comme des déconnexions fréquentes, en cas de densité basse du réseau et de capacité fonctionnelle limitée comme dans les DTNs, les protocoles de routage classiques ne seraient pas en mesure de fonctionner efficacement.

Le paradigme de routage store-carry-forward fournit un moyen pour permettre la communication dans des environnements contestés, même si cela implique en soi un grand retard de communication. L'approche de routage store-carry-forward permet aux nœuds de prendre avantage des possibilités de transmission lors de leurs contacts avec d'autres nœuds pour l'échange de paquets.

Ainsi, la stratégie de cette transmission est parfois aussi appelée la routage opportuniste [12]. Plusieurs méthodes de propagation des paquets dans les DTNs ont été étudiées dans le routage opportuniste, par exemple, *le routage de Pulvérisation et-Attendez* [13], *le routage probabiliste* [14], *pulvérisation en plusieurs périodes* [15], livraison directe (DD) et à *le routage deux-sauts* [16] et *le routage épidémique* [17].

En résumé, les protocoles pour les réseaux filaires traditionnels et sans fil semblent ne pas convenir aux réseaux DTNs, ce qui motive le développement de nouvelles approches pour le transport et le routage dans les DTNs. La plupart des propositions de recherche dans les DTNs portent sur le routage. Des solutions de transport sont principalement proposées pour la communication spatiale. Les solutions existantes dans les MANETs comptent sur

signalisation cross-layer entre la couche transport et les couches inférieures de façon à informer la première sur les échecs de route [21], [22], [23]. Cette stratégie ne peut pas être utilisée dans les DTNs puisque seul le routage opportuniste peut être exécuté. Par conséquent, l'objectif principal de cette thèse est la conception d'un protocole de transport fiable efficace pour les DTNs terrestres.

## B.2 Notre Approche

C'est un malentendu commun que minimiser le délai n'est pas important dans les DTNs. En fait, le terme délai tolérant a le sens que les protocoles doivent être conçus de telle manière à ce que les applications peuvent s'exécuter sur le réseau même en présence de longs retards causés par des perturbations fréquentes. Dans cette thèse, la minimisation du délai de bout-en-bout est utilisée comme un critère pour la conception d'un protocole de transport efficace.

Des protocoles basés sur les codes *rateless* sont des alternatives à TCP pour l'Internet du futur, en général, et dans les DTNs, en particulier [28]. Considérons un fichier constitué de  $K$  paquets qui doit être transféré à partir d'une source vers une destination. Des protocoles basés sur les codes rateless permettent à la source et/ou aux relais de continuer à envoyer des paquets nouvellement générés constitués de combinaisons de paquets sélectionnés de manière aléatoire, dits RLCs (Random Linear Combinations), qu'ils ont déjà reçus. La perte d'un RLC peut être compensée par un autre RLC, alors que sans codage, la perte d'un paquet doit être compensée en retransmettant le même paquet. Avec le codage, la seule chose qui importe pour la destination est de répondre à un certain Degree of Freedom (DoF), correspondant au nombre de RLCs linéairement indépendants que la destination doit recevoir pour être en mesure de décoder le fichier, et non la réception des paquets spécifiques.

En l'absence de retour d'information par la destination, la source ne peut pas savoir combien de paquets, qu'ils soient codés ou non, ont réussi à arriver à la destination, et donc seulement une forme probabiliste de la fiabilité est obtenue: la source ne peut garantir qu'une certaine probabilité de réussite de livraison de paquets à la destination. Le codage est utilisé pour améliorer le retard et / ou la performance de débit en augmentant la probabilité de réussite de RLCs reçus dans un délai déterminé. L'ACK est le moyen d'assurer que les paquets/RLCs ont en effet atteint la destination, c'est à dire qu'il apporte une forme déterministe de fiabilité.

Pour atteindre la forme déterministe de la fiabilité et pour réduire les retards dans les DTNs, nous utilisons des accusés de réception, en plus du codage. L'utilisation du codage permet non seulement de réduire au minimum le temps de livraison [29], [30], mais aussi améliorer la forme probabiliste de fiabilité [28]. Toutefois, nous utilisons aussi les ACKs afin d'être sûr que les combinaisons linéaires aléatoires (RLCs), codé sur les paquets, ont en effet atteint la destination, tout comme TCP fait pour les paquets. L'utilisation d'ACK est essentielle pour assurer la forme déterministe de fiabilité.

Comme plusieurs copies du même paquet sont autorisées à se propager dans le réseau, par conséquent, le suivi du nombre de copies en ligne est très important. Pour une livraison de message unique, il est nécessaire d'estimer l'état du réseau pour contrôler un message en propagation. Dans certaines applications multicast/broadcast mobiles, il est également important de contrôler la diffusion du message à un certain pourcentage d'utilisateurs mobiles [31]. Nous abordons ce problème dans le cas de livraison directe et le routage épidémique.

## B.3 Les Contributions de la Thèse

Nous faisons les contributions suivantes dans cette thèse:

- **Transport Fiable:** Ce travail se concentre sur le transfert d'un fichier composé de quelques paquets provenant d'une source unique vers une seule destination à travers des nœuds intermédiaires correspondant à un seul flux. La source envoie des RLCs des paquets originaux et la destination envoie une mise à jour ACK en retour. RLC et ACK sont répliqués de manière épidémique, avec des mesures de contrôle sur la diffusion des

RLCs. Nous ne considérons que le codage à la source. Nous supposons une distribution exponentielle pour les temps de contacts entre les nœuds. Nous faisons également hypothèse que les nœuds relais peuvent stocker au maximum un paquet ou un ACK à la fois. Notre objectif est de réduire les retards et d'assurer la fiabilité déterministe par codage et ACKs.

- **Proposition:** Nous proposons et étudions un nouveau système de transport fiable, basé sur les ACKs et le codage à la source. Notre objectif est d'assurer une forme déterministe de la fiabilité, autrement dit, nous voulons être sûr que les RLCs ont en effet atteint la destination, tout comme TCP fait pour les paquets. Et donc, nous utilisons les ACKs, en plus du codage.
- **Modélisation:** Nous modélisons et évaluons le réseau conformément à notre plan proposé pour donner des perspectives de performance dans le cas d'un grand nombre de nœuds et nous démontrons sa précision par comparaison avec des simulations.
- **Optimisation:** Nous concevons de façon optimale notre système de transport qui minimise le délai moyen aller-retour. Nous réalisons une optimisation du nombre de RLCs envoyer dans un cycle, ainsi que la valeur de la temporisation (time-out), de façon à minimiser le temps de transfert d'un fichier.

Nous trouvons que l'optimisation conjointe d'un ensemble de paramètres optimaux, i.e., nombre de RLCs à envoyer, le temps de propagation des RLCs et les temps d'attente d'ACKs, joue un rôle significatif dans la minimisation du délai. La modification de l'un des paramètres optimaux à un impact sur le délai de livraison. La procédure optimale fournit les valeurs de ces paramètres qui nous permettent d'arriver au délai minimum de livraison.

- **Améliorations:** Nous proposons un nouvel ensemble de mécanismes visant à améliorer (augmenter) la performance du transport fiable à la fois pour unicast et multicast. Nous considérons différentes topologies du réseau constituées d'une ou plusieurs sources et des destinations. Chaque source envoie un paquet vers sa destination prévue. Les paquets de différentes sources sont combinés à l'intérieur du réseau pour former des RLCs (codage inter-session) qui contribue à améliorer *le chemin aller* du transport fiable. Les paquets/RLCs et ACKs sont répliqués épidémiquement. Nous proposons des mécanismes intelligents d'ACKs pour améliorer le chemin retour résultant du transport fiable.
  - Nous proposons et étudions plusieurs fonctions d'amélioration de l'ACK pour améliorer la performance du transport fiable. L'amélioration permet d'augmenter la probabilité de recevoir de l'ACK interne (i.e., l'ACK indiquant DoF manqué à zéro) dans un délai spécifique. De façon équivalente, ils diminuent le temps d'attente (i.e., timeouts) nécessaires pour atteindre une probabilité de réception de l'ACK final. Nous proposons un nouvel accusé de réception Global Sélectif (G-SACK).

Nous apprenons que l'utilisation du codage réseau améliore la probabilité de succès, la variance et la moyenne du délai de transfert des paquets à partir de la source(s) jusqu'à la destination(s), et qu'une utilisation intelligente des ACKs fournit une amélioration énorme dans la probabilité de succès, la variance et la moyenne des retards pour la réception d'information de paquet(s) générée par la destination(s) à la source(s).

- **Poursuite des messages dans la livraison directe et le routage épidémique:**

Dans ce travail, nous considérons un DTN composé de quelques sources statiques, observateurs extérieurs statiques et des utilisateurs mobiles. Les sources et les observateurs sont reliés par un réseau filaire. Nous supposons qu'un nœud observateur entre en contact avec des nœuds relais à et compte le nombre de copies du message. Les nœuds qu'il rencontre les informent du fait qu'il a une copie du paquet. Le problème est d'obtenir une estimation aussi précise que possible du nombre de nœuds avec des copies en utilisant les mesures de l'observateur.

- **Filtre de Kalman:** nous dérivons le filtre de Kalman en temps discret basé sur des approximations de la diffusion pour le processus d'étalement (spreading process) et les mesures.
- **Prédiction:** nous estimons le moment où une certaine fraction de la population a reçu des copies du fichier.
- **MMSE:** Nous dérivons les expressions exactes pour le Minimum-Mean-Squared-Error (MMSE), estimateur basé sur le comptage de l'observateur à l'instant  $t$  dans DD (Direct Delivery) et comparons ses performances avec le filtre de Kalman en termes de précision.

## B.4 Aperçu de la thèse

Le reste de la thèse est organisé comme suit.

Dans la première partie du travail, nous décrivons notre proposition de transport fiable et fournissons un modèle analytique pour notre proposition et son optimisation. Nous proposons également deux approximations à notre proposition pour surmonter la complexité liée au calcul.

Dans la deuxième partie, nous proposons des fonctionnalités d'amélioration de notre transport fiable qui aident à réduire les délais de transmission et permettent de réussir la transmission dans un temps fini.

Dans la troisième partie, nous suivons un message se propageant dans les DTNs sous delivery directe et routage épidémique.

Finalement, nous concluons notre travail et présentons des perspectives des travaux futurs.

## B.5 1ère Partie du travail: Un Nouveau Schéma de Transport Fiable

Nous proposons un nouveau plan de transport fiable pour DTNs basé sur l'utilisation de ACKs aussi bien que le codage linéaire aléatoire. En absence de retour d'information de la destination, la source ne peut pas savoir combien de paquets, ont été codés, l'a fait avec succès à la destination et de là, seulement une *probabiliste* forme de la fiabilité est obtenue. La source peut seulement assurer une certaine probabilité de livraison des paquets à la destination. Le codage est utilisé pour améliorer le retard et/ou les performances du débit en augmentant la probabilité de recevoir le message/fichier dans un temps spécifique. Notre but, cependant, est d'assurer la forme de fiabilité *déterministe*, c'est-à-dire, nous voulons être sûr que les paquets codés aient en effet atteint la destination. De là, en plus du codage, nous utilisons aussi les ACKs pour répondre aux pertes dans le réseau. Des paquets spécifiques ne sont pas importants pour la destination avec codage, au lieu de cela, la seule chose qui compte pour la destination est se rencontrer un certain DoF, correspondance aux nombres de RLCs linéaires indépendant, il doit recevoir pour être capable de décoder le dossier.

Nous développons modèles fluides pour tirer des expressions pour la performance du retard de notre plan de transport fiable proposé. En employant l'algorithme d'optimisation *évolution différentielle* proposé par Prix et Storn [119], nous obtenons les valeurs optimales pour le nombre de combinaisons linéaires aléatoires à envoyer avant la fin du cycle aussi bien que la valeur optimale du temps du cycle lui-même, qui, à son tour, minimise le temps de transfert du dossier. Nous invitons le lecteur à se référer à [120] et [121] pour le détails sur l'évolution différentielle.

Nous proposons un modèle dans lequel la transmission est organisée en cycles. Au cours d'un cycle, la source envoie un nombre spécifique de RLCs, qui est une fonction du nombre de DoFs manquants un façon dos-à-dos sans attendre aucune réaction de la destination. La source permet aux RLCs de se propager à la destination, puis elle attend les accusés de réception de revenir jusqu'à ce qu'un *cycle time-out* se produit. Chaque RLC reçu par le destinataire déclenche le renvoi d'un ACK, indiquant à la source le nombre des DoFs toujours manquants à la destination. Afin qu'aucun flux ou sous-ensemble des flux monopolise les tampons des relais, nous permettre une



mécanisme que l'on appelle *emph* tampon expiration du délai pour chaque individuel RLC dans la mémoire tampon du relais, à la fin de laquelle la spécifique RLC est abandonné. Le cycle se termine à la fin du cycle de temporisation, à laquelle, la source envoie les nouvelles RLCs, dont le nombre dépend des DoFs manquants comme indiqué par le dernier ACK que la source a reçu à ce jour. Le processus est répété jusqu'à ce que l'achèvement de la transmission de l'ensemble du dossier soit rélé. C'est à dire jusqu'à ce que la destination a reçu tous les DoFs nécessaires pour récupérer le dossier.

Tout d'abord, nous développons un modèle analytique pour notre schéma proposé, que nous appelons " le modèle proposée". Ensuite, pour avoir les DoFs manquants égal à 1, nous validons notre modèle en comparant les résultats d'analyse fournis par la procédure optimization contre les résultats de simulation obtenus. Ensuite, en raison de la complexité du computation optimal, nous proposons deux approximations qui éliminent la complexité du calcul des paramètres dans une large mesure, mais encore fournissent des résultats assez précis.

**Nos Contributions:** dans cette partie, nous faisons les contributions suivantes:

- Nous proposons un nouveau schéma de transport fiable pour DTNs basé sur ACKs et le codage à la source.
- Nous développons des modèles fluides pour la dynamique du réseau conformément à notre schéma proposé et démontrons leur précision par comparaison à l'aide ldes simulations.
- Notre schéma de transport est conçu pour minimiser le délai moyen aller-retour. Nous réalisons une optimisation du nombre de RLCs à envoyer dans un cycle, ainsi que la valeur de la temporisation, de manière à minimiser le temps de transfert du dossier.

**Configuration réseau:** Nous nous concentrons sur le transfert des paquets à partir d'une seule source vers une seule destination correspondant à un seul écoulement . Le réseau se compose de  $N_0 + 2$  nœuds mobiles. Il y a un nœud source, un nœud destination, et le reste  $N_0$  relais. La source veut transférer un dossier constitué de  $M$  paquets à la destination. La source génère des paquets *codés* (RLCs de  $M$  paquets d'information) et la destination renvoie ACK indiquant les nombres de DoFs toujours manquants à la destination.

Les RLCs et ACKs sont répliqués en utilisant *roulage épidémique*. Nous disons que deux nœuds se "rencontrent" quand ils viennent dans la gamme de communication de l'autre. Nous supposons que les successives inter-réunions entre les deux relais spécifiques, par exemple  $i$  et  $j \neq i$ , et entre la source (resp. le destination) et un relais spécifique, sont i.i.d exponentiel variables aléatoire.

Nous supposons que les relais ont une capacité tampon pour stocker au plus un paquet ou un ACK à un moment spécifique. Par ailleurs, pour faire de la place pour les paquets et ACK d'autres flux, un paquet est retenu dans un tampon relais uniquement pour une durée  $\tau_{ue}$ , appelé *buffer expiry time-out*, puis abandonné. Cependant, nous supposons que les ACKs sont jamais abandonnés pour faire place pour d'autres paquets ou ACKs, Les ACKs sont beaucoup plus petits que les paquets et ils contiennent des informations important/précieuses.

Nous considérons le codage à la source. La source génère les RLCs des paquets d'informations qu'elle souhaite envoyer à la destination. Les coefficients aléatoires utilisés pour générer un RLC, appelé *vecteur codant*, est inclus dans l'en-tête du paquet codé.

**Notre Schéma de Transport Fiable:** L'objectif est de transférer un dossier contenant  $M$  paquets d'information de la source à la destination d'une manière fiable. Sur se rencontre avec relais, la source génère RLCs des  $M$  paquets d'information.

Les ACKs générés par la destination et reçus par la source offrent à cette dernière l'information nécessaire pour évaluer la progression du transfert. Avec le codage, l'objectif de la destination est d'avoir  $M$  (DoFs) en accumulant "n'importe quel"  $M$  linéairement indépendants RLCs, et pas de la réception de paquets spécifiques d'informations. En conséquence, l'ACK dans notre schéma indique le nombre de DoFs manquants à la destination, et pas de la réception de paquets pécifiques d'information. Pour assurer la fiabilité, notre schéma évalue la progression du transfert à des intervalles appropriés, appelés cycles, en utilisant les informations de retour fournies par l'ACKs, et puis prend les mesures correctives selon l'information fournie par les ACKs.

Nous proposons trois propositions de transport fiable résumées comme suit:

• **Schéma 1:**

1. Il n'y a pas de conflit entre les ACKs. Les ACKs les plus importants remplacent les ACKs les moins importants.
2. Le conflit entre RLC et ACK existe à l'exception que l'ACK final peut remplacer RLC.

• **Schéma 2:**

1. Il n'y a pas de conflit entre les ACKs. Les ACKs les plus importants remplacent les ACKs les moins importants.
2. Le conflit entier entre RLCs et ACKs existe en contraste avec le schéma 1 où ACK indiquant 0 DoFs manquants à la destination (c'est-à-dire, l'ACK final) peut remplacer tous les autres ACKs.

• **Schéma 3:**

1. Il n'y a pas de conflit entre les ACKs. Les ACKs les plus importants remplacent les ACKs les moins importants.
2. Le conflit entier entre RLCs et ACKs existe.

En effet, le Schéma 1 et Schéma 2 contiennent quelques fonctionnalités supplémentaires par rapport au Schéma 3. Le Schéma 1 contient une fonction en plus que Schéma 2 (c'est-à-dire l'ACK indiquant 0 DoFs manquants à la destination peut remplacer tous les autres ACKs et RLCs)

**Modélisation Analytique et Optimisation:** Soit  $n_s^{(N)}$  et  $n_d^{(N)}$  le nombre des sources et des destinations, respectivement. Soit  $X_k^{(N)}(t)$ ,  $k = 1, 2, \dots, M_i$ , le nombre des relais qui ont une copie de  $k$  RLC à un instant  $t$ . Soit  $Y_l^{(N)}(t)$ ,  $l = 0, 1, \dots, i - 1$ , le nombre de relais qui ont une copie de  $l$  ACK à un instant  $t$ . Notre réseau spécifique contient  $N = N_0$  relais avec  $n_s^{(N_0)} = 1$ ,  $n_d^{(N_0)} = 1$ .

Pour obtenir des résultats numériques, quand  $N$  est grand, nous analysons le réseau à la limite lorsque, que  $N \rightarrow \infty$ .

1. Les taux de rencontre par paires  $\beta_r, \beta_s$  et  $\beta_d$  échange avec  $N$  tel que les quantités  $\lambda_r := N\beta_r$ ,  $\lambda_s := N\beta_s$ , et  $\lambda_d := N\beta_d$ , restent constantes,
2. Le nombres de sources  $n_s^{(N)}$  et le nombres des destinations  $n_d^{(N)}$  change avec  $N$  tel que les ratios  $s := n_s^{(N)}/N$  and  $d := n_d^{(N)}/N$  restent constants, et
3. Les valeurs initiales  $X_k^{(N)}(0)$  and  $Y_l^{(N)}(0)$  échange avec  $N$  tel que les ratios  $x(0) := \frac{X_k^{(N)}(0)}{N}$  and  $y(0) := \frac{Y_l^{(N)}(0)}{N}$  restent constants.

Soit  $S_l$  le temps aléatoire au cours duquel le DoF manquant à la destination change de  $l + 1$  vers  $l$ ,  $\forall l$ ,  $l = 0, 1, \dots, i - 1$ . Afin d'obtenir les équations dynamiques pour les ACKs, nous avons d'abord la condition sur  $S_l = s_l$ ,  $\forall l$ ,  $l = 0, 1, \dots, i - 1$ . Nous avons ensuite résolu les équations dynamiques pour les ACKs ainsi que celle des RLC. Enfin, nous déconditionnons sur toutes les valeurs possibles de la variable aléatoire  $S_l$ ,  $l = 0, 1, \dots, i - 1$  en utilisant leur conjointe fonction de densité.

Soit  $P_{X_k}(t)$ 's et  $P_{Y_l}(t)$ 's, la probabilité que la destination et la source ont reçu le  $k$  RLC et  $l$  ACK, respectivement, jusqu'à l'instant  $t$ .

Appliquant le théorème 3.1 de [127], nous observons que les espérances  $E(X_k^{(N)}(t))$  et  $E(Y_l^{(N)}(t))$  sont bien rapprochés par  $Nx_k(t)$  et  $Ny_l(t)$ , respectivement. Nous fournissons des equations pour notre Schéma 1.

$$\frac{dx_k(t)}{dt} = \begin{cases} (s\lambda_s q_k(t) + \lambda_r x_k(t))(1 - x(t) - y(t)) - d\lambda_d x_k(t) \\ -\beta_e x_k(t) - \lambda_r x_k(t)y_0(t) \\ \text{for } 0 < t \leq \frac{M_i}{\lambda_s} + \tau_{i,S}, \\ \\ -d\lambda_d x_k(t) - \beta_e x_k(t) - \lambda_r x_k(t)y_0(t) \\ \text{for } \frac{M_i}{\lambda_s} + \tau_{i,S} < t \leq \tau_i. \end{cases}$$

$$\frac{dy_l(t)}{dt} = \begin{cases} 0 & \text{for } 0 \leq t < s_l, \text{ and } 0 \leq l \leq i-1, \\ \\ (d\lambda_d + \lambda_r y_l(t))(1 - x(t) - y(t)) + d\lambda_d x(t) \\ + \lambda_r y_l(t) \sum_{m>l} y_m(t) - \lambda_r y_l(t) \sum_{m<l} y_m(t) \\ \text{for } s_l < t \leq s_{l-1}, \text{ and } 1 \leq l \leq i-1, \\ \\ (d\lambda_d + \lambda_r y_l(t))(1 - y_l(t)) + d\lambda_d x(t) \\ \text{for } s_0 < t \leq \tau_i, \text{ and } l = 0, \end{cases}$$

avec les conditions initiales suivants,  $\forall k = 1, 2, \dots, M_i$ ,  $x_k(0) = 0$  et  $\forall l = i-1, i-2, \dots, 0$   $y_l(0) = 0$  pour  $0 \leq t < s_l$ , et où  $x(t) := \sum_{k=1}^{M_i} x_k$ ,  $y(t) := \sum_{l=1}^i y_l$  et  $q_k(t)$  désigne la probabilité que la source a exactement  $k-1$  rencontres avec des relais à temps  $t$ , fournie par

$$q_k(t) = \text{Poisson}(\lambda, k-1) = e^{-\lambda_s t} \frac{(\lambda_s t)^{k-1}}{(k-1)!},$$

Comme dans [28, 128],  $P_{X_k}(t)$ 's et  $P_{Y_l}(t)$ 's, sont données par

$$\begin{aligned} \frac{dP_{X_k}(t)}{dt} &= \lambda_d x_k(t)(1 - P_{X_k}(t)) \\ \frac{dP_{Y_l}(t, s_{i-1}, s_{i-2}, \dots, s_0)}{dt} &= \lambda_s y_l(t)(1 - P_{Y_l}(t, s_{i-1}, s_{i-2}, \dots, s_0)) \end{aligned}$$

avec des conditions initiales  $P_{X_k}(0) = 0$  et  $P_{Y_l}(0, s_{i-1}, s_{i-2}, \dots, s_0) = 0$ .

Le inconditionnelle CDF  $P_{Y_l}(t)$  est obtenue comme

$$P_{Y_l}(t) = \int_{s_{i-1}=0}^t \int_{s_{i-2}=s_{i-1}}^t \dots \int_{s_0=s_1}^t f_{S_{i-1} \dots S_0}(s_{i-1}, \dots, s_0) P_{Y_l}(t, s_{i-1}, \dots, s_0) ds_{i-1}, \dots, ds_0$$

où  $f_{S_{i-1} \dots S_0}(s_{i-1}, \dots, s_0)$  désigne la densité jointe des variables aléatoires  $S_l$ ,  $\forall l$ ,  $l = 0, 1, \dots, i-1$ .

Nous procédons maintenant à la description de la séquence de cycles. Soit  $\Delta_n$  désigne les DoFs manquants à la destination au début de la  $n$ -ième cycle. Il est facile de voir que,  $\{\Delta_n, n \geq 1\}$  est une chaîne de Markov à espace d'état  $\{0, 1, 2, \dots, M\}$ . La chaîne de Markov  $\{\Delta_n, n \geq 1\}$  commence avec  $\Delta_1 = M$ , et est absorbé dans l'état 0. Soit  $P_{ij}$  désigne la probabilité de transition d'un état  $i$  à état  $j$ .

Ainsi, les probabilités de transition  $P_{ij}$ ,  $j = 0, 1, \dots, i-1$ , sont

$$P_{ij} = P_{Y_j}(\tau_i) \prod_{l=0}^{j-1} (1 - P_{Y_l}(\tau_i)),$$

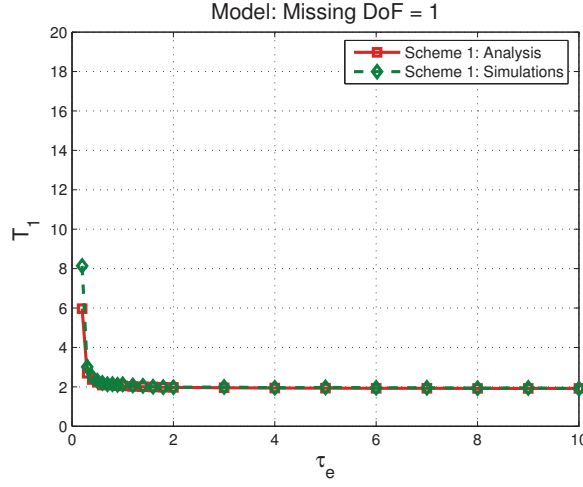


Figure B.1: Schéma 1: les valeurs optimales de  $T_M$  contre  $\tau_e$  et comparaison avec la simulation lorsque DoF manquant = 1.

et

$$P_{ii} = 1 - \sum_{j=0}^{i-1} P_{ij}.$$

Soit  $T_i$ ,  $i = 1, 2, \dots, M$ , le temps prévu pour atteindre l'état de 0 DoFs manquants, à partir du début de  $i$ ème cycle. Clairement,  $T_M$  représente le temps d'achèvement prévu pour le transfert de l'ensemble du dossier. Soit  $E_{Direct}(T_{i \rightarrow 0})$  le temps nécessaire pour atteindre directement l'état 0 dans le même cycle, à partir de  $i$  DoFs manquants. Ensuite,

$$T_i = \sum_{j=1}^i P_{ij}(\tau_i)(T_j + \tau_i) + P_{i0}(\tau_i)E_{Direct}[T_{i \rightarrow 0}],$$

où

$$E_{Direct}[T_{i \rightarrow 0}] = \tau_i - \int_0^{\tau_i} \frac{P_{i0}(t)}{P_{i0}(\tau_i)} dt,$$

ce qui conduit à

$$P_{i0}(\tau_i)E_{Direct}[T_{i \rightarrow 0}] = \tau_i P_{i0}(\tau_i) - \int_0^{\tau_i} P_{i0}(t) dt,$$

En effet, en mettant la valeur de  $P_{i0}(\tau_i)E_{Direct}[T_{i \rightarrow 0}]$  dans l'expression de  $T_i$  et par simplifications, nous obtenons,

$$T_i = \frac{\sum_{j=1}^{i-1} P_{ij}T_j + \tau_i - \int_0^{\tau_i} P_{i0}(t) dt}{1 - P_{ii}(\tau_i)}.$$

**Validation du Modèle :** Figure B.1 correspond au Schéma 1 de notre modèle proposé. On peut observer que le temps moyen de transfert de dossier fourni par procédure optimale et obtenu par simulations sont en bon accord. Cela valide la procédure pour minimiser le temps moyen de transfert du dossier sur la base de notre modèle au se basant sut notre modèle.

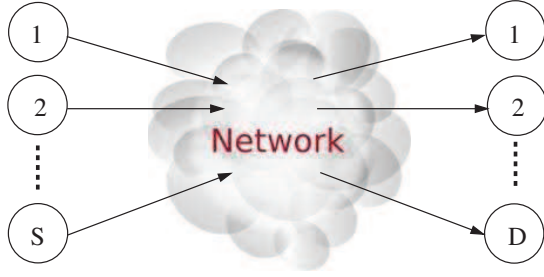


Figure B.2: Le réseau de  $N$  nœuds mobiles (représentée par le nuage) connectés les  $S$  sources et les  $D$  destinations.

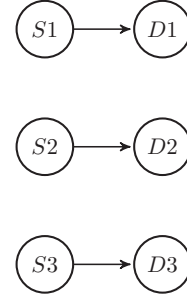


Figure B.3: Three unicast flows.

## B.6 2ème Partie du travail: Améliorations au Transport Fiable

Dans cette partie, nous proposons une nouvelle série de schémas pour améliorer la transport fiable, pour les flux unicast et multicast, dans les DTNs. La fiabilité est assurée par l'utilisation d'un nouveau type d'accusés de réception qui contiennent les informations détaillées sur la réception de paquets à tous les destinations ainsi améliorant le chemin de retour de transport fiable. La motivation pour l'utilisation de nouveaux mécanismes des ACKs provient de l'observation que l'on doit profiter au maximum du contact qui se produit rarement dans DTNs. Nous permettons également le codage linéaire aléatoire à nœuds relais ainsi améliorant le chemin d'aller de transport fiable.

Nous proposons un nouveau mécanisme d'ACK pour améliorer le transport fiable pour DTNs qui est basé sur un nouveau Global Sélectif ACK (G-SACK). Un G-SACK peut potentiellement contenir des informations *globales* sur la réception de paquets à chaque destination dans le réseau. Nous avons aussi utiliser le codage linéaire aléatoire [122] de paquets de différentes sources afin de partager "l'espace payload de paquet" entre les paquets de différents flux dans le réseau.

### Configuration réseau:

Nous considérons un DTN composé de  $S + D + N$  les nœuds mobiles. Il y a  $S$  nœuds *sources*,  $D$  nœuds *destinations* et  $N$  *relais nœuds* (voir figure B.2). Nous considérons *unicast* (une source à une destination) et *multicast* (une source vers plusieurs destinations) transfère de données. En outre, un nœud peut être une destination pour les multiples sources. Les sources envoient de paquets à leurs destination à travers les relais, et les destinations envoient les ACKs, pour chaque paquet reçu, à leurs sources correspondantes par l'intermédiaire nœuds. nous étudions le transfert de paquets avec et sans codage, et avec trois différents Schémas d'ACKs - un simple *schéma baseline* et deux améliorations à celle-ci. Paquets et ACKs sont transférés à travers les relais comme routage épidémique [17].

Chaque paire source-destination  $(i, j)$ ,  $i = 1, \dots, S$ ,  $j = 1, \dots, D$ , définit un *flux*. Un flux multicast  $(i, \mathcal{J})$  se compose de multiples flux avec la même source et multiple destinations, c'est à dire,

$$(i, \mathcal{J}) := \{(i, j') : j' \in \mathcal{J}, \mathcal{J} \subset \{1, \dots, D\}, |\mathcal{J}| \geq 2\}.$$

Le *matrice de flux*  $A = [a_{ij}]$  est une matrice  $S \times D$ , où, pour tous  $i$  et  $j$ ,  $i = 1, \dots, S$ ,  $j = 1, \dots, D$ , l'entrée  $a_{ij} = 1$ , si la source  $i$  a un paquet à envoyer à une destination  $j$ ; telque,  $a_{ij} = 0$ . Par exemple, la matrice de flux correspondant au réseau de la Figure B.3, qui se compose de trois flux unicast, est donnée par

$$A_U = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

On note le nombre de flux par  $N_{pw}$ . Clairement,  $N_{pw} = \sum_{i=1}^S \sum_{j=1}^D a_{ij}$ .

**Limitations et problèmes:** Il y a des limites fondamentales qui doivent être abordées pour atteindre nos objectifs. Les deux principales limites sont *aléatoire* et la *capacité finie*. *Aléatoire* se réfère à l'aspect aléatoire dans le processus relatif à la mobilité des nœuds. La durée des réunions successives entre deux nœuds et la durée pendant laquelle deux nœuds restent à l'intérieur de chaque zone de communication sont aléatoires. Le terme *Capacité finie* se réfère à un nombre fini de relais avec l'espace fini de tampon et des durées finies de contact. Notez bien que *la quantité de données que peut être échangée dans une réunion est déterminée par la durée de contact*.

Le temps de propagation aller-retour associé au transfert d'un paquet est constitué d'une *composante aller* (le délai après lequel la destination (s) recevra le paquet) et une *composante retour* (le délai après lequel la source recevra l'ACK correspondant). Que la capacité (c'est-à-dire, le nombre de relais et/ou l'espace de tampon à relais et/ou la durée de contact) diminue et/ou le temps moyen entre les réunions augmente, valeurs d'aller et le retour attendu, et par conséquent, le délai d'aller-retour, augmente. Le caractère aléatoire dans le processus de réunion/contact, combiné avec une capacité finie, détermine les moments des retards second ordre ou d'ordre supérieur. Le caractère, aléatoire rend le réseau inefficace. L'aléatoire, combiné à la capacité finie, dégrade les performances et pose des questions de non-accessibilité et d'interdépendance. En raison de ces problèmes, un paquet est transféré rapidement ce qui implique que le transfert d'autres paquets devient plus lent.

**Nos améliorations proposées:** Nos améliorations sont:

1. **Codage linéaire aléatoire de paquets de données:** Les paquets sont combinés à des relais pour former des combinaisons linéaires aléatoires. Cela permet à des paquets provenant de différentes sources de partager l'espace paquet payload.
2. **Génération de Acknowledgments Sélectifs à la destination:** À la réception d'un paquet, une destination génère un ACK Sélectif (SACK) indiquant l'ensemble des sources à partir desquelles il a déjà reçu le paquet(s). Ceci est en contradiction avec le schéma de reconnaissance de base, désormais appelé "le schéma ACK", dans lequel seul le paquet en cours de réception est reconnu. Le SACK peut reconnaître des *multiples sources* au sujet de la réception de paquets à une *destination spécifique*.
3. **Mise à jour ACKs/SACKs à l'intérieur du réseau pour former G-ACKs/G-SACKs:** Quand un nœud portant un ACK (ou un SACK) généré par une destination, sur son chemin de retour à la source(s), rencontre d'autres nœuds portant des ACKs/SACK générés par d'autres destinations, les informations contenues dans les ACKs/SACKs sont combinées pour former Global SACKs (G-SACKs). Dans le cas particulier où toutes les destinations génèrent seulement les ACKs et l'information du paquet reçu combinées à l'intérieur du réseau, que nous appelons comme Global ACKs (G-ACK). Un G-SACK (resp. G-ACK) schéma est en contraste avec la SACK (resp. l'ACK) schéma en où les SACKs (resp. l'ACKs) générés par la destination(s) d'atteindre les sources sans être mis à jour à l'intérieur du réseau. Un G-ACK/G-SACK peut reconnaître des *sources multiples* lors de la réception de paquets au niveau de *destinations multiples*.

Nous définissons le *degré* de la source et le *degré de la destination* comme suit:

- **Degrés de la source:** Nous définissons le degré de la source comme le nombre de sources pour lesquelles il s'agit d'une destination.
- **Degrés de la destination:** Nous définissons le degré de la destination comme le nombre de destinations pour lesquelles il s'agit d'une source.

Il s'avère que les avantages de nos propositions s'améliorent avec l'augmentation du degré de la source et/ou le degré de la destination. Des valeurs plus élevées de degré de la source et de la destination, nous permettent également de mettre en œuvre les G-SACKs d'une manière plus efficace comme dans une matrice.

**Métriques de performance:** Nous quantifions le gain par des indicateurs de performance:

**Délais aller:** Il correspond au délai entre l'envoi de la première copie du paquet de la source  $i$  et la réception de la première copie à la destination  $j$  et le noterons par  $D_{ij}^f$ . On note le délais d'aller aléatoire maximal par  $D_{max}^f$ .  $\square$

**Délais Retour:** Il correspond au délai entre la réception de la première copie du paquet de la source  $i$  à la destination  $j$  et la réception de la première copie de l'ACK correspondant à la source et le noterons par  $D_{ij}^r$ . On note le délais de retour aléatoire maximal par  $D_{max}^r$ .  $\square$

**Délais Aller-retour:** Il est simplement égal à la somme du délai d'aller et du retour et le noterons par  $D_{ij}^{rt}$ . On note le délais de aller-retour aléatoire maximal par  $D_{max}^{rt}$ .  $\square$

**La probabilité de succès du trajet aller:** Elle se réfère à la fraction de simulations dans lesquelles le paquet atteint à partir de la source  $i$  à la destination  $j$  dans le temps de la simulation  $T$ , et le noterons par  $P_{ij}^f$ . La probabilité de succès d'aller l'ensemble du réseau  $P^f$  se réfère à la fraction de simulations dans lesquelles les trajets d'aller pour tous les flux dans le réseau sont couronnées de succès au sein de la simulation en temps  $T$ .  $\square$

**La probabilité de succès du trajet retour:** Elle se réfère à la fraction de simulations dans lesquelles l'ACK de destination  $j$  atteint à la source  $i$  dans la simulation en temps  $T$  conditionnée par l'achèvement de trajet aller et le noterons par  $P_{ij}^r$ . La probabilité de succès de retour l'ensemble du réseau  $P^r$  se réfère à la fraction de simulations dans lesquelles les trajets retours pour tous les flux dans le réseau sont couronnées de succès au sein de la simulation en temps  $T$  conditionnée à l'achèvement de la trajet retour pour tous les flux.  $\square$

**La probabilité de succès du trajet aller-retour:** Elle se réfère à la fraction de simulations dans lesquelles le trajet aller-retour entre la source  $i$  et la destination  $j$  est achevé dans la simulation en temps  $T$  et le noterons par  $P_{ij}^{rt}$ . De même, la probabilité ensemble du réseau aller-retour réussite est notée par  $P^{rt}$ .  $\square$

Ici, nous fournissons des prestations de notre protocole proposé par rapport au Schéma de base.

**Le Schéma de Base:** Lorsque le relais  $i$ , qui est vide, rencontre une source, le relais  $i$  reçoit une copie du paquet en provenance de la source. Quand le relais  $j$ , qui est vide, rencontre avec le relais  $i$ , qui a un paquet, le relais  $j$  reçoit le paquet. Lorsque le relais  $j$ , qui a un paquet, rencontre une destination (du paquet), la destination reçoit le paquet et le paquet du relais  $j$  est remplacé par un ACK pour le paquet reçu récemment, appelé l'ACK. Lorsque le relais  $k$ , qui est vide, rencontre le relais  $j$ , qui a un ACK, le relais  $k$  reçoit l'ACK. Lorsque le relais  $k$ , qui a un ACK, rencontre une source (qui est le destinataire prévu de cette ACK), la source reçoit l'ACK.  $\square$

**Le Schéma Proposé:** Notre Schéma propos se compose de (i) codage à relais pour améliorer le trajet d'aller et (G-SACK) pour améliorer le trajet de retour.  $\square$

Dans Figures B.4 and B.5, nous comparons le délai CDFs de l'aller-retour pour une seule paire et le maximum (pour tous les flux), de notre Schéma Proposé avec le Schéma de Base en utilisant la topology de Figure B.3. Le délai moyen, la probabilité de succès et les écarts-types correspondants ainsi que l'amélioration sont donnée dans Tableau B.1

Nous observons que les délais d'aller-retour de paire ainsi que maximum l'ensemble du réseau de notre Schéma Proposé sont stochastique nettement plus petit par rapport à au Schéma de Base. Cela est vrai pour n'importe quelle topology. La diminution du retard moyenne d'aller-retour avec notre Schéma Proposé par rapport au Schéma de Base est en effet remarquable. En plus, la probabilité de succès avec notre Schéma Proposé est très élevée par rapport au Schéma de Base. Nous observons également que la performance du Schéma de Base au diminue avec l'augmentation de la moyenne du degré de la source et degré de la destination, alors que son impact est très faible sur les performances du Schéma Proposé. En fait, le pourcentage d'amélioration du délai moyen l'ensemble du réseau augment avec augmentation de degré de la source/destination pour le Schéma Proposé par rapport au Schéma de Base.

Table B.1: Schéma Proposé sur Schéma Base

Scenario	Forward			Return			Round Trip		
	$P_{11}^f$	$E[D_{11}^f]$	$\sigma[D_{11}^f]$	$P_{11}^r$	$E[D_{11}^r]$	$\sigma[D_{11}^r]$	$P_{11}^{rt}$	$E[D_{11}^{rt}]$	$\sigma[D_{11}^{rt}]$
Basic	0.981	2.2295	3.4012	0.994	16.0823	16.1756	0.975	18.3038	16.8181
Proposed	1.0	1.2661	0.5103	1.0	7.7358	7.0283	1.0	9.0018	7.1703
Improvement(%)	1.94	43.21	85.0	0.62	51.90	56.55	2.56	50.82	57.37
Scenario	$P^f$	$E[D_{avg}^f]$	$\sigma[D_{avg}^f]$	$P^r$	$E[D_{avg}^r]$	$\sigma[D_{avg}^r]$	$P^{rt}$	$E[D_{avg}^{rt}]$	$\sigma[D_{avg}^{rt}]$
Basic	0.937	2.4574	2.5864	0.9785	16.8021	10.2819	0.917	19.2071	10.5719
Proposed	1.0	1.2639	0.2951	1.0	7.8367	4.5957	1.0	9.1006	4.6528
Improvement(%)	6.72	48.571	88.59	2.20	53.36	55.30	9.05	52.62	55.99
Scenario	$P^f$	$E[D_{max}^f]$	$\sigma[D_{max}^f]$	$P^r$	$E[D_{max}^r]$	$\sigma[D_{max}^r]$	$P^{rt}$	$E[D_{max}^{rt}]$	$\sigma[D_{max}^{rt}]$
Basic	0.937	5.0969	7.3967	0.9785	31.3528	19.8724	0.917	34.5955	20.1253
Proposed	1.0	1.7284	0.4293	1.0	13.8886	9.0026	1.0	15.3496	9.0492
Improvement(%)	6.72	66.09	94.20	2.20	55.70	54.70	9.05	55.63	55.04

## B.7 3ème Partie du travail: Poursuite de Diffusion de Message dans les DTNs

Dans cette partie, notre objectif est d'estimer et de suivre le degré de propagation d'un message dans le réseau. En effet, une tel informations en temps réel est critique pour le contrôle en ligne de routage et de dépense énergétique. Il bénéficie également de l'application multi-casting. En effet, depuis plusieurs copies du même paquet sont autorisés à diffuser dans le réseau, il est important de suivre le nombre de copies afin d'avoir en ligne une réplication adaptative politique. Dans cette partie, nous adressons le problème de poursuite du nombre de copies dans le cas de livraison directe [16] et aussi bien que dans le cas du routage épidémique [17].

Il est important de se rendre compte qu'avec seulement une connaissance probabiliste disponible sur la dynamique du réseau, il serait impossible de déduire la diffusion des messages/copies de paquets dans des *réalisations spécifiques* (*trajectoires*). Le problème d'intérêt est suivre une réalisation particulière. Il faut recourir à la prise de mesures et de les combiner avec caractérisation probabiliste de la dynamique des réseaux ce qui permet d'extraire informations importantes sur les trajectoires spécifiques. Pour la prise des mesures, nous supposons qu'il existe plusieurs nœuds *observateur* dans le réseau qui comptent le nombre de copies du message. Notre approche consiste à maintenir le processus de comptage *dynamique* (c'est-à-dire, les IDs des utilisateurs ne sont pas révélés aux observateurs) et *léger* (c'est-à-dire, l'histoire de la rencontre avec des utilisateurs spécifiques n'est pas maintenue) de sorte qu'un assez grand nombre d'observateurs (peut-être tiers) peut être utilisé pour le suivi afin d'obtenir une grande précision.

**Notre Contributions:** Nous obtenons l'approximation fluide du processus de l'écartement. Ce processus permet d'obtenir des informations intéressantes et significatives, voire en absence de mesures. Ce qui permet de déduire la caractérisation probabiliste de la dynamique des réseaux important avec un nombre de nœuds.

Deuxièmement, nous dérivons des approximations de diffusion pour la diffusion et l'observation du processus, et les regroupons avec des observations réelles, afin de suivre le processus à l'aide de Kalman filters.

Troisièmement, nous évaluons la précision du poursuite. Ensuite, nous calculons, dans le cas de livraison directe, les expressions exacte pour l'estimateur linéaire Minimum-Mean-Squared-Error (MMSE) et nous comparons ses performances en termes de précision avec celle filtre de Kalman par rapport à la prévision de temps de passages à niveau.



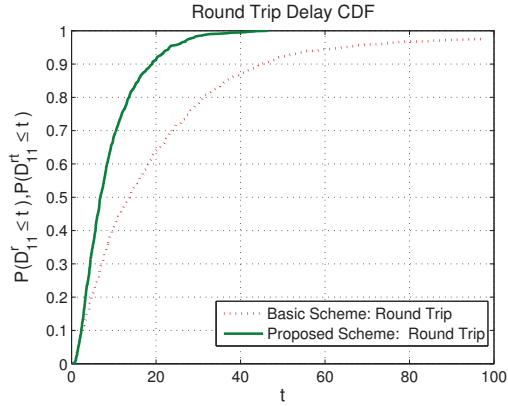


Figure B.4: Comparaison des délai CDFs aller-retour (paires) du Schéma de Base par rapport Schéma Proposé: Unicast Topologie.

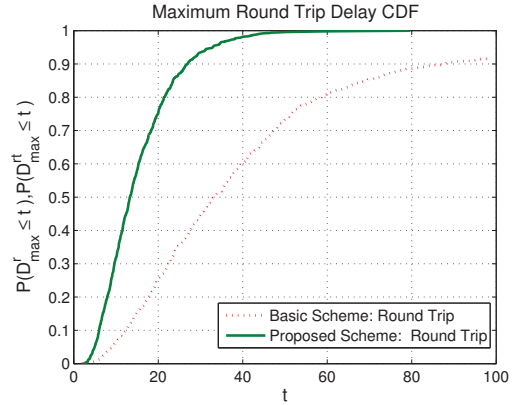


Figure B.5: Comparaison des délai CDFs aller-retour (maximale l'ensemble du réseau) du Schéma de Base par rapport Schéma Proposé: Unicast Topologie.

**Configuration réseau:** Nous considérons un DTN composé de  $S_0$  sources,  $N_0$  utilisateurs, et  $H_0$  observateurs. Les utilisateurs sont mobiles, mais les sources et les observateurs sont *static*. Les sources statiques et les observateurs statiques sont connecté par un réseau filaire. On peut penser à des sources et à des observateurs comme des stations de base (BS) et/ou des points d'accès WiFi (PA), et les utilisateurs en tant que terminaux mobiles (MT) et/ou des véhicules avec un dispositif sans fil. La Figure B.6 illustre un DTN qui compte un déploiement clairsemé de stations de base et une population dispersée de MTs.

**Objectif et approche** Notre objectif est de résoudre les problèmes suivants:

- P1: Analytiquement, prédire la trajectoire moyenne du processus.
- P2: Suivre les trajectoires spécifiques du processus de l'aide de l'histoire de l'observation.
- P3: Estimer le moment où le processus traverse un certain niveau.

Ensuite, nous présentons notre approche pour résoudre P1-P3.

**Problème P1:** pour résoudre P1, Nous obtenons l'approximation fluide du processus. □

**Problème P2:** pour résoudre P2, Nous obtenons l'approximation de diffusion du processus ainsi que pour la observation. Ensuite, nous échantillonne l'approximation de la diffusion du processus (resp. de l'observation) pour obtenir stochastique linéaire équation aux différences en temps discret décrivant la dynamique du système (resp. décrivant la dépendance de des mesures sur le processus). Puis, en utilisant l'équation dynamique du système et l'équation de mesure, et l'application du filtrage de Kalman, nous analysons la trajectoire du processus. □

**Problème P3:** P3 est résolu en utilisant les solutions de P1 et P2. P3 constitue la motivation pour P1 et P2. Cependant, P1 et P2 sont également importants pour le ces du. □

Nous simulons le DTN dans le cas de livraison directe et la routage épidémique dans différents scénarios. Nous observons que les performances de Filtre du Kalman s'améliore avec l'augmentation du nombre d'observateurs ou augmentation du taux de la réunion des observateurs-utilisateurs ou les deux. Les Figures B.7-B.8 montrent les performances du filtre de Kalman pour livraison directe.

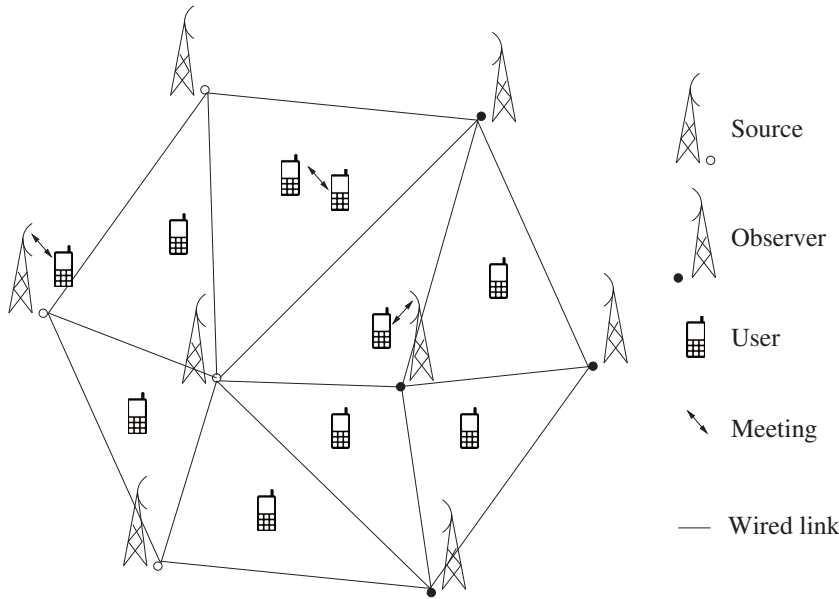


Figure B.6: Un DTN composé de  $S_0$  sources,  $H_0$  observateurs et  $N_0$  utilisateurs.

## B.8 Conclusions et Perspectives

**Conclusions:** Dans cette thèse, nous avons d'abord proposé un nouveau protocole de transport fiable pour les DTNs. Notre protocole proposé est basé sur l'utilisation des ACKs et le codage. Nous avons modélisé l'évolution du réseau dans le cadre de notre protocole en utilisant l'approche fluide. Nous avons optimisé, en utilisant l'approche évolution différentielle, le nombre de RLCs restant à envoyer dos-à-dos avant l'expiration d'un délai d'attente ainsi que la valeur du délai d'attente, de manière à minimiser le temps de transfert du fichier. Nous avons obtenu des temps moyens de transfert de fichier et nous avons démontré, en comparant l'analyse numérique par rapport à des simulations, la validité de nos modèles. Nous avons également démontré que la capacité d'adaptation de notre procédure optimale a des variations de temps d'expiration. Nous avons également proposé deux approximations pour surmonter la complexité de calcul et avons observé que ces approximations sont très proches de notre proposition initiale.

Deuxièmement, nous avons proposé et étudié par de nombreuses simulations une proposition pour améliorer le transport fiable dans les DTNs. Nous avons permis le codage dans les relais. Nous avons observé que l'utilisation du codage permet, non seulement, la livraison rapide du paquet(s) à la destination (s), mais elle permet aussi d'augmenter la probabilité de succès de transfert des données. Nous avons proposé et utilisé des G-SACKs pour atteindre la fiabilité. Nous avons quantifié les gains atteints grâce à diverses caractéristiques de notre proposition.

Enfin, nous avons abordé le problème de l'estimation de propagation du fichier dans les DTNs utilisant la livraison directe et le routage épidémique. Nous avons fourni une base analytique à notre cadre d'estimation. Nous avons également fourni des validations par des simulations. Nous avons obtenu une approximation fluide de la propagation. Nous avons obtenu les approximations de diffusion pour le processus de diffusion et d'observation, et nous l'avons combiné avec des observations réelles, afin de suivre le processus à l'aide de filtre de Kalman. Nous avons évalué la précision de la poursuite. Nous avons aussi calculé, dans le cas de la livraison directe, l'expression exacte pour l'estimateur MMSE et nous avons comparé ses performances en termes de précision avec celle du filtre de Kalman.

**Perspectives:** Comme les nœuds mobiles dans les DTNs fonctionnent sur batteries, la minimisation de la consommation d'énergie, en limitant le nombre de copies de RLCs et ACKs, est donc une question centrale dans la conception de protocoles pour les DTNs. Bien que notre protocole de transport fiable ne se concentre pas sur les

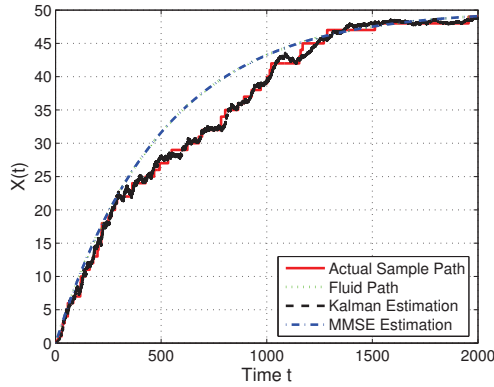


Figure B.7: Performance de MMSE estimation et filtre de Kalman estimation de la processus pour directe delivery.

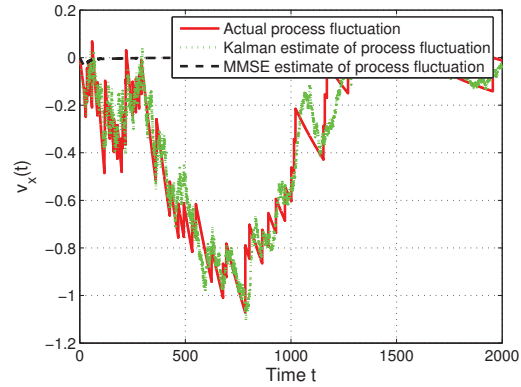


Figure B.8: MMSE estimation et filtre de Kalman estimation de la processus fluctuations pour directe delivery.

questions d'énergie (en rapport avec la réplique de RLCs et ACKs), il convient de mentionner que  $\tau_{i,S}$  et  $\tau_{i,W}$  de notre protocole proposé peuvent être utilisés pour un compromis entre le retard et l'énergie. Par conséquent, il serait intéressant d'étudier le compromis entre la fiabilité et la consommation d'énergie en raison de la réplique des paquets et les ACKs dans les relais.

Ensuite, un grand intérêt pratique sont les scénarios où: (1) plusieurs paquets par flux sont autorisés dans les tampons de relais, et (2) le nombre de paquets d'information par flux est du même ordre de grandeur que le nombre de nœuds dans le réseau. Notre protocole de transport fiable peut être étendu pour couvrir tels scénarios en liaison avec les politiques appropriées de gestion de tampon.

Il serait très intéressant d'évaluer les schémas d'amélioration pour le transport fiable avec plusieurs tampons de paquets et des transferts multiples de paquets entre des paires source-destination.

Il serait également intéressant de concevoir une proposition de transport fiable considérant le codage au niveau des nœuds intermédiaires.

Le problème de contrôle adaptatif du processus de diffusion sous la livraison directe serait également très important à étudier.