Routing and inter-session network coding in delay tolerant mobile social networks

Neetya Shrestha

UNIVERSITÉ NICE SOPHIA ANTIPOLIS
ÉCOLE DOCTORALE STIC
SCIENCES ET TECHNOLOGIES DE L'INFORMATION
ET DE LA COMMUNICATION

THESE DE DOCTORAT

pour obtenir le titre de

Docteur en Sciences
de l'Université Nice Sophia Antipolis
Mention : Informatique

Soutenue par

Neetya SHRESTHA

Routage et Codage Réseau Inter-session dans les Réseaux Sociaux Mobiles Tolérant le Délai
Routing and Inter-Session Network Coding in Delay Tolerant Mobile Social Networks

Soutenue le 29 avril 2015

Jury :

Rapporteur et Président	Prof. Jérôme Lacan, Inst. Sup. de l'Aéronautique et de l'Espace, France
Rapporteur	Prof. Muriel Médard, Massachusetts Institute of Technology, Etats-Unis
Examinateurs	Dr. Katia Jaffrès-Runser, Université de Toulouse, France
	Dr. Thrasyvoulos Spyropoulos, Institut Eurecom, France
Directeurs de thèse	Dr. Lucile Sassatelli, Université Nice Sophia Antipolis, France
	Prof. Guillaume Urvoy-Keller, Université Nice Sophia Antipolis, France

Abstract

We consider Delay Tolerant Mobile Social Networks (DTMSN), made of wireless nodes with intermittent connections and clustered into social communities. This thesis deals with the analysis and design of information transfer strategies in DTMSN. It is mostly dedicated to investigate the use of Inter-Session Network Coding (ISNC) towards this goal. Network coding is a generalization of routing and ISNC is known as a difficult optimization problem in general, specifically because it can easily get detrimental, compared to no coding, if not designed carefully.

To leverage the diversity of the strengths of social ties, a number of utility-based routing policies have been proposed in the literature. In the first part of this thesis, we first address theoretically the optimization problem of the routing policy in DTMSN, under a multi-community network model, and we prove that the optimal policies have a per-community threshold structure, thereby generalizing the existing works for homogeneous mobility DTN. We also compare the online utility-based policies of the literature to these optimal policies.

The second part of the thesis focuses on modeling ISNC-based routing in DTMSN. In particular, we introduce a parameterized pairwise ISNC control policy for heterogeneous DTN, that encompasses both routing and coding controls with an energy constraint. We derive its performance modeling thanks to a mean-field approximation leading to a fluid model of the dissemination process, and validate the model with numerical experiments. We discuss the optimization problem of ISNC control in social DTN.

In order to tackle heuristically the optimization problem, the third chapter presents an experimental study of pairwise ISNC to investigate when it can be beneficial or detrimental. We have found out that, although the contacts are considered bidirectionnal (and asynchronous), which usually prevents NC to be interesting, ISNC can be beneficial when coupled with a socio-aware routing algorithm (SimBet in our study). We examine the impact on ISNC performance of a number of parameters, such as the constraint on the maximum number of copies per packet, the network load, the buffer size of the relay nodes and the buffer management policies.

The fourth chapter addresses the design of decentralized coding criteria allowing to trigger online session mixing if ISNC may be beneficial. We test these coding criteria on both toy topologies and real-world traces, pointing out and explaining the limits of our approach.

Résumé

Nous considérons les Réseaux Sociaux Mobiles Tolérant le Délai (DTMSN), constitués de nœuds sans-fil avec une connectivité intermittente, et groupés en communautés sociales. Cette thèse traite de l'analyse et de la conception de stratégies de transfert de l'information dans les DTMSN. Elle est principalement dédiée à l'étude de codage réseau inter-session (ISNC) dans ce but. Le codage réseau est une généralisation du routage et ISNC est connu comme un problème d'optimisation difficile en général, spécifiquement parce qu'il peut vite devenir nuisible si non conçu avec soin.

Le premier chapitre répond théoriquement au problème d'optimisation du routage (sans ISNC) dans les DTMSN. Nous généralisons les résultants existants pour les topologies homogènes.

Le deuxième chapitre conçoit et modélise un contrle de ISNC par paire, qui englobe conjointement le contrle du routage et du codage, avec une contrainte d'énergie.

Pour s'attaquer de façon heuristique à l'optimisation de ce contrle, le troisième chapitre présente une étude expérimentale visant à identifier quand ISNC est bénéfique ou nuisible, en fonction du nombre maximum de copies par paquet, de la charge du réseau, de la taille de buffer des nœuds relais et de la gestion de buffer.

Le quatrième chapitre présente la conception de critères décentralisés de codage, pour déclencher en ligne le mélange de sessions si ISNC peut être bénéfique. Nous testons ces critères sur des topologies simples et sur des traces réelles, en expliquant les limites de notre approche.

Acknowledgments

I offer my sincere gratitude to my advisor, Dr. Lucile Sassatelli for supporting me throughout this research and providing me invaluable feedbacks, comments and suggestions. Her strong belief in the potential of this research and her continuous encouragement has been a tremendous influence towards the completion of this work. I would also like to thank Prof. Guillaume Urvoy-Keller and Dr. Dino Lopez-Pacheco for their supports.

I would like to thank the members of my jury: Prof. Muriel Médard, Prof. Jérôme Lacan, Dr. Katia Jaffrès-Runser and Dr. Thrasyvoulos Spyropoulos. They contributed their precious time to read my thesis, and provided valuable suggestions and comments that helped to improve the quality of this thesis.

This thesis would not have been possible without the assistance of many people. I do not forget to thank my friends: Sonia, Michelle, Riccadro, Danielle, Lorenzo, Fernando, Jose, Leandro, with whom I spent unforgettable moments. I would also like to thank my colleagues and friends at SigNet group.

Finally, my thanks and deepest gratitude, I address to my dear family, who supported and encouraged me during my years of study.

Contents

Abstract iii

Résumé iv

Acknowledgments v

List of Figures x

List of Tables xii

Acronyms xiv

Introduction 2
Context 2
Motivation 3
Outline 4
Contributions 4

1 State of the Art 7
1.1 Mobile Ad-Hoc Networks 7
1.1.1 The evolution of DTN in communication timeline 8
1.1.2 Characteristics of DTN 8
1.1.3 Applications of DTN 9
1.2 Routing in DTN 11
1.2.1 Routing in homogeneous DTN 12
1.2.2 Routing in heterogeneous DTN: Mobile Social DTN 13
1.2.3 Modeling and performance prediction of routing protocols in DTMSN 16
1.2.4 Buffer management 17
1.3 Network coding 19
1.3.1 Intra-session NC 20
1.3.2 Inter-session NC (ISNC) 22
1.4 Conclusion and open issues 26

2 On Optimality of Routing Policies in Delay-Tolerant Mobile Social Networks 27
2.1 Network model 28
2.2 Optimization results 30

2.2.1 The optimization problem with an energy constraint 31
2.2.2 The structure of optimal routing policies 31
2.2.3 Example of numerical optimization 32
2.3 Distance from optimality of decentralized utility-based policies 33
2.4 Conclusion 36

3 On Control of Inter-session Network Coding in Delay-Tolerant Mobile Social Networks 37
3.1 Network model 38
3.1.1 Inter-session NC 39
3.1.2 The parameterized pairwise ISNC control policy 40
3.2 Modeling dissemination 41
3.2.1 Evolution of the buffer occupancy distribution 41
3.2.2 Evolution of the index dissemination distribution 47
3.2.3 Numerical validation 50
3.3 The ISNC control problem 51
3.3.1 Discussion on the optimization problem 51
3.3.2 Numerical example 52
3.3.3 Relevance to real-world traces 53
3.4 Conclusion 54

4 Experimental study of pairwise Inter-session NC in DTMSN 55
4.1 The chain topology 56
4.1.1 Starting point 57
4.1.2 Analysis 59
4.1.3 A chain with a hub node 64
4.2 The butterfly topology 66
4.2.1 Preamble: ISNC and epidemic routing 66
4.2.2 Combining ISNC and social routing 68
4.2.3 The case of 2 single crossed sessions 69
4.2.3.1 Buffer and Copy Counter Management 69
4.2.3.2 Simulation Settings 71
4.2.3.3 Simulation Results 71
4.2.4 The case of any number of concurrent sessions 75
4.2.4.1 Buffer Management 76
4.2.4.2 Simulation Results 78
4.3 Conclusion 78

5 Designing online ISNC policies for DTMSN 80
5.1 Preamble: Transmission protocol with several concurrent sessions 80
5.2 Design of the decentralized coding criteria 81
5.2.1 General principle 81
5.2.2 Approximation framework 81
5.2.3 Estimation of quantities 82
5.2.3.1 Criterion I 82
5.2.3.2 Criterion II 83
5.2.4 Real-world traces and comparison on several metrics 87

5.2.4.1 Experimental datasets 87
5.2.4.2 Community detection 88
5.2.4.3 Simulation results 89
5.3 Choosing the coding criterion with an analytical model 90
5.4 Conclusion 92

Conclusions and Perspectives **93**
Conclusions 93
Perspectives 94

A Dissemination modeling of concurrent sessions in DTMSN **96**
A.1 Modeling Dissemination 96
A.1.1 Evolution of the buffer occupancy distribution 96
A.1.2 Evolution of the index dissemination distribution 101
A.1.3 Numerical Results 103

B ISNC transmission algorithm for multiple unicast sessions **105**

Bibliography **108**

Traductions Françaises **119**
B.1 Introduction 119
Contexte 119
Motivation 120
Plan du manuscript 121
Contributions 121
B.2 Résumé du Chapitre 2 123
B.3 Résumé du Chapitre 3 124
B.4 Résumé du Chapitre 4 126
B.5 Résumé du Chapitre 5 127
B.6 Conclusions et Perspectives 128
B.6.1 Conclusions 128
B.6.2 Perspectives 128

List of Figures

1.1 Timeline of communication [1]. 8
1.2 Butterfly network: b_i represents data for i-th connection for i=1,2 [2]. 23
1.3 Illustration of commodities in off-line case. The direction of the poison flows (labeled P) is the reverse of their physical (causal) direction from the canonical butterfly example [3]. 24

2.1 Multi-community network model. 29
2.2 Graphs of communities connections corresponding to two β matrices. 34
2.3 CDF of the delivery delay, under $L = E = 10$. 35

3.1 Generation of an inter-session network coded packet 40
3.2 Evolution along time of the number of infected nodes with packets of different types. 50
3.3 Evolution along time of the number of DoF of each source received by D_1. . . 51
3.4 The community topology considered. Each community is made of 333 nodes and the inter-meeting intensities are in green. For both policies: $K_1 = K_2 = 2$, $K'_1 = K'_2 = 5$, $M = Q = 60$. (a) The non ISNC policy (that is, a policy where only intra-session NC is used). (b) The ISNC policy. 53
3.5 The obtained objective values for the above social DTN topology and policies. . 53

4.1 Chain topology with two stations connected through an Access Point (AP). . . 56
4.2 Chain topology with 3 communities. 57
4.3 Evolution along time of the number of infected nodes with packet of different types taking $K = 2$. 57
4.4 Simulation results for chain topology with 3 communities. 58
4.5 Bandwidth sharing in a network topology with 2 communities. 59
4.6 Performance analysis for network topology with 2 communities. 61
4.7 Chain topology with a hub node. 65
4.8 bidirectional, $K = 1$, $M = 200$. 66
4.9 unidirectional, $K = 1$, $M = 200$. 66
4.10 Butterfly topology: the side (blue-filled) communities are made each of $N = 250$ nodes, while the middle node is a single hub node. All nodes have buffer size 1, unless mentioned otherwise. 67
4.11 (a) All possible routes are depicted in blue and red for each session 1 and 2, respectively. (b) The only routes each session should take so as to get a gain with ISNC. 68
4.12 Butterfly topology considered for experiments. 72
4.13 Simulation results for 2 crossed sessions. 72
4.14 Mean number of nodes infected with each type of packets in each community. . 73

4.15 Similarity, Betweenness and Tie-strength utilities averaged over each community, plotted in log scale. 74
4.16 2 crossed sessions, $M = 200$ 75
4.17 2 crossed sessions, $M = 40$ 76
4.18 4 crossed sessions, $M = 200$ 76
4.19 4 crossed sessions, $M = 40$ 77
4.20 Simulation results for 10 crossed sessions, $M = 200$. 78
4.21 Simulation results for 10 homogeneous sessions (source-destinations pairs picked up uniformly at random in the different communities), $M = 200$. 79

5.1 Simulation results for 2 crossed sessions, criterion I. 83
5.2 Simulation results for 10 crossed sessions, criterion I. 84
5.3 Simulation results for 10 homogeneous sessions, criterion I. 84
5.4 Mean number of nodes infected with each type of packets in each community. . . 85
5.5 Simulation results for 10 cross sessions. 86
5.6 Simulation results for 10 homogeneous sessions. 87
5.7 β value for identified butterfly topology inside Infocom05 trace. 89
5.8 Simulation results for Cambridge dataset with 50 concurrent sessions, each with 40 copies. 90
5.9 Simulation results for Infocom05 dataset with 50 concurrent sessions, each with 40 copies. 90

A.1 $N_{s,1} = 1000$. 103
A.2 $N_{s,1} = 800$. 104
A.3 $N_{s,1} = 500$. 104
A.4 $N_{s,1} = 0$. 104

B.1 Multi-community network model. 123
B.2 Generation of an inter-session network coded packet 125

List of Tables

2.1 Analysis of the optimal threshold policy. 33

2.2 Mean number of transmissions in each community by the LSF policy, under $E = 10$. 35

3.1 Main notation used for ISNC control policy. 39

4.1 Buffer management between two meeting nodes n_1 and n_2. 60

4.2 The values of u_{ij} considered for chain topology with 3 communities. 61

5.1 Characteristics of experimental datasets [4, 5, 6]. 87

5.2 Number of communities and modularity (Q). 88

B.1 Main notation used for Multi-session ISNC model. 105

Acronyms

AP	**A**ccess **P**oint
ACK	**A**cknowledgement
DTN	**D**elay (or **D**isruption) **T**olerant **N**etworks
DTNRG	**D**elay-**T**olerant **N**etworking **R**esearch **G**roup
FIFO	**F**irst **I**n **F**irst **O**ut
GBSD	**G**lobal knowledge **B**ased **S**cheduling and **D**rop
HBSD	**H**istory **B**ased **S**cheduling and **D**rop
HJB	**H**amilton-**J**acobi-**B**ellman
IPN	**I**nter**P**la**N**etary Internet
IRTF	**I**nternet **R**esearch **T**ask **F**orce
ISNC	**I**nter-**S**ession **N**etwork **C**oding
LEPR	**LE**ast **PR**obable first
LSF	**L**ast **S**een **F**irst
MANET	**M**obile **A**d-hoc **NET**works
MCMC	**M**arkov **C**hain **M**onte **C**arlo
MOFO	**MO**st **FO**rwarded first
MMF	**M**ost **M**obile **F**irst
MSF	**M**ost **S**ocial **F**irst
NC	**N**etwork **C**oding
ODEs	**O**rdinary **D**ifferential **E**quations
OR	**O**verlap **R**atio
OW	**O**verlap **W**idth
PSN	**P**ocket **S**witched **N**etworks
RLC	**R**andom **L**inear network **C**oding
RSC	**R**outing-**S**cheduling-**C**oding
SaW	**S**pray-**a**nd-**W**ait

SHLI	**SH**ortest **LI**fe time first
VANET	**V**ehicular **A**d hoc **NET**works
WNA	**W**eighted **N**etwork **A**nalysis

Introduction

Context

In 2002, Kevin Fall coined the term Delay tolerant networking when he was adopting some of the ideas in the InterPlaNetary Internet (IPN) design to the deployment of terrestrial networks [7] and since then significant research has focused on the design of Delay (or Disruption) Tolerant Networks (DTN). DTN are wireless networks that lack continuous connectivity as they do not have any fixed infrastructure. The wireless devices use ad hoc mode of communication to share/transfer information when they come in close proximity to each other and hence, the communication in such network is opportunistic. They rely on mobility of the participating devices that can be random as in extreme terrestrial environments, or deterministic as in a planned network in space. The increasing integration of wireless short-range communication technologies (e.g., Bluetooth and 802.11/WiFi) in the mobile devices provide contact opportunities to communicate making such networks more interesting. When DTN is made of mobile devices carried by human, these networks may cluster into different classes or communities owing to various characteristics such as heterogeneous mobility or resource availability. We refer to such DTN made of mobile nodes clustered into different communities as Delay Tolerant Mobile Social Networks (DTMSN).

The technological advancements, widespread use of portable electronic devices, cheap installations and financial benefits make DTN widely applicable in many environments. While the IRTF (Internet Research Task Force) DTN Research Group [8, 9] focuses mainly on Deep-Space communications, there are many other applications of DTN such as: civilian applications (e.g., Pocket Switched Networks(PSN), Vehicular networks), military operations, sensor networks (e.g., ZebraNet [10]), and social networking. The three main goals in civilian applications are (i) to provide network access to remote communities, (ii) provide cheaper content access by file exchange in ad hoc mode, (iii) to exploit the user interactions to convey information, thereby assist to offload the telecommunication networks and to meet the growing demand of users. We can name some recent prototype implementations of DTN as Bytewalla [11], Liberouter [12], Saami Network Connectivity [13].

Delay Tolerant Networks (DTN) exhibit frequent and long lasting partitions because of the sparse distribution of nodes in the topology. These networks need to operate in different scenarios that involve heterogeneity of standards, intermittent connectivity between adjacent nodes, and lack of contemporaneous end-to-end links. In such challenging environments, the nodes are extremely limited in their resources; such as processing power, memory and network capacity [14]. It is therefore important to take buffer management or queuing issues into consideration. One of the fundamental problems that arises when designing networks that handle disconnection is routing. Before a network can be usable, it must be possible to get data from the source to the destination [15]. Routing in DTN can be viewed as an optimization problem where edges may be unavailable for extended periods of time and a storage constraint exists at each node [14]. The traditional Internet routing protocols and mobile ad hoc network routing protocols assume that the network is connected and there is a contemporaneous end-to-end path between any source/destination pair. In absence of fixed infrastructure, these protocols are not applicable for DTN [16, 17]. Several routing protocols have been proposed for DTN with strategies ranging from flooding to limited-replication approaches. This context introduces routing in DTN and its optimization as one of the important areas of research in DTN.

Motivation

Routing is one of the challenging issues in DTN because there is no guarantee that an end-to-end path exists between source and destination at any instant of time. Owing to the intermittent connectivity, in order to achieve end-to-end communication in DTN, the nodes must rely on the Store-Carry-and-Forward paradigm which inherently entails a delay for communication. To lower the delivery delay, multiple copies of the same packet can be spread. How to spread multiple copies has been investigated in several proposals and many opportunistic routing protocols have been proposed to achieve trade-offs between network resource consumption and protocol performance [18, 19, 20]. However, they provide better performance for DTN where the participating nodes have homogeneous characteristics in terms of mobility patterns and resources.

When the DTN is made of humans, human mobility characterizes the forwarding opportunities. Human mobility exhibits heterogeneous mobility pattern where the nodes cluster into different groups or communities owing to the social relationships (e.g., friendship, trade, status, etc.) [21]. A number of routing policies have been proposed for DTN taking into account the social features to improve the routing performance [4, 18, 22, 23]. This motivates us to design optimal routing policy in DTMSN taking into account the heterogeneous mobility pattern and social features of the network. Additionally, in order to perform efficient routing in DTN, the nodes need to utilize any available information about future link establishments, exploit mobility, and wisely allocate buffer resources to save power because DTN need to operate in resource contrained environments such as limited buffer storage and bandwidth.

To improve the benefit of disseminating redundant packets, coded redundant packets can be generated by the relays instead of or additionally to replicated copies, that is performing

Network Coding (NC) in DTN. NC is a networking paradigm that is a generalization of routing [24, 25]. Specifically, random NC [26] has attracted an increasing interest for DTN [27, 28]. The benefits are increase in throughput, as well as adaptability to network topology changes. There are two types of NC: in intra-session NC, only the packets belonging to the same session are coded together (i.e., combined), while in inter-session NC (ISNC) packets pertaining to different sessions can be combined. ISNC is necessary to achieve optimal throughput in general [29] but represents a difficult optimization problem. In undirected graphs, NC has been proven theoretically to be hardly beneficial [30]. However, even though bidirectional contacts in DTN render the equivalent contact graph undirected, that can be even exacerbated depending on the chosen routing algorithm, the scarce resources (such as buffer and bandwidth) can make ISNC attractive as in wireless mesh networks [31]. That is why we tackle in this thesis the open problem of identifying whether and under what conditions ISNC can be beneficial in DTMSN, and designing an inter-session NC policy to efficiently serve several users, where efficiency refers to metrics such as average delay, delivery probability, or fairness in memory occupancy.

Outline

This thesis encompasses 4 main chapters, that all aim at gradually tackling the problem of designing an inter-session NC policy to efficiently serve several users, where efficiency refers to metrics such as average delay, delivery probability, or fairness in memory occupancy.

Before delving into the problem of ISNC, firstly chapter 2 addresses the problem of optimizing routing policies in mobile social DTN. An optimal forwarding policy is designed under a multi-community network model and heterogeneous mobility, based on mean field approximation leading to a fluid model of the dissemination process. The optimal policy is theoretically verified and compared to the existing online utility-based policies to assess the distance to optimality of the practical routing policies. In Chapter 3, a parameterized pairwise ISNC control policy is designed that encompasses both routing and coding controls with an energy constraint. We also discuss the optimization problem of ISNC control in social DTN. In Chapter 4, we examine ISNC policy on a number of toy topologies to identify when and in what kind of topologies it can be beneficial. We build on SimBet routing protocol an ISNC policy that provide benefits to social DTN. In Chapter 5, we devise decentralized coding criteria allowing the nodes to trigger ISNC based on local information available in the network and test our approach on synthetic and real-world traces.

Contributions

In this thesis, we proposed the following contributions:

- On Optimality of Routing Policies in Delay-Tolerant Mobile Social Networks.

  - We formulate the problem of finding the optimal time-dependent forwarding probabilities between any two communities to maximize the delivery probability by a certain deadline under a given constraint of energy by using mean-field approximations leading to a fluid model of the dissemination process for a multi-community network model.
  - We prove that optimal policies have a per-community threshold structure, thereby generalizing the existing works for homogeneous mobility DTN.
  - We provide a numerical illustration by using a heuristic optimization algorithm, and discuss the comparison of such optimal policies with the existing online utility-based routing policies.

- Dissemination modeling of concurrent sessions in DTMSN.

  - We design a parameterized pairwise ISNC control policy for DTMSN, that encompasses both routing and coding controls with an energy constraint. We present the resulting dissemination protocol.
  - We derive its performance modeling thanks to a mean-field approximation leading to a fluid model of the dissemination process, and validate the model by numerical experiments.
  - We discuss the optimization of ISNC control policy benefits in social DTN, and show that the fluid model can be used to devise such optimal policy that jointly exploits the nodes' social acquaintances and the ISNC. By showing numerical gains, we illustrate the relevance of our control policy that is based on the coarse-grained underlying community structure of the social network.

- Experimental study of pairwise Inter-session NC in heterogeneous DTN.

  - We investigate ISNC on toy topologies, namely chain and butterfly topologies to identify when ISNC can be beneficial or detrimental.
  - We examine the impact on ISNC performance of a number of parameters, such as the constraint on the maximum number of copies per packet, the network load, the buffer size of the relay nodes and the buffer management policies.
  - We build ISNC on the social routing algorithm, SimBet [4] and show that on the set of routes selected by the SimBet, ISNC can be beneficial via simulation results.

- Designing online ISNC policies for DTMSN.

  - We design decentralized coding criteria allowing to trigger ISNC when it may be beneficial based only on the local information available in the network for multiple concurrent unicast sessions in DTMSN.

- We test our approach on the toy topologies and real-world traces.
- We present a simplified analytical model to tackle the optimization problem of ISNC policy explaining certain limitations of our model.

Chapter 1

State of the Art

In this chapter, we introduce background for our research in DTN. We provide an overview on different applications and challenges in this research area amongst which routing is one of the most important challenges. We discuss a list of related works, presenting the routing protocols in homogeneous and heterogeneous DTN. We also present different NC schemes that can bring a number of advantages in different communication scenarios in DTN.

1.1 Mobile Ad-Hoc Networks

The wireless networks have become increasingly popular and deployable these days with the proliferation of mobile devices (smart-phones, laptops, PDAs, etc.) that can be connected via wireless short and medium range communication technologies (Bluetooth, 802.11/WiFi) giving rise to infrastructure-less communication paradigms and applications. Such infrastructure-less wireless networks are called mobile ad hoc networks (MANET) [32, 33]. In such networks, the participating network nodes are mobile and communication is spontaneous. The mobile nodes are enabled to communicate with each other even if there is no direct route between them by using any possible node coming in contact as relays to carry the message. We call such networks as opportunistic networks. The two prominent characteristics of opportunistic networks are: (i) a user provides his personal device as a network node, and (ii) users are a priori unknown to each other. These networks represent complex distributed systems that can freely and dynamically self-organize allowing people and devices to communicate without the need of any fixed infrastructure.

Delay/Disruption Tolerant Networks (DTN) are sparse Mobile Ad-hoc NETworks (MANET) in which the mobile nodes (e.g., pedestrian, vehicles, zebras or under-water sensors) are intermittently connected, frequently partitioned and there is no guarantee that a path exists between a source and a destination at any instant of time [15, 34, 35, 36, 37]. However, node mobility plays a fundamental role in providing end-to-end-connection. In these type of networks, when two nodes enter into communication range, they take this opportunity to transmit the message and relay nodes store and carry them until the next transmission opportunity comes up. The network based on this Store-Carry-and-Forward paradigm inherently entails delay for communication and it is the reason why such sparse MANET are referred to as Delay Tolerant Networks (DTN).

1.1.1 The evolution of DTN in communication timeline

Communication generally is of two types: conversational (synchronous) and epistolary (asynchronous). The Figure 1.1 extracted from [1] shows the timeline for communication. In the beginning of humanity, people communicated via speech and gestures like smoke and fire, then people were able to communicate with somebody who is not next to them by writing letters. With the domesticated horse, roads, ships and railroad, the epistolary communication became more faster and feasible over long distances. After the invention of telegraph in 1833, there was development of telephone, radio, satellite communication and Internet. On the other hand, epistolary communication advanced to using magnetic tape, CD-ROM and USB flash drives. As the distance increases, the conversational model alone cannot handle the full range of everything that we do and we need both the synchronous and asynchronous communication, then came development of DTN-based communication.

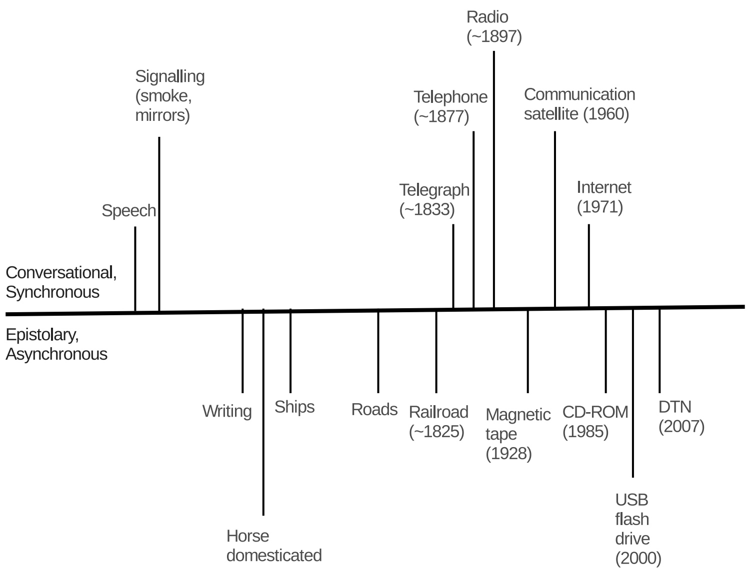

FIGURE 1.1: Timeline of communication [1].

1.1.2 Characteristics of DTN

We present below some of the important characteristics of DTN [7, 34, 38, 39, 40, 41]:

- Sparse: DTN are sparse mobile wireless networks, where the network nodes have very low (typically less than 1) average number of neighbors and they suffer from frequent, often chronic, partitions.

- Disconnection: In DTN, the end-to-end path do not always exists between a source and a destination. The path disconnections are frequent and arises mainly due to two main factors, mainly motion and limited power at nodes.

- High Latency and Low Data Rate: The transmission and propagation delays in DTN can be extremely long and unpredictable, e.g., low data rate of about 10kbps in underwater acoustic modems, no return channel may be available in some military communication, etc.

- Long Queuing Times: The end-to-end delay of data delivery in DTN is often dominated by the queuing time which could be extremely large and in the worst case scenario it may take some hours or days .

- Limited Longevity: The mobile nodes may not survive for long time when they are deployed in hostile environments such as the battle fields and disaster areas. For example, the sensor nodes used for military detection or disaster recovery are brittle to be broken down by the awful surroundings.

- Limited Resources: The participating nodes in DTN have low battery and limited resources such as buffer space and bandwidth.

1.1.3 Applications of DTN

Despite the intermittent connection, node mobility, power outages or delay in communication, these networks find many applications. While the IRTF (Internet Research Task Force) [8] DTN Research Group [9] focuses mainly on Deep-Space communications, the broader DTN research community has investigated delay-tolerant communications for developing countries, vehicular networks, and social networking [14, 42, 43, 44, 45, 46].

- Pocket Switched Network (PSN): It is a social DTN in which people are intermittently connected via different wireless devices such as mobile phones and smart GPS devices [22, 47]. They make use of human mobility and social network features to exchange messages. In [47], Chaintreau *et al.* introduce this kind of network at the Infocom 2005 conference. Similarly, in the Haggle project [48], researchers attempt to establish a people-centric approach to DTN.

- Vehicular Ad-hoc NETworks (VANET): VANET have attracted a lot of attention in recent years and their goal is to provide connectivity to commuters such as public buses, bikes or taxicabs [18, 49]. They are useful for dissemination of location-dependent information such as traffic reports, parking information, etc. In [50], Burgess *et al.* introduce UMass-DieselNet, a DTN testbed composed of 30 buses. Each bus is equipped with 802.11b wireless interfaces and GPS devices and they can transmit data to one another whenever they are within range.

- Military Applications: DTN is of great interest in the military due to the flexibility of the network to expand and contract continuously and remain consistent in a highly changing and challenging environment, e.g., the disconnection occurs due to war or for security reason some links need to be shut down.

We present below some of the recent prototype implementations of DTN:

- Ring Road: It is a low-cost communications satellite network designed to provide high latency but highly robust data interchange capability by using CubeSat satellites in the near-circular low-Earth orbit as “data mules” in a network established by Delay-Tolerant Networking so that most of Earth’s inhabited surface and even the most economically disadvantaged area can access the fundamental information resources of the Internet. While the innately high latency of Ring Road communication rules out some kinds of network services (for e.g., Internet telephony, Skype, teleconferencing, and multi-player games, etc.), can support numerous applications that are delay-tolerant, for e.g., email (one can send email to distant server and addressee picks it up later), news feeds, weather advisories (anything that is a unidirectional data flow and do not need instant response back), warnings of disaster events, requests for relief services (latency is unavoidable but it is important to ensure the receival of message), Social media: Facebook, Twitter, Instragram, etc. (instant response is not necessary). The Ring Road network concept has many advantages: surmounts geographic obstacles, cannot be disabled by earthquake, flood, fire, difficult to disable intentionally, low operating cost, etc.

- Bytewalla: The project “Bytewalla DTN on Android Phones” is hosted by SourceForge.net which is the world’s largest provider of hosting for Open Source software development projects [11]. It allows delay-tolerant networking among Android phone users who travel from villages to cities and need to communicate data, or in areas where existing links need to be avoided for security reasons or in case a government authority shut down the Internet like it happened in some Arab countries during the spring of 2011 [51].

- Saami Network Connectivity: The project Saami Network Connectivity (SNC) seeks to establish Internet communication for the Saami population of Reindeer Herders, who live in remote areas in Swedish Lapland, and relocate their base according to the yearly cycle dictated by the natural behavior of reindeer [13]. Also, the population lack reliable wired, wireless or satellite communication in major areas where they work and/or stay because of topographic circumstances (mountains). The goal of SNC project is to provide communication via delay-tolerant networking supporting the population.

- Liberouter: It is an opportunistic communication network that enables the nearby mobile users to communicate directly without relying on the Internet [12]. The two important approaches of Liberouter are: (i) it relies on direct peer-wise contacts to pass messages like opportunistic networking, and (ii) it does not require direct device-to-device contacts unlike opportunistic approaches but instead introduce “opportunistic routers”. It combines general purpose computing platforms like the Raspberry Pi with SCAMPI platform to produce cheap opportunistic router. Each router can bootstrap and provide services to mobile devices. This framework allows bootstrapping Android devices to become additional routers and install applications. Furthermore, it allows other mobile devices to assist in message forwarding and offers web-based access to content of the neighborhood stored locally.

  In order to use the Liberouter applications, the user must first get the application which can be downloaded from mobile app stores, such as Googles Play Store. After the user installs and launches the Liberouter application, it will connect to the SCAMPI router which when combined with Android can automatically connect to any Liberouter Wi-Fi networks and this enables the device to swap messages with any Liberouter within communication range.

- DTN-Bone: The DTN research group (DTNRG) made DTN-Bone with an objective to establish a worldwide collection of nodes running DTN bundle agents and applications. These nodes cannot take the place of internal test beds in various research organizations; rather, they augment these internal environments in areas such as remote management and control of nodes, interoperability, application deployment and testing, as well as operations across administrative boundaries [52].

1.2 Routing in DTN

Routing is the process of transferring or transporting data from source to destination. The above mentioned characteristics of DTN specially the lack of continuous connectivity and disconnection between end-to-end nodes make routing as one of the challenging problems.

The traditional Internet routing protocols, such as RIP (Routing Information Protocol), OSPF (Open Shortest Path First) assume that a contemporaneous path between two nodes exists and mobile ad hoc network routing protocols, such as DSDV (Destination Sequenced Distance Vector), TORA (Temporally Ordered Routing Algorithm), DSR (Dynamic Source Routing), AODV (Ad-hoc On-Demand Distance Vector) focus on techniques for determining end-to-end contemporaneous paths in a network of mobile nodes to send data [16, 17, 47]. However, those protocols would fail in DTN due to the absence of such end-to-end path and intermittent connectivity. This does not mean messages can never be delivered under intermittent connectivity. If the sequence of connectivity graphs generated over time by nodes’ mobility are overlapped, the end-to-end-path might exist. Several routing protocols have been proposed for DTN that exhibit following basic characteristics: (i) mobility assisted: a relay node stores and carries message for long periods until a new forwarding opportunity arises, (ii) opportunistic: local forwarding decisions are made independently and with little or no knowledge about the destination, and (iii) multiple copies: a message can have multiple copies that can be spread in parallel to increase the probability of message delivery [18, 53, 54]. The principal goal of these routing protocols is to deliver messages to destinations with high success probability and with low end-to-end delay when a path to the destination currently does not exist and there is little or no information about the network.

One of the characteristic of DTN is the distribution of the node inter-contact durations and based on this we can classify DTN into two categories: (i) homogeneous DTN, and (ii) heterogeneous DTN.

1.2.1 Routing in homogeneous DTN

In homogeneous DTN, all the participating nodes have uniform mobility pattern, i.e. the distribution of inter-contact times across the nodes is identical. There are several routing protocols designed to work in homogeneous DTN and we present some of them below:

Epidemic routing [55, 56, 57], also called flooding is one of the simplest routing protocol in DTN. In this routing protocol, a mobile node that has a message keeps on relaying the message to another node that comes within its transmission range and that does not yet have the message. It floods the entire network exploring all opportunistic paths from the source to the destination. It provides high probability of delivery and minimize delay but it wastes a lot of network resources in terms of memory used in relaying nodes and in terms of energy used for transmitting multiple copies of the message. Moreover, most mobile nodes in DTN have limited energy and may prefer fewer transmissions to prolong network lifetime. For these reasons, probabilistic routing [19] and Spray-and-Wait (SaW) [20] are proposed.

In SaW [17, 58] routing scheme, few message copies are spread into the network and then each of the nodes that received a copy in the first phase is independently allowed to give it to the destination. This approach has following advantages: (i) it consumes less resources than epidemic routing, (ii) it can be controlled to achieve trade-offs between network resource consumption and protocol performance by appropriately choosing the number of copies. The question arises what are the best nodes to give the packet to for optimal routing. It has been proven that, if nodes are homogeneous and mobility is stochastic and IID, it is optimal to hand over the copies greedily to the first encountered nodes [18]. But greedy replication may fail if nodes in the network are heterogeneous and majority of nodes spend their time mostly around the same nodes (e.g., employees in same office, animals in same herd) while only a small number of nodes tend to move between disconnected parts of network (e.g., vehicles).

In [59], Altman *et al.* present optimal control problem in homogeneous DTN for epidemic routing and two-hops routing. The forwarding scheme which they used is probabilistic where a message is forwarded to a relay with some probability p at every transmission opportunity. They consider two optimization approaches: (i) static approach: a message is forwarded with constant probability and (ii) dynamic approach: this forwarding probability is allowed to vary in time. They formulate optimal control problem for both static and dynamic approaches based on fluid model of the system's dynamics. The fluid models are the deterministic approximation of the stochastic spreading of messages. Their objective is to maximize the fraction of delivered messages and to minimize the delay under some energy constraints within a given deadline. They show that static control policy, i.e. probabilistic forwarding with constant probability are suboptimal while time-dependent dynamic control problem is optimized by threshold type policies.

Our work differs from above mentioned work as we consider heterogeneous DTN. In Chapter 2, we extend the work of [59] to identify the structure of optimal routing policies in heterogeneous DTN, under a multi-community network model with heterogeneous mobility.

1.2.2 Routing in heterogeneous DTN: Mobile Social DTN

In heterogeneous DTN, the participating nodes can have diverse characteristics and heterogeneous mobility patterns. The distribution of inter-contact times across the nodes is non-identical such that a network divides into several groups or communities. Below we present a selection of routing protocols for heterogeneous DTN, that we consider representative of the existing algorithms for this purpose.

BubbleRap [23] is one of the routing protocols designed for DTMSN. It focuses on two specific aspects of society: community and centrality. Each node can belong to one or more communities. Therefore, each node has local ranking within its local community and global ranking across the whole system and if a node belongs to multiple communities then, it can have multiple local rankings. Within a community, some people are more popular and interact with more people than others (i.e., have high centrality). If a node has message to forward to, first it bubbles the message up using the global ranking system until it reaches a node which belongs to the same community as the destination node. Then local ranking system is used which continues to bubble the message up until it reaches the destination node. It does not require every node to know the ranking of every other node in the network. It only requires the two nodes meeting to be able to compare the ranking and push the message towards the destination using a greedy approach. This scheme improves the forwarding efficiency significantly with much lower resource utilization compared to flooding and PROPHET algorithm.

SimBet [4] is another routing protocol for DTMSN that uses social network analysis metrics to support forwarding schemes providing efficient message delivery. Centrality [60, 61] has been shown to be useful for path finding in the static networks, however, it does not consider time-varying nature of links and a link to a central node may not be available or there is congestion on highly central nodes. To overcome this SimBet uses three different metrics, namely, similarity, betweenness and tie strength.

We detail on the SimBet protocol and the utility calculation because we use it; (i) first as a proof of concept to show that ISNC makes sense in DTMSN operated with a social-based routing policy, even though these networks are in essence bidirectional in Chapter 4, and (ii) then we build on SimBet to design a decentralized coding criterion in Chapter 5.

Similarity utility is calculated as the sum of common neighbors between the current node and a destination. Betweenness utility is calculated using an ego network [62, 63] representation of the nodes with which ego node has come in contact. It measures the extent to which a node lies on the geodesic paths linking other nodes. Tie-strength utility is an aggregation of three indicators, namely frequency, intimacy and recency. The frequency indicator is based on the frequency with which a nodes is encountered. The intimacy indicator is based on the amount of time the node has spent connected to a given node and the recency indicator is based on how recently the node has been encountered. The betweenness utility $BetUtil_n$, the similarity utility $SimUtil_n$, and the tie strength utility $TSUtil_n$ of node n for delivering a message to destination node d compared to node m are given by

$$BetUtil_n(d) = \frac{Bet_n}{Bet_n + Bet_m},$$

$$SimUtil_n(d) = \frac{Sim_n(d)}{Sim_n(d) + Sim_m(d)},$$

$$TSUtil_n(d) = \frac{TieStrength_n(d)}{TieStrength_n(d) + TieStrength_m(d)}.$$

The SimBet utility, $U(n \to d)$ is given by combining the normalized relative weights of the attributes, where $U = BetUtil, SimUtil, TSUtil$:

$$U(n \to d) = \sum^{u \epsilon U} u(n \to d).$$

The replication method is used in order to share the number of copies among two meeting nodes. Each message is assigned a replication value R. When two nodes meet, if the replication value is $R > 1$ then a message copy is made. The value of R is divided between the two copies of the message. This division is dependent on the SimBet utility value of each node; therefore, the division of the replication number for destination d between node n and node m is given by

$$R_n = \lfloor R_{cur} \frac{U(n \to d)}{U(n \to d) + U(m \to d)} \rfloor$$

$$R_n = R_{cur} - R_n$$

The node with higher utility value receives a higher replication value with this method. If $R = 1$, then the forwarding becomes a single-copy strategy where a copy is forwarded from a sending node to receiving node only if the later node has higher SimBet utility and deleted from the former node.

PeopleRank [64] is a social distributed algorithm which is also based on the social information available in the network. Similar to the PageRank algorithm developed for web pages that identifies the most popular nodes by ranking, PeopleRank gives higher rank to the nodes that are socially connected to other important nodes in the network. But PageRank has centralized data information while PeopleRank is implemented in a distributed fashion and is able to provide an end-to-end delay and a success rate close to flooding algorithm while reducing the number of retransmission.

Bulut *et al.* [22] consider a network model reflecting the social structure based on the node interaction. This model is presented in the next section. They assume the network is divided into m communities (C_1 to C_m) and there are N_i nodes in each community C_i. The nodes of community i meet the nodes of community j with an average inter-meeting time of β_{ij} (i.e. they contact each other after each t time units where t is an exponential distributed with mean β_{ij}) such that $\beta_{ii} > \beta_{ij}$. If L is the total copies that can be spread in the network, source distributes $L_{in} - 1$ copies to nodes to the other nodes that in same community as itself and remaining $L - L_{in}$ copies to nodes that are in the same community with the destination. Each of these nodes can deliver message independently on meeting the destination. The experimental results show that routing based on community structure outperforms the routing with normal spraying both in terms of average delivery delay and average copies used for each message. However, they limit their experiments to only two communities: source and destination communities in the network while there could be several intermediate communities.

In [18], Spyropoulos *et al.* present utility-based replication scheme, where instead of greedily handing over the copies to first encountered nodes, the relays are chosen carefully according to some utility function. They examine their protocol in heterogeneous DTN where nodes have different capabilities and mobility patterns. They present three different replication schemes: (i) LSF: a relay node is chosen that has seen the destination most recently, (ii) MMF: utility of a node is calculated based on mobility and a relay node is chosen that is most mobile in the network, and (iii) MSF: utility of a node is calculated based on its sociability and a relay node is chosen that is most social in the network. They propose an analytical framework based on fluid models for epidemic routing. The basic concept is that nodes already having a copy of message (called infected nodes) may forward an additional copy to node that does not have a copy yet (called susceptible nodes). A fluid model is then formulated to capture the rate of message propagation ("infection") among the nodes until the expected time when a destination node receives the message. Their work clearly shows that utility-based replication improve the performance over the greedy replication.

1.2.3 Modeling and performance prediction of routing protocols in DTMSN

We present here two aspects of modeling in the context of DTMSN operation. First, we discuss the network model we consider, and that stems from social network analysis. Then, we discuss the existing performance modelings of protocols.

For the problems at hand, the decisive characteristic of the network is the nodes meetings. Based on the findings of [65], we assume that the meetings of any pair of nodes are Poisson distributed, and the inter-meeting intensity is defined as the mean number of meetings per time unit. This allows the tractability of the models presented below and those derived in this thesis. We consider the multi-community model of [18, 22]. The network is made of N nodes clustered into C communities, each made of N_c nodes, $c \in \{1, C\}$. The inter-meeting intensity is assumed to be the same, β_{ii}, for all pairs of nodes pertaining to community i, while β_{ij} denotes the inter-meeting intensity of any pair of nodes in community i and j. The concept of community imposes that $\beta_{ii} > \beta ij$, for all $i \neq j$.

To analyze the performance of a protocol operating a DTN, be it for, e.g., routing or content placement [66], two main approaches prevail. First, using a fluid model of the dissemination process, thanks to a mean-field approximation, allows to obtain, in closed form or numerically by solving differential equations, performance metrics. This is the case of models derived and used in different contexts in [18, 59, 67] and in this thesis. The detailed introduction of such models is performed in the beginning of Chapter 2. The other set of models resort to a Markov model of the states in which the network can evolve over time, and that are governed both by the random meeting process and the algorithm under scrutiny. We detail two of these works in the next two paragraphs.

In [68], Diana and Lochin propose Markov chain-based model to obtain the end-to-end delay for homogeneous and heterogeneous DTN under binary SaW routing protocol. The number of states of the Markov chain depends on the maximum number of copies L allowed to spread in the network (203 states for $L = 32$), so the high computational complexity prevents their method from being applicable to large networks. Furthermore, they consider spreading of packet assuming that there is no packet of other session, meaning that they do not consider the case of contending sessions. Our work differs as we present the dissemination modeling of several concurrent sessions in DTMSN in Chapter 3.

In [69], Picu and Spyropoulos introduce DTN-Meteo which is an analytical model designed for heterogeneous DTN to predict performance for utility-based algorithms and heterogeneous node contact rates as observed in real world scenario [22, 70, 71]. In this model, DTN optimization problem is formulated using a Markov chain model, where each state (e.g. assignment of content replicas to nodes) is a potential solution, and transition probabilities are defined by two factors: (i) heterogeneous node mobility pattern, which offers solution to the algorithm based on nodes inter-contact probabilities and (ii) the algorithm, which orders the states using a utility function and accepts better ones deterministically (greedy) or randomly. It allows to derive interesting performance metrics such as delay and delivery probability using transient analysis of Markov chain.

The main drawback of DTN-Meteo is that the number of states depends both on the number of nodes and on the maximum number of replicas (exponentially). The authors of [69] tackle this state explosion by introducing a so-called “copy independence”: assuming that copies are very likely not to compete to access the same buffer when the ratio between the maximum number of copies L and the number of nodes is low enough, they come up with the desired final performance metric by considering L parallel Markov chains that are then merged. The number of states also scales with the buffer state, that is what kind of packets can be stored. Also, the performance of only one session is considered in DTN-Meteo, and a session is assumed to be made of only one packet. Compared with this Markov chain-based model, the fluid model of the multi-community scenario we derive and use throughout this thesis allows to derive performance that account for several packets per session, background traffic (though only pairwise), and has a complexity that scales neither in the number of network nodes nor in the number of replicas. Instead, it scales in the number of packets per session, the buffer size and the number of communities. The great advantage of DTN-Meteo is that it takes into account the full heterogeneity of the network, where each pair of node can have a different (Poisson) distribution of contacts. The latter advantage of scale-freeness in the number of nodes of the model we consider comes at the expense of restricting the network heterogeneity: we consider a finite set of communities made of similar nodes. We discuss again the link between DTN-Meteo and our fluid model at the end of the last chapter of the thesis.

1.2.4 Buffer management

The routing protocols in DTN are based on store-carry-and-forward paradigm. The messages must be buffered for long periods of time in order to cope with disconnections and the relay nodes require enough buffer space to store all the messages until the future communication opportunities arise. However, the nodes in DTN usually have limited buffer capacity and therefore, when the buffer is full, nodes cannot exchange all the data resulting in data loss. Intelligent buffer management schemes are required that will allow each node to smartly decide which messages to drop when its buffer is full so as to maximize the delivery rate for all the messages in the network.

In [72], Ioannidis *et al.* propose PSEPHOS, a caching policy for network with heterogeneous environment. PHEPHOS is (i) adaptive: it does not require a priori knowledge of user demands or mobility patterns, but adapts to them while constructing the caching policies, (ii) distributed: each user computes its caching policy individually, by exchanging messages with its neighbors, (iii) simple and easy to implement: users maintain a “vote” for each possible item they can store which reflect the amount of requests for content a user receives, and (iv) optimal: the caching policies selected by this mechanism maximize the aggregate utility (i.e. the social welfare) of users participating in the system. This policy however was designed for a peer-to-peer content sharing system over a network of mobile devices, and therefore cannot be directly implemented in DTN which lacks fixed infrastructure and has inherent delay.

Maxprop [50, 73] is a DTN routing protocol which is based on prioritizing both the schedule of packets transmitted to other nodes and the schedule of packets to be dropped when buffers are low on space. Packets are ranked based on a cost assigned to each destination. The cost is an estimation of delivery likelihood. The packets ranked with highest priority are the first to be transmitted during a transmission opportunity and the ones ranked lowest are the first to be deleted to provide space for incoming new packets. If two packets have destination with the same cost then the tie is broken giving priority to the one that has not traversed far in the network. In this way, it provides a way to give new packets a head start in the network by giving them higher priority. Additionally, acknowledgements are sent to all the nodes in the network once the packet has been delivered allowing the packet to be deleted and providing opportunity to store and transmit other packets.

RAPID [74, 75] handles DTN routing as an optimal resource allocation problem taking into account bandwidth, buffer constraints and node mobility. Although RAPID outperforms some protocols like Maxprop and SaW, but it has some limitations [41]: it is based on per-packet utility calculation to optimize metrics (e.g. average delay) but in order to derive those utilities it requires the flooding of information, and it does not address the issue of signalling overhead.

Lindgren and Phanse present a number of queue management policies for intermittently connected networks [76] such as (i) FIFO (First In First Out): the message to be dropped first is the one that first entered into the queue, (ii) MOFO (MOst FOrwarded first): the message to be dropped first is the one that has been forwarded the largest number of times , (iii) SHLI (SHortest LIfe time first): the message that has the shortest remaining life time is the first to be dropped and (iv) LEPR (LEast PRobable first): the message that has lowest probability to get delivered is dropped first.

Most of the routing policies focus on the selection of best nodes to give the packet to and they have an incidental effect on desired performance metrics such as average delay or delivery probability. For example, SaW and other routing protocols [18, 19] that route the packets using the number of copies as the heuristic to enhance the performance do not take into account the storage constraints which raises the question if they could provide the same performance in the resource constrained environment. In [67], Zhang *et al.* investigate how resources such as buffer space and the number of copies made for a packet can be traded for faster delivery by employing ODE models coupled with Markov models for buffer-constrained epidemic routing. They present three buffer management strategies: (i) droptail where newly arriving packets are dropped if the buffer is full, (ii) drophead where the oldest packet in the buffer is dropped to accept newly arriving packets, and (iii) drophead-sp (source-prioritized drophead), where the oldest relay packet (i.e., a packet received from other node) is dropped when a packet arrives to the node with full buffer giving priority to packets arriving directly from the node itself. The authors conclude that drophead and drophead-sp outperform droptail in the context of homogeneous DTN.

In [41, 77], Krifa *et al.* present theoretically optimal buffer management policy along with an efficient distributed implementation. They develop a framework based on epidemic message dissemination, and propose a greedy optimal joint content scheduling and storage management policy, GBSD (Global knowledge Based Scheduling and Drop), that can either maximize the average delivery rate or minimize delivery delay in the context of a congested delay tolerant network. In GSBD, per-message utility is derived taking into account all information that are relevant for message delivery. However, it requires global knowledge to derive these utilities, and therefore is not practically implementable given the intermittently connected nature of targeted environments of DTN. Then, to amend this problem, another policy called HBSD (History Based Scheduling and Drop), a distributed (local) algorithm based on statistical learning is proposed. HBSD uses network history to estimate the current state of required (global) network parameters and uses these estimates, rather than actual values (as in GBSD), to calculate message utilities for each performance target metric [78].

In [79], Santos *et al.* present Content-Centric Dissemination Algorithm (CEDO) focusing on end-to-end traffic sent by one device to another well known device. They consider a multicast communication scenario where different contents in the network have different popularity and nodes can store copies for only a small subset of them due to storage limitations. This turns the scheduling and drop problem into a storage allocation problem. They assume that each node generates at random times requests for random contents, and that each such request has some "deadline"(denoted as TTL) after which it disappears from the requesting node and the network. Whenever a node meets another node which has a content that the former node was interested in, the former node retrieves the content and that request is recorded as *satisfied* otherwise *missed*. Each node calculates per content utility that is used to rank the contents it stores. The drop strategy is simply to remove contents with minimum utilities and the scheduling strategy is to forward contents to encountered nodes by decreasing order of utilities.

The buffer management policy greatly impacts the routing performance in DTN. Instead of only replacing or dropping packets as the above mentioned policies do, we analyze in this thesis, specifically in Chapters 4 and 5, what can be the gain of giving the buffer management an extra choice of storing a combination of different sessions' packets.

1.3 Network coding

Network Coding (NC) is a technique which allows the intermediate relay nodes to perform coding operations besides the replication and forwarding [24, 80]. There has been several research showing the benefits of NC for wireless networks such as improving throughput, saving energy and bandwidth, and adaptability to dynamic topology changes. Communication networks generally share same fundamental principal. The information whether it is signal in a phone network or packets over the Internet, is transported in the same way as fluid share pipes, cars share a highway or people share strategies. The independent data streams share the network resources but data itself is separate. Network coding is a generalization of routing where the nodes do not simply forward the packets they have received but they may combine several input packets to one and send the coded packets over the shared network. The successful reception of the information does not depend on receiving specific packet but it depends on receiving a sufficient number of independent packets so that the destination node can decode the coded packets to retrieve the original information packet. It further makes the system robust against link failure and packet loss, and also facilitates the design of distributed algorithms whose decisions are based on partial information available in the network. We classify NC into following two categories:

1.3.1 Intra-session NC

When network coding is considered for a single session for example unicast session from one source to one destination or multicast session from one source to multiple destinations, it is called as intra-session NC. In this type of coding, only the information belonging to the same session are coded together. Each node can form coded packet as random linear combinations of the input packets and each destination node can decode once it receives enough independent linear combinations of the source packets.

Random linear network coding (RLC) has attracted an increasing interests for DTN [26, 27, 81, 82, 83]. It is a simple, randomized coding method which maintains a vector of coefficients for each of the source processes. The vector coefficients are updated at each coding node and these coefficients can be easily accommodated in communication network consisting of error-free links: packet networks. It is a normal practice to store such information (e.g., sequence numbers, session type, etc.) in the packet headers. Suppose a source node has packet P_1 which is random linear combination of K message packets $w_1, w_2, \ldots, w_k$ which are the vectors of length λ over some finite field F_q such that $P_1 = \sum_{k=1}^{K} \gamma_k w_k$. We can take $\lambda = \lceil b/log_2 q \rceil$ if the packet length is b bits. γ is called the encoding vector of P_1 and it is sent along with P_1 as side information in packets header. The received packets are stored in the nodes memory and a coded packet is formed with random linear combinations of its memory contents. When the packets are large, the overhead it incurs (namely, $K log_2 q$ bits) is negligible.

Alswede *et al.* [24] first brought forward the benefit of coding at intermediate nodes in point-to-point communication network with multicast sessions. They characterized the admissible coding rate region for a multicast session and their result can be regarded as Max-flow-Min-cut theorem for network information flow. Given a multicast network represented by directed graph

G, the multicast capacity is defined as minimum of the maximum flows between the source and each receiver.

The advantage of RLC over routing is demonstrated in [26] for distributed transmission in rectangular grid network consisting of multiple source processes. The upper bound on the routing success probability for a source-destination pair in terms of their relative grid locations is surpassed by the corresponding lower bound for randomized coding in sufficiently large finite fields. This lower bound on the success probability of randomized coding holds for linearly correlated sources which implies that linearly correlated information can be compressed effectively to the capacity of any network cut that is feasible.

These results obtained in [24, 26] for wireless networks are not directly applicable to DTN due to their dynamically changing topology.

Erasure codes [84, 85] have been leveraged in the design of routing algorithms for DTN from the onset of the research on this topic. Specifically, legacy schemes where coded packets are generated only by the source have been studied, such as in [86] Wang *et al.* proposed the erasure-coding based routing algorithm to increase the efficiency of DTN under uncertain mobility patterns. The erasure encoding takes as input a message of size M and a replication factor r. This algorithm produces $M * \frac{r}{b}$ equal sized code blocks of size b, such that the original message can be reconstructed with any $(1+\epsilon).\frac{M}{b}$ erasure coded blocks, where ϵ is a small constant and varies depending on the exact algorithm used, such as Tornado codes.The key aspect is that when using erasure coding with a replication factor of r, only $\frac{1}{r}$ of the code blocks are required to decode the message. They compare erasure-coding scheme with simple replication. The replication scheme has either high overhead due to excessive transmissions or long delays due to the possibility of making wrong choices when forwarding a few redundant copies. But erasure coding allows use of a large number of relays while maintaining a constant overhead, which results in fewer cases of long delays. Recently, in [87] Thai *et al.* have introduced Tetrys which is a redundancy adaptation algorithm based on an on-the-fly erasure network coding scheme called. They used Tetrys for transmission of real-time video.

Lin *et al.* [28] present stochastic analytical framework based on ordinary differential equation (ODE) using network coding with epidemic routing in DTN. The advantage of using network coding is that a node can transmit any coded packet because all the coded packet equally contribute to the packet delivery to the destination with high probability. In the realistic environment when the bandwidth and node buffer is limited, epidemic routing using network coding performs better than epidemic routing based on replication by delivering packets with shorter delays. The price that one has to pay for using network coding is that destination node may have to wait for long time to receive enough number of coded packets to be able to decode to get the desired original data packet.

The above mentioned policies [28, 86] consider only intra-session NC. ISNC is necessary to achieve optimal throughput in general [29], but it represents difficult optimization problem, in particular for DTN. We present in next section, some works in literature that address inter-session NC.

1.3.2 Inter-session NC (ISNC)

When there are multiple sessions sharing the network then inter-session NC is necessary where the information belonging to different sessions are coded together. ISNC is more complicated than intra-session NC, and nodes cannot simply combine all the input packets randomly because destination nodes may not have sufficient capacity to decode all the randomly combined source packets. ISNC requires coding to be done in a strategical way to ensure that each destination node can decode to get the desired source packet. In general, it requires inter-session NC to obtain optimal throughput.

Yang *et al.* [88] present linear inter-session network coding for multiple multicast connections. They first divide the sessions into different groups and construct linear network coding for each group instead of using intra-session network coding for combination of all the sessions. They proposed two metrics for session division: Overlap Ratio (OR) and Overlap Width (OW). The OR method measures the overlap by the percentage of the overlapped links between two sessions. It gives higher priority to the sessions that have most common links. The OW method measures the overlap by the percentage of overlapped paths between two sessions. It gives higher priority to the sessions that have the most number of the paths crossed.

In [89], Heindlmaier *et al.* present an approach of using virtual multicast for inter-session network coding in wireless networks. They use flow based optimization problem where a virtual multicast session can be created that combines the packets from different sessions using random linear network codes. Their technique is carpooling approach where packets are traveling in same direction unlike COPE (that use reverse carpooling approach) where the coded packet travel in opposite directions to the original packets. On using inter-session network coding, the performance rate increases than compared to using only intra-session network coding where flows of different sessions cannot be combined to share the resources. The difficulty lies in setting up virtual terminal set and computational complexity.

In case of a multicast session, the joint problem of subgraph selection and coding decisions can be solved independently [90]. However, we are considering multiple unicast sessions in our work and the optimization problem of ISNC for multiple unicast sessions has been proven to be NP-hard [25]. Hence, the works addressing ISNC require sub-optimal solution. We now present works in literature addressing ISNC for multiple unicast sessions.

Traskov *et al.* [2] present two suboptimal network coding construction techniques: linear program and integer program for wired network with point-to-point links. They consider multiple unicast setup for which the problem of subgraph selection and code-construction have to be solved jointly to obtain minimum cost. They model the cost function as minimum cost flow problem and use poison-antidote approach and binary XOR for code construction. Considering

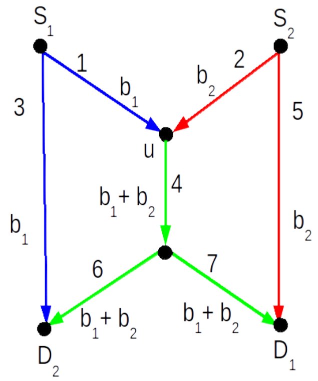

FIGURE 1.2: Butterfly network: b_i represents data for i-th connection for i=1,2 [2].

a butterfly network as shown in Figure 1.2, where all the edges have unit capacity, they verify that $R_1 = R_2 = 1$ (R_i is the rate for communication from S_i to D_i) is feasible with network coding which is not possible with routing. The linear and integer program are quite similar and their main difference lies in computational complexity. The linear program can result in large optimization problem while the integer program is of considerably smaller size and have fewer variables and constraints than the linear program but solving integer programming is NP-complete. They consider different distribution models and observe that there is network coding gains when the capacities of the links are non-uniformly distributed and gains disappear when the link capacities are uniformly distributed. They however considered wireline network with directed network topology and static flows.

Eryilmaz and Lun [29] propose a dynamic Routing-Scheduling-Coding (RSC) strategy for serving multiple unicast sessions when linear network coding is allowed across sessions. The RSC approach does not operate on given flow rates but it operates solely on the queue state information. Their strategy applies to both lossless wireless and wireline networks. They describe the algorithm in a butterfly network and then extend to general wireline networks. A virtual multicast session could exist in the network with multiple unicast sessions and with RSC approach it determines whether it is beneficial to perform inter-session network coding, if yes at which nodes which sessions should be involved in coding, if not which queues to serve.

Kreishah *et al.* [91] develop a distributed algorithm with rate control and utility maximization for inter-session NC for multiple unicast-sessions. They use pairwise random coding scheme which is a modified version of the random linear coding scheme and allows pairwise inter-session NC between any pair of sessions, where coded symbol is formed by coding over at most two original symbols. The pairwise random coding scheme decouples the coding and rate-control decisions and facilitates the development of a fully distributed algorithm. The objective function is the sum of the utilities based on the rates supported by each unicast session. Each unicast session has the freedom to select its own utility function. Combining the distributed rate control and decentralized coding scheme, they eliminate the need to exchange the queue length information among intermediate nodes (such as in [29]) and to find specific structures in the network (such as butterfly structures in [2]). They show that the capacity region obtained via their approach is considerably larger than via pattern search algorithm [2].

Ho *et al.* [3] propose network coding solutions for multiple unicast sessions for both wired and wireless networks. They construct throughput-optimal network codes which is restricted to XOR coding. Their approach is inspired by back pressure techniques, where each node maintains a queue for each session's packet and coding decision is based on queue length information. The two uncoded packets P_1 and P_2 from different sessions can be coded together to form poison packet and they produce two remedy packets each at the nodes previously traversed by P_1 and P_2 respectively. The poison packet is decoded to form the original uncoded packets on meeting any node having corresponding remedy packet. Unlike [29] which requires the coding node to remotely choose the remedy origination locations and decoding locations, in their algorithm the decoding locations are not predetermined but are chosen locally. In off-line algorithm, they

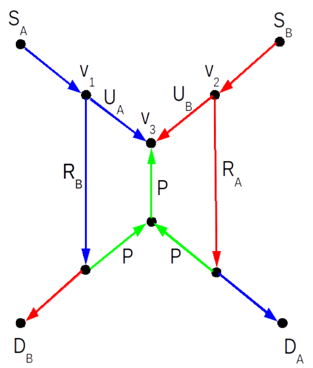

FIGURE 1.3: Illustration of commodities in off-line case. The direction of the poison flows (labeled P) is the reverse of their physical (causal) direction from the canonical butterfly example [3].

reverse the direction of the poison flows without changing the link capacity usage to reduce the computation complexity while online algorithm is constrained to the original flow direction of the packets, i.e., in off-line remedy generate poison on route to the coding node while in online case poison generate remedy as it travels away from coding node as shown in Figure 1.3, where U_A represents uncoded packet of session-A, R_A represents remedy packet for session-A and P represents coded packet.

I^2NC is a constructive network coding scheme proposed for lossy wireless mesh network with multiple unicast sessions [92]. It combines intra-session and inter-session network coding and can perform well even in high loss rate and additionally it does not require the intermediate node to have the knowledge of what its neighbors have overheard to perform coding. Firstly they use intra-session network coding in which they use generation-based network coding approach. At the source, packets within the same generation are linearly combined to generate coded packets. Each intermediate node along the path of a certain flow adds parity packets (redundancy), depending on the loss rates of the links involved for this flow. After an intermediate node has added redundancy, inter-session network coding is applied on top of already intra-coded flows.

Zhang *et al.* [93] investigate the benefits of using both intra- and inter-session NC for unicast applications in homogeneous DTN. For unicast sessions with different sources and destinations, uncontrolled ISNC is shown not to perform better than intra-session NC. They show that when there is only bandwidth constraints, the RLC scheme reduces group delivery delay but its performance is worse in terms of average packet delay and number of transmissions. They use replication control mechanisms to control the number of times a packet is transmitted in the network. With replication control, RLC schemes reduce group delivery delays without increasing the number of transmissions, thereby improving the trade off between delay and number of transmissions. In general, the benefits of RLC schemes are more significant under high resource constraints (limited bandwidth and buffer), high limited signaling, highly dynamic topology and when applied to packets in the same flow. However, they do not consider the heterogeneous mobility pattern while we design parameterized pairwise ISNC control policy for heterogeneous DTN that encompasses both routing and NC and we show that controlled ISNC can bring throughput gain in DTMSN.

Hubcode [94] is hub-based forwarding protocol with network coding designed for people-centric DTN that exhibits power-law behavior. In such networks, small number of nodes called as hubs have higher connectivity than the other nodes in the network. The HubCode uses those hubs as message relays. The hubs employ random linear network coding to encode messages addressed to the same destination, thus reducing forwarding overheads. Moreover, the use of hubs as relays ensures that most messages are delivered to the destinations. They present two version of HubCodes: (i) HubCodeV1: In this case, the hubs encode the messages headed to same direction and the destination node performs decoding to obtain the transmitted packet. (ii) HubCodeV2: In this case, the hubs can also participate in decoding. Since the hubs can decode the coded messages to recover native messages, it reduces the data overhead but it comes at the expense of extra computational complexity than compared to HubCodeV1. But they applied NC across packets destined to the same destination node only. However, in our work we consider multiple unicast sessions with different destinations.

Most of the above studies focused applying inter-session NC on directed network with unidirectional links. However, there is a priori no reason for considering that two nodes can exchange packets in a single direction in DTN.

Li and Li in [30] have shown theoretically what can be the maximum throughput improvement with intra-session NC in undirected networks, compared with what can be obtained with integral, half-integer and fractional routing. In particular, for the multicast problem, the throughput increase ratio is upper-bounded by two between NC and half-integer routing, or even less with fractional routing [30, 95]. However, the shared resources (buffer, contact bandwidth) in DTN make ISNC attractive as in wireless mesh networks [31], though these networks are undirected. Hence, we are tackling the open problem of ISNC design in social DTN. We design tunable pairwise ISNC control policy and discuss the optimization problem of ISNC control in social DTN.

1.4 Conclusion and open issues

In this chapter, we have described the background behind our work starting from the definition of DTN and its various applications. Routing is one of the most important challenges in these type of networks owing to the lack of fixed infrastructure and dynamic topology. A number of opportunistic routing policies have been proposed for DTN such as probabilisti routing, SaW, etc. However, these protocols provide good performance in homogeneous DTN only. Our main interest is in DTMSN which is clustered into social communities with heterogeneous mobility patterns.

We presented the routing protocols designed for DTMSN such as BubbleRaP, SimBet, PeopleRank, etc. They take into account the community features and social-based metrics to improve the performance in different settings. However, they do not address the optimality of the routing policies. We address the optimality problem in heterogeneous DTN presenting a theoretical model in Chapter 2. We also highlighted some works in literature related to modeling and performance prediction of routing protocols in DTN, and buffer management policies.

The relay nodes instead of only forwarding replicated copies can generate the coded packets, that is merging routing with NC in DTN. In recent years, many researchers have investigated the use of NC in DTN (e.g., [24, 29, 30, 93, 94]). We presented intra-session NC and ISNC schemes proposed in the literature. We develop controlled ISNC policy for DTMSN that encompasses both routing and coding controls in Chapter 3. We examine ISNC policy on toy topologies to identify when and where it can be beneficial, and design decentralized ISNC coding criteria in Chapters 4 and 5.

Chapter 2

On Optimality of Routing Policies in Delay-Tolerant Mobile Social Networks

Before addressing NC, we first consider the optimization problem of the routing policy in DTMSN. This chapter aims at identifying the structure of optimal routing policies, given a multi-community network model, and then assess the distance to optimal policy of some utility-based routing policies.

Contributions: Our contributions are threefold:

- From a multi-community network model, based on mean-field approximations leading to a fluid model of the dissemination process, we formulate the problem of finding the time-dependent forwarding probabilities between any two communities to maximize the delivery probability by a certain deadline under a given constraint of energy.
- We prove that optimal forwarding policies are per-community threshold policies. We provide a numerical illustration by using a heuristic optimization algorithm.
- We discuss the comparison of the main existing decentralized utility-based routing policies to the optimal policy, so as to assess the distance of these practical routing policies to the optimal in terms of the underlying network social structure.

Two kinds of works are related to ours: the study of what are the best nodes to give the packet to, i.e., that of optimal routing policies, and the design of decentralized utility-based routing policies relying on a smart choice of the utility criterion.

The first set of works include [59], [66] and [96]. In [66], Picu and Spyropoulos consider multicast traffic and identify the best relays to carry the L copies of a packet so as to minimize the maximum time for a destination to retrieve a copy (L is an upper-bound on the number of copies allowed to spread in the network). To do so, they assume that the available knowledge (centralized at an oracle) is only the degree of each node, i.e., the number of different nodes a node meets within a time window. To go from the set of nodes' degrees to the set of probability of contact between any two nodes, they build on the assumption that the probability of meeting between nodes i and j is proportional to both degrees of i and j, coming from the configuration model [66]. Doing so, they come up with the result that the L best relays to carry the L copies are those with the highest degrees. Then, in [96], the same authors present a Markov Chain Monte Carlo (MCMC) algorithm as a distributed solution for online placement to the above defined relays. Our work differs from [66] and [96] in that we consider unicast traffic, and we do not make such assumption on the network connectivity but only assume that the nodes are clustered into communities into which the nodes have the same mobility features, following the model of [53]. Some communities may be "hub" communities, meaning that they can often act as relays between other communities. This allows us to define, in a centralized manner for theoretical purpose, not only what are the communities that must receive the L copies of a packet and how many nodes must be infected in each community, but also what are the paths these copies must follow. These parameters are hence dependent on the communities of the source and destination nodes of the unicast session. In [59], Altman *et al.* consider a homogeneous network defined by the number of nodes and the mean inter-meeting time between any pair of nodes. For probabilistic forwarding in two-hop and epidemic routing, they formulate the optimal control problem, based on a fluid model of the system's dynamics, to minimize the delay under some energy constraint. In particular, they show that the time-dependent problem is optimized by threshold type policies. We extend the work of [59] to identify the structure of optimal routing policies that account for the social features of real-world scenario, i.e., a multi-community environment with heterogeneous mobility.

On the other hand, a number of routing policies have been proposed for DTN to improve the trade-off between performance and energy (or memory) consumption by accounting for the social features of PSN. Their principle is not to spend the allowed number of transmissions with the first met nodes (in a greedy manner), but instead to smartly choose the relays to give the copies to. Some examples of such routing policies are MaxProp [50], BubbleRap[23], PeopleRank [64] and SimBet [4]. Other similar utilities were investigated in [18]: Last-Seen-First (LSF), Most-Mobile-First (MMF) and Most-Social-First (MSF). The same multi-community model as ours is considered, and a fluid model of the network dynamics is used to prove that the utility-based replication (MMF) achieves a lower delivery delay than a greedy-based replication. However, they do not investigate the optimality of forwarding policies based on such model, as we do in our work. As well, Bulut *et al.* [22] studied the effects on the performance of multi-copy based two-hop routing algorithm under the same model. However, they limit the analysis to only two communities. We formulate and address the problem in a more general case.

2.1 Network model

We use the heterogeneous mobility model considered in [18, 22]: the network is made of N mobile nodes divided into M communities (possibly centered around home-points) as shown in

Figure B.1. Our focus is on the network which is made of people carrying portable devices such as Pocket Switched Networks (PSN). In such networks, people are bound together by different social relationships such as friendship, trade, status, etc. and they divide forming different group or communities such that people inside the same community meet more often than those between different communities.

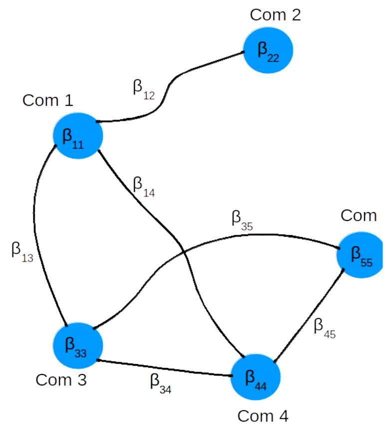

FIGURE 2.1: Multi-community network model.

In our network model we consider the number of nodes in community i is N_i, and we assume that a node pertains to only one community. We denote the total number of nodes by $N = \sum_{i=1}^{M} N_i$, and consider the node partition vector $\mathbf{N} = (N_1, \ldots, N_M)$. The time between contacts of any pair of nodes of communities i and j is exponentially distributed [65, 97] with certain mean and the accuracy of this model has been discussed in [98] and shown for a number of mobility models (Random Walker, Random Direction, Random Waypoint). We assume the mean value is $1/\beta ij$, where β_{ij} is the inter-meeting intensity defined as the mean number of meetings per time unit between a given node of community i and a given node of community j and β_{ii} is the intra-meeting intensity between nodes of community i, and we assume that $\beta_{ii} > \beta_{ij}$, for $i \neq j$, for all $i, j \in \{1, \ldots, M\}$, i.e. the nodes meet more often when they belong to the same community than when they belong to different communities. Let β be the matrix storing the $\{\beta_{ij}\}_{i,j=1}^{M}$. The scenario under study is a source node S of community c_s that wants to send a message (or packet) to a destination node of community c_d. We consider the multi-packet case later in our work. We consider a unicast session without background traffic. Let τ be the delivery deadline until which the message is relevant to the destination. For the sake of lighter notation, a node of community i is denoted as i-node.

Let $\hat{X}_i^{(\mathbf{N})}(t)$ be the fraction of i-nodes (over N) that have a copy of the message at time t, we call such nodes as the infected nodes. Let $\hat{X}^{(\mathbf{N})}(t) = \sum_{i=1}^{M} \hat{X}_i^{(\mathbf{N})}(t)$. Let $\mathbf{u}(t)$ be the policy controlling the message spreading: $u_{ij}(t)$ is the probability that a j-node gives the packet to a i-node when they meet.

Mean-field approximations: Mean field approximations are used to analyze the limit behavior of systems made of N interacting objects as N tends to infinity which has been addressed in number of articles such as [67, 99, 100]. In the limit of large number of objects, it has been found that the system converges, in mean-square, to a deterministic ordinary differential equation (ODE) which is referred to as fluid limit or fluid model.

We build on [67, 99] and consider time t sampled over the discrete domain, i.e., $t \in \mathbb{N}$. The expected change in the number of infected i-nodes in one time slot is defined by

$$f_i^{(\mathbf{N})}(\mathbf{m}) = E\left[\hat{X}_i^{(\mathbf{N})}(t+1) - \hat{X}_i^{(\mathbf{N})}(t)) \mid \hat{\mathbf{X}}^{(\mathbf{N})}(t) = \mathbf{m}\right] .$$

In our model, we have $f_i^{(\mathbf{N})}(\mathbf{m}) = \sum_{j=1}^{M} u_{ij}(t)\beta_{ij}N_j m_j(\frac{N_i}{N} - m_i)$. When N tends to infinity, as we consider a sparse network where the density remains constant when the total number of nodes increases (the ratios N_i/N, for all $i = 1, \ldots, M$, keep constant), $\lambda_{ij} = \beta_{ij}N_j$ remains constant in N. Therefore, for all $i = 1, \ldots, M$:

$$\lim_{N\infty} f_i^{(\mathbf{N})}(\mathbf{m}) = \sum_{j=1}^{M} u_{ij}(t)\lambda_{ij}m_j(\frac{N_i}{N} - m_i)$$

that is independent of N. Then Theorem 3.1 of [99] ensures that the process $\hat{\mathbf{X}}^{(\mathbf{N})}(t)$ converges to a deterministic process $\mathbf{X}(t) = (X_1(t), \ldots, X_M(t))$ which is the solution of the Ordinary Differential Equation (ODE):

$$\forall i = 1, \ldots, M, \quad \frac{dX_i(t)}{dt} = \sum_{j=1}^{M} u_{ij}(t)\lambda_{ij}X_j(t)(\frac{N_i}{N} - X_i(t)),$$

where $X_{c_s}(0) = z$ and $X_i(0) = 0$ for $i \neq c_s$. The above equation is referred to as the fluid model (or mean-field limit) of the dissemination process.

Let D_{c_s,c_d} be the random variable describing the time of delivery. As well, we can derive the fluid limit $P_{c_s,c_d}(\tau)$ of the cumulative distribution function (CDF) of the delay (we assume that the probability that a node gives the packet to the destination node upon meeting is 1): $\frac{dP_{c_s,c_d}(t)}{dt} = \sum_{j=1}^{M} \lambda_{c_d j}X_j(1 - P_{c_s,c_d}(t))$, whereby:

$$P_{c_s,c_d}(\tau) = 1 - \exp^{-\int_0^\tau \sum_{j=1}^{M} \lambda_{c_d j}X_j(t)dt} . \tag{2.1}$$

2.2 Optimization results

In this section, we first express the problem of optimizing the routing policy that is $\mathbf{u}(t) = \{u_{ij}(t)\}_{i,j=1}^{M}$ subjected to an energy constraint, and secondly, we derive the structure of the optimal routing policies for a given optimization problem. For the sake of clarity, let $X_i(t)$ turn to be the number of i-nodes infected by the message, for all $i = 1, \ldots, M$. We consider that the energy consumed by the whole network from time 0 up to τ is proportional to the total number of transmissions that occurred within this time interval. As we do not consider any buffer cleaning mechanism, the energy is hence proportional to $X(\tau) - X(0)$. Let $\epsilon(\tau)$ be defined as this total number of transmissions: $\epsilon(\tau) = X(\tau) = \sum_{j=1}^{M} X_j(\tau)$. Enforcing an energy constraint therefore consists in finding a policy $\mathbf{u}(t)$ such that $\epsilon(\tau) \leq E$, where E is given by the problem.

2.2.1 The optimization problem with an energy constraint

We consider the following constrained optimization problem (CP): [Find $\mathbf{u}(t)$ that maximizes $P_{c_s,c_d}(\tau)$ subject to $\epsilon(\tau) \leq E$].

The problem (CP) is equivalent to maximize $J_{c_s,c_d}(\tau, \mathbf{u}(t)) = \int_0^\tau \sum_{j=1}^{M} \beta_{c_d j} X_j(t) dt$ as we can see from eq. (2.1). Expressing $J_{c_s,c_d}(\tau, \mathbf{u}(t))$ as $J_{c_s,c_d}(\tau, \mathbf{u}(t)) = \sum_{j=1}^{M} \beta_{c_d j} \int_0^\tau X_j(t) dt$, we can see that problem (CP) is a linear optimization problem in the $\int_0^\tau X_j(t)dt$, for $j = 1, \ldots, M$, but a non-linear optimization problem in $\mathbf{u}(t)$.

2.2.2 The structure of optimal routing policies

Although the optimization problem (CP) is non-linear, we are able to identify the subset of policies the optimal policies belong to. We now present the way to the main result of our paper [101], that is the per-community threshold structure of optimal policies.

Definition 2.2.1. ((Condition (C)) A policy $\mathbf{u}(t)$ verifies condition (C) if and only if there exists a couple of indices (I, j) with $0 < u_{Ij}(t) < 1$ for some non-empty interval $[a, b] \subset [0, \tau]$. Let $C(\mathbf{u}(t)) = I$ denote the former I index.

Definition 2.2.2. Consider a policy $\mathbf{u}(t)$ verifying condition (C) and $X(\tau) \leq E$. We define a threshold policy $\bar{\mathbf{u}}(t)$ obtained from $\mathbf{u}(t)$ by the following procedure:

- Initialization: $I = C(\mathbf{u}(t))$
- Recursion: Do$\{$
  $\bar{\mathbf{u}}(t)$ =output of atomic step with input $(\mathbf{u}(t), I)$
  $I = C(\bar{\mathbf{u}}(t))$
  $\mathbf{u}(t) = \bar{\mathbf{u}}(t)$
  $\}$while($\bar{\mathbf{u}}(t)$ satisfies condition (C))
- Atomic step: Let $\mathbf{X}(t)$ be the state process under policy $\mathbf{u}(t)$, and $\bar{\mathbf{X}}(t)$ that under policy $\bar{\mathbf{u}}(t)$.
  We first take, for all $i, j = 1, \ldots, M$:

$$\bar{u}_{ij}(t) = \begin{cases} u_{ij}(t) & \text{, if } i \neq I \\ 1 & \text{, if } i = I \text{ and } t \leq t_I \\ 0 & \text{, if } i = I \text{ and } t > t_I \end{cases}$$

where t_I is such that $\bar{X}_I(\tau) = \bar{X}_I(t_I) = X_I(\tau)$. Then appropriately threshold all $\bar{u}_{ij}(t)$ for $i \neq I$:

$$\bar{u}_{ij}(t) = \begin{cases} u_{ij}(t) & \text{, if } i \neq I \text{ and } t \leq t_i \\ 0 & \text{, if } i \neq I \text{ and } t > t_i \end{cases}$$

where t_i is such that $\bar{X}_i(\tau) = \bar{X}_i(t_i) = X_i(\tau)$.

Lemma 2.2.1. *Let the success probability for policy $u_{ij}(t)$ be $P_s(\tau)$ and that for policy $\bar{u}_{ij}(t)$ be $P'_s(\tau)$. Then: (i) $\bar{u}_{ij}(t)$ satisfies the energy constraint $\bar{X}(\tau) \leq \tau$, and (ii) $P'_s(\tau) > P_s(\tau)$.*

Proof: (i) By construction of the improvement procedure of definition (2.2.2). (ii) By construction, each of the atomic steps generates $\bar{X}_i(t) \geq X_i(t)$ and $\bar{X}_i(\tau) = X_i(\tau)$ for all $i = 1, \dots, M$, and $\bar{X}_I(t) = X_I(t)$ for $t \in [a, b]$ (Def. 2.2.2, whereby the result $P'_s(\tau) > P_s(\tau)$. ⋄

Theorem 2.2.1. *An optimal policy for problem (CP) is a per-community threshold policy, i.e., has the following structure: for all $i, j = 1, \dots, M$, there are thresholds $s_i \in [0, \tau]$ for which, for all $j = 1, \dots, M$, $u_{ij}(t) = 1$ for $t \in [0, s_i]$ and $u_{ij}(t) = 0$ for $t > s_i$.*

Proof: Let $\mathbf{u}(t)$ be an arbitrary policy which satisfies the energy constraint but is not a threshold policy as defined above. Then, there exists some couple (i, j) and some non-empty interval $[a, b] \subset [0, \tau]$ on which $0 < u_{ij}(t) < 1$. So $\mathbf{u}(t)$ can be strictly improved according to Lemma 2.2.1. Hence, $\mathbf{u}(t)$ is not optimal. ⋄

This theoretical result means that, given the β and $\mathbf{N}$ parameters, the optimal number of copies to spread in each community is decided by the optimization solution, and the way to spread those copies is the fastest as possible, that is, for each community i, in an epidemic way from any communities j until s_i, i.e., until the number of copies is reached. This is in accordance with the results in the case of homogeneous DTN, where [58, 59] showed that threshold-policies or Spray-and-Wait, are the best in terms of mean delivery delay under an energy constraint. It is worth noticing that the so-called optimal policies, obtained from the optimization of problem (CP), are offline policies.

2.2.3 Example of numerical optimization

Thanks to a heuristic differential evolution algorithm [102], that is partly a hill-climbing method and partly a genetic algorithm, we use the above results on modeling the dissemination process and optimality of per-community threshold policies to find out the best thresholds for a certain communication requirement.

In this section, we analyze the resulting optimal threshold policy on a toy-example in which there are $M = 3$ communities, the number of nodes in each community is $\mathbf{N} = (33, 33, 34)$, the β matrix is given below:

$$\beta = \begin{bmatrix} 0.2 & 0.1 & 0.05 \\ 0.1 & 0.4 & 0.1 \\ 0.05 & 0.1 & 0.3 \end{bmatrix}.$$

We set the community of the source and destination to $c_s = 1$ and $c_d = 3$, and the deadline $\tau = 0.7$ for which $P_s(\tau)$ is the objective to maximize.

E	$h_{i,j}$	TX_{ij}	TX
100	$\begin{bmatrix} 0.695 & 0.682 & 0.673 \\ 0.678 & 0.680 & 0.696 \\ 0.688 & 0.679 & 0.678 \end{bmatrix}$	$\begin{bmatrix} 17 & 5 & 4 \\ 10 & 23 & 9 \\ 4 & 4 & 21 \end{bmatrix}$	97
40	$\begin{bmatrix} 0.127 & 0.252 & 0.501 \\ 0.098 & 0.245 & 0.581 \\ 0.137 & 0.219 & 0.505 \end{bmatrix}$	$\begin{bmatrix} 1 & 2 & 1 \\ 0 & 7 & 6 \\ 0 & 1 & 22 \end{bmatrix}$	40
10	$\begin{bmatrix} 0.132 & 0.063 & 0.325 \\ 0.119 & 0.080 & 0.296 \\ 0.106 & 0.018 & 0.332 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 7 \end{bmatrix}$	9

TABLE 2.1: Analysis of the optimal threshold policy.

Let h_{ji} for all $i, j = 1, \ldots, M$ be the time threshold up to when a j-node gives a i-node a copy upon meeting, and stops doing so thereafter. Table 2.1 shows the optimized time thresholds h_{ji}, TX_{ij} which is the total number of transmissions from community i to community j by time τ, and the total number of transmissions to be compared with E. We have $TX_{ij} = \int_0^\tau \beta_{i,j} u_{i,j}(t) X_i(t)(N_j - X_j(t))dt$ and $TX = \sum_{i,j=1}^{M} TX_{ij}$. Let us first comment the h_{ji} values. Theorem 2.2.1 states that the h_{ji}, for given i and all $j = 1, \ldots, M$, must be equal (to the per-community threshold s_i). In the optimization procedure, the h_{ji} are the output shown in Table 2.1. They can be set independently, and they appear to be almost constant per column, as predicted by Theorem 2.2.1. Only the second column for $E = 10$ exhibits significant differences, but they do not impact the number of transmissions received in the second community, that keeps 0. Let us now comment the TX_{ij} values, that allow easier interpretation of the thresholds by making appear the sharing of the energy budget across the communities. When the maximum number of copies is highly constrained ($E = 10$), it is better for the source to give a copy to community c_d, and to let the allowed number of copies be fully allocated to spreading inside c_d. However, when the allowed number of copies E increases to 40, then some spare copies, additionally to that in the destination, are worthy spreading in community 2 because the β matrix shows that community 2 has a higher meeting rate with $c_d = 3$ than $c_s = 1$ has.

2.3 Distance from optimality of decentralized utility-based policies

After having theoretically identified the structure of (offline) optimal policies and derived the numerical optimization, now we compare the existing online utility-based policies to the optimal ones, so as to analyze in what cases we can expect these solutions will perform relatively close to, or far from, the optimal policy.

Main classes of utility-based policies have been summarized in [18]. Each node $i = 1, \ldots, N$ maintains a utility function $U_i(j)$ for each other node j. If node i, carrying a copy of the packet destined to node d, has $r > 1$ forwarding tokens and encounters node j with no copy, then i decides to give a copy to j based on the following rules: (i) if $U_j(d) > U_i(d)$ (R1) or (ii) if $U_j(d) > U_{th}$ for some threshold value U_{th} (R2). For example the so-called *Last-Seen-First* (LSF) policy is such that $U_i(j) = \frac{1}{1+\tau_i(j)}$ where $\tau_i(j)$ is the time elapsed since the nodes i and j last encountered each other.

For the purpose of comparison of utility-based policies to the optimal policies devised in Section 2.2, consider the multi-community network model of Section 2.1. Let c_i and c_j be the communities of nodes i and j, respectively. Let us consider the LSF policy. The inter-meeting time between i and j is exponentially distributed with mean $\frac{1}{\beta_{c_i c_j}}$. The expectation of $U_i(j)$ can be computed rigorously and is equal to $E[U_i(j)] = \beta_{c_i c_j} e^{\beta_{c_i c_j}} \Gamma(0, \beta_{c_i c_j})$, where $\Gamma(0, \beta_{c_i c_j})$ is the upper incomplete Gamma function. Consider the approximation of $E[U_i(j)]$ by $\beta_{c_i c_j}$, valid for low $\beta_{c_i c_j}$.

Considering rule (R1), the fluid model for optimal threshold policy and utility-based policy are as follows:

$$\frac{dX_k(t)}{dt} = (N_k - X_k(t))u_k(t) \sum_{l=1}^{M} \beta_{kl} X_l(t)$$

for optimal threshold policy where, $u_{ij}(t)$ turns to be independent of i according to Theorem 2.2.1, and

$$\frac{dX_k(t)}{dt} = (N_k - X_k(t)) \sum_{l:\beta_{lc_d} > \beta_{kc_d}} \beta_{kl} X_l(t), \text{ until } X(t) = E$$

for utility-based policy with rule (R1).

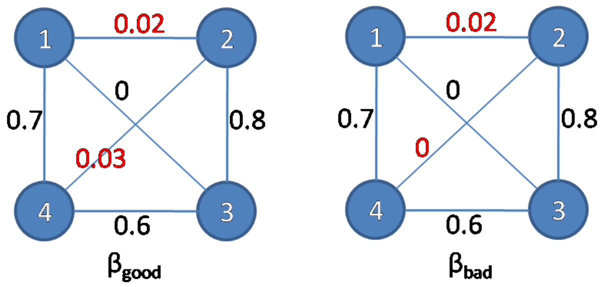

FIGURE 2.2: Graphs of communities connections corresponding to two β matrices.

Figure 2.3 shows the CDF of the delivery delay both for LSF and the optimal policies, for two β matrices, with $M = 4$ communities, $N/4 = 25$ nodes per community, $c_s = 1$ and $c_d = 2$. The simulations are done by averaging over 200 runs for each point, and the contact traces for each β matrix have been generated synthetically based on exponentially distributed inter-meeting times. The confidence intervals are not plotted for the sake of visibility, but the differences are statistically significant. As expected, the performance of LSF is worse than that of the optimal policy, as the latter assumes full knowledge of the network topology, whereas the former learns the graph topology from the encounters. The question that arises is therefore whether the gap between LSF and optimal is only due to the absence of assumption on the network topology for LSF, or also to the very definition of the utility done in LSF. To answer this question, Figure 2.3 also shows the performance of "LSF steady" which denotes the LSF policy where the utility $U_i(j)$ is set to β_{ij} from the beginning, so as to lift the impact of utility convergence on the difference between LSF and optimal. We observe from Figure 2.3 that such impact either fully explains the difference (for β_{Good}) or only partly (for β_{Bad}). Let us analyze why.

In utility-based routing, the packet can be disseminated only from source community c_s to successive communities $i_1, i_2, \dots$ with $\beta_{sd} < \beta_{i_1 d} < \beta_{i_2 d}$ (in a so-called gradient-based manner) until packet reaches destination. Doing so, some low-delay paths from the source to the destination can be missed by the LSF policy. For example, let us consider the two toy-examples in Figure 2.2 with $c_s = 1$ and $c_d = 2$. The performance is shown in Figure 2.3. The LSF policy is expected to perform well on $\beta_{\mathbf{Good}}$ and bad on $\beta_{\mathbf{Bad}}$, as in the latter case, LSF will not allow the packet to get to community 4, and hence to travel along the low-delay path $1-4-3-2$. However, an optimal threshold policy $\mathbf{u}(t)$ can be such that a i-node can give a copy to a j-node even though $\beta_{jc_d} < \beta_{ic_d}$. The optimal policy is hence able to get the best path for any matrix β.

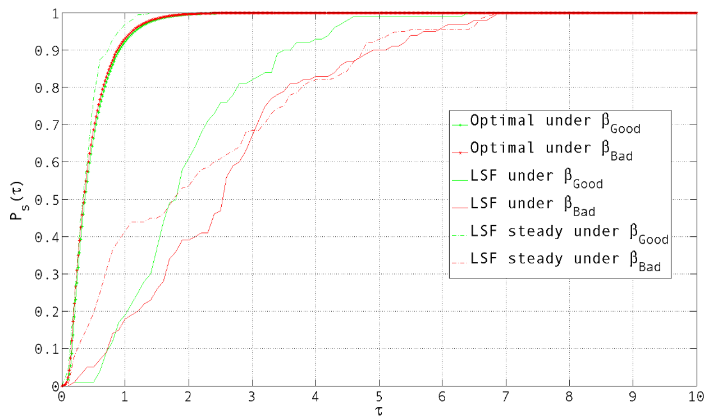

FIGURE 2.3: CDF of the delivery delay, under $L = E = 10$.

Community	1	2	3	4
β_{Good}	1.74	2.65	1.61	0.640
β_{Bad}	2.03	3.48	1.19	0

TABLE 2.2: Mean number of transmissions in each community by the LSF policy, under $E = 10$.

Utility-based policy based on order-2 neighbors can perform better than LSF on network configurations such as β_{bad}. That is the case of Prophet [19], that uses history of encounters, and the utility value is updated with the transitivity property, e.g. if node A frequently encounters node B, and node B frequently encounters node C, then node C probably is a good node to forward messages destined for node A. This feature brings Prophet closer to the optimal routing than LSF is, at the expense of complexity. For the other policies studied in [18], namely *Most Mobile First* (MMF) and *Most Social First* (MSF), the corresponding definitions of the utilities can also be expressed with the network parameterization into communities and β matrix. In MaxProp [50], each node i keeps the record of the meeting rate with node j, f_j^i, for all $i, j = 1, \ldots, M$ (that is similar to β_{ij} estimations) of each path towards destination and selects the best path, in terms of delivery likelihood, to forward the packet. It therefore tries to approach the optimal solution. However, these routing algorithms work with a single copy. In BubbleRap [23], each node keeps the record of its community label, its local rank and its global rank. Local rank and global rank are based on betweenness centralities estimated by the number of different nodes met within a time window inside and outside the node's community, respectively. Such parameters imply a different model than the multi-community model presented in Section 2.1, which does not allow to distinguish between nodes inside the same community. Therefore the global rank can be expressed thanks to the multi-community model, but the local rank will be the same for all the same community nodes. The multi-community model does not encompass such possibility for the sake of simplicity, so as to be able to derive theoretical results on optimal routing policies as done in Section 2.2, then to compare with existing online utility-based policies.

2.4 Conclusion

In this chapter, we have addressed the problem of optimizing routing policies in mobile social DTN. Thanks to a mean-field approximation of the spreading process, we have formulated the problem of finding the time-dependent forwarding probabilities between any two communities to maximize the delivery probability by a certain deadline under given energy constraints. We have proven theoretically that the optimal forwarding policies are per-community threshold policies, providing analysis of this result on a numerical example and we have discussed comparison of the existing decentralized online utility-based polices to assess the distance of these practical routing policies to the optimal routing policies in terms of the underlying network social structure.

Chapter 3

On Control of Inter-session Network Coding in Delay-Tolerant Mobile Social Networks

To improve the benefit of disseminating redundant packets, coded redundant packets can be generated by the relays instead of or additionally to replicated copies, that is performing Network Coding (NC) in DTN. NC is a networking paradigm that is a generalization of routing [24, 25]. Specifically, random NC [26] has attracted an increasing interest for DTN [27, 28]. The benefits are increase in throughput, as well as adaptability to network topology changes. There are two types of NC: in intra-session NC, only the packets belonging to the same session are coded together (i.e., combined), while in inter-session NC (ISNC) packets pertaining to different sessions can be combined. ISNC is necessary to achieve optimal throughput in general (see [29] and references therein) but represents a difficult optimization problem, in particular for DTN.

Contributions: Our contributions are threefold:

- We design a parameterized pairwise ISNC control policy for heterogeneous DTN, that encompasses both routing and coding controls with an energy constraint. We present the resulting dissemination protocol.
- We derive its performance modeling thanks to a mean-field approximation leading to a fluid model of the dissemination process. We validate the model by numerical experiments.
- We discuss the optimization of ISNC control policy benefits in social DTN, and show that the fluid model can be used to devise such optimal policy that jointly exploits the nodes' social acquaintances and the ISNC. By showing numerical gains, we illustrate the relevance of our ISNC control policy that is based on the coarse-grained community structure rather on individual nodes.

This chapter does not aim at presenting a self-contained decentralized ISNC protocol that can be confronted with existing routing policies in DTN. It aims at devising, modeling and proving the benefit of a centralized social-aware (community-based) pairwise ISNC policy.

For homogeneous DTN, several works have considered intra-session NC. Lin *et al.* in [28] investigated the use of intra-session NC using the SaW algorithm and analyzed the performance in terms of the bandwidth of contacts, the energy constraint and the buffer size. However, neither background traffic nor other running session are assumed beside the unicast session of interest. In [103], we have lifted this assumption and modeled information dissemination of several concurrent unicast sessions in homogeneous DTN, when ISNC and SaW routing are employed. Now, we extend this work not only to heterogeneous DTN to predict the performance of contending unicast sessions, either inter-session network coded or not, but also model the control of ISNC decisions based on the social features of the DTN.

In Appendix A, we present the information dissemination model for several contending unicast sessions in heterogeneous DTN, when ISNC and SaW routing are employed.

3.1 Network model

We consider the network model presented in Section 2.1 which is made of N mobile nodes divided into C communities such that $N = \sum_{i=1}^{C} N_i$ where N_i is the number of nodes in community i, and we assume that a node pertains to only one community. In Table 3.1, we present the main parameters' notation used for this model.

In this model, the time between two consecutive contacts is exponentially distributed with a certain mean. The accuracy of this model has been discussed in [98] and shown for a number of mobility models (Random Walker, Random Direction, Random Waypoint). The inter-meeting intensity β_{ij} is defined as the inverse of this mean and represents the mean number of contacts per time unit between a given node of community i and another given node of community j. We assume that $\beta_{ii} > \beta_{ij}$, for $i \neq j$, for all $i, j \in \{1, \ldots, C\}$. The matrix β storing the $\{\beta_{ij}\}_{i,j=1}^{C}$ defines the inter-meeting intensity of any pair of nodes.

Two sources S_1 and S_2 of communities s_1 and s_2, respectively, want to send a file each to their respective destinations D_1 and D_2 in communities d_1 and d_2, respectively. We assume that the file to be transferred needs to be split into K packets: this occurs owing to the finite duration of contacts among mobile nodes or when the file is large with respect to the buffering capabilities of the nodes. The message is considered to be well received if and only if all the K packets of the source are recovered at the destination. We do not assume any feedback.

We assume that the bandwidth, defined as in [28] as the number of packets that can be exchanged during a contact in each direction (thereby accounting both for the rate and the contact duration), is stochastic and follows any known distribution of mean Bw. The buffer size is assumed to be any known integer, denoted by B, equal for all the nodes in the network. Note that

Symbol	*Meaning*
Network settings	
N	total number of nodes excluding the sources and the destinations
C	number of node communities
N_i	number of nodes in community i
β_{ij}	inter-meeting intensity of a node in community i with a node in community j
Bw	bandwidth: mean number of packets that can be exchanged during a contact in each direction
Communication settings	
S_1, S_2	source node of session 1, 2
D_1, D_2	destination node of session 1, 2
K_1, K_2	number of information packets of session 1, 2
K'_1, K'_2	maximum number of packets that can be released by S_1, S_2
M, Q	maximum number of copies of an index released by S_1, S_2
S_{11}, S_{22}	set of indices associated to pure payloads sent out by source S_1, S_2
S_{31}, S_{32}	set of indices emitted by S_1 (resp. S_2) associated to a mixed payload, that a combination from pure payloads of S_1 and S_2
X_{ic}, Y_{ic}	number of nodes in community c that carry i indices in S_{11} (resp. S_{22})
Z^1_{ic}, Z^2_{ic}	number of nodes in community c that carry i indices in S_{31} (resp. S_{32})
$\tilde{X}_{Ic}, \tilde{Y}_{Ic}$	number of nodes in community c that carry index I of S_{11} (resp. S_{22})
$\tilde{Z}^1_{Ic}$, $\tilde{Z}^2_{Ic}$	number of nodes in community c that carry index I of S_{31} (resp. S_{32})
$u^{11}_{ce}(t)$, $u^{22}_{ce}(t)$	probability that a node of community c gives a packet with an index in S_{11} (resp. S_{22}) to a node of community e upon meeting at time t, provided that it is possible to copy such an index (it exists at the sending node and its spray-counter is below M (resp. Q))
$u^{31}_{ce}(t)$, $u^{32}_{ce}(t)$	probability that a node of community c gives a packet with an index in S_{31} (resp. S_{32}) to a node of community e upon meeting at time t, provided that it is possible to copy such an index (it exists at the sending node and its spray-counter is below M (resp. Q))
l	$l = \sum_{i=11,22,31,32} l_i$ for a $(c, \mathbf{1})$-node

TABLE 3.1: Main notation used for ISNC control policy.

these assumptions are not necessary for the dissemination protocol presented in Algorithm 1-2 to work.

3.1.1 Inter-session NC

Let us now describe simply how a node having two packets of two different sessions, performs ISNC to forge a new coded packet to be sent out. The process described hereafter is depicted in Figure B.2. All nodes can identify the session number of each packet. Consider that packets P_1 and P_2, belonging to sessions S_1 and S_2, are Random Linear Combinations (RLC) of the K_1 and K_2 original information packets, respectively. The header coefficients of P_1 and

P_2 are hence K_1-long and K_2-long, while payloads are L_1 and L_2-long, where L_1 and L_2 are the maximum size of packets of S_1 and S_2, respectively. The packet resulting from an RLC of P_1 and P_2 has header coefficients $(K_1 + K_2)$-long, and payload $\max(L_1, L_2)$-long. The original K_1 packets of session 1 can be recovered if and only if the matrix made of the coding coefficients can undergo a Gauss-Jordan elimination resulting in only elements of the K_1-size identity matrix over the K_1 columns assigned to session 1 and for the corresponding rows, all the other columns are zero. Thereafter, the number of received Degree of Freedom (DoF) of session 1 is the number of identity elements over these K_1 columns.

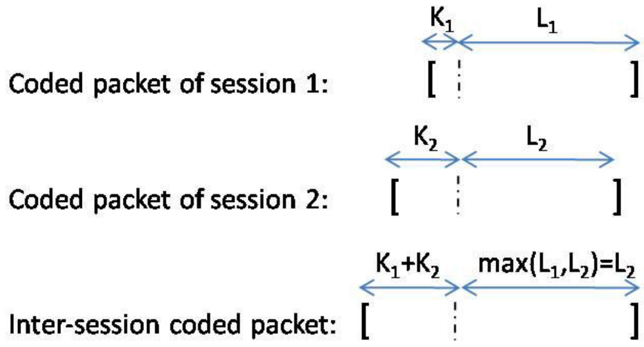

FIGURE 3.1: Generation of an inter-session network coded packet

3.1.2 The parameterized pairwise ISNC control policy

We build on the buffer structure for intra-session NC framework employed with SaW [82], that we modify to account for ISNC of two sessions. Note that binary SaW is considered for implementation, but any other spraying with a token mechanism allowing to control the dissemination, such as that of [93], can be considered for implementation and modeling. At a relay node buffer, a packet is associated with 3 fields: index, spray-counter, payload. Both sources S_1 and S_2 spread $K_1' \geq K_1$ (resp. $K_2' \geq K_2$) RLCs over the K_1 (resp. K_2) information packets. Each RLC sent out by S_1 (resp. S_2) is associated with an index in $S_{11} = \{1, \ldots, K_1'\}$ (resp. $S_{22} = \{K_1' + 1, \ldots, K_1' + K_2'\}$) and with a counter M (resp. Q). A node is said to be a $(c, \mathbf{l})$-node, if it belongs to community c and has in its buffer $\mathbf{l} = (l_{11}, l_{22}, l_{31}, l_{32})$ indices of S_{11}, S_{22}, S_{31} and S_{32}, respectively. Also, we take $l = \sum_i l_i$. In order to control the ISNC decisions, we introduce the policy $\mathbf{u}_{ab}(t) = (u_{ab}^{11}(t), u_{ab}^{31}(t), u_{ab}^{22}(t), u_{ab}^{32}(t))$, for all a and b in $\{1, \ldots, C\}$. It corresponds to the probability to draw, at each transmission opportunity, an index of each kind for sending it from a node of community a to a node of community b. As aforementioned, the number of transmission opportunities at each contact has an arbitrary distribution with mean Bw. The components of $\mathbf{u}_{ab}$ are defined in Table 3.1. We constrain $\sum_i u_{ab}^i \leq 1$ to allow for a node not to spread as many packets as possible if it is better to keep some transmissions for later meetings. Algorithm 1-2 describes the dissemination protocol we consider with ISNC. The cases where the sources or destinations are met can be found in Algorithm 1.

Definition 3.1.1. An index I is said to be part of S_{11} or S_{31} (resp. S_{22} or S_{32}) if either I or $I-(K_1'+K_2')$ are in $[1,K_1']$ (resp. I or $I-(K_1'+K_2')$ are in $[K_1'+1,K_1'+K_2']$).

For this protocol, we consider that no packet exchange is possible once the buffer of the receiving node is full, but note that it is straightforward to adapt the framework to any kind of buffer management policy (including possible exchange when the buffers are full). As well, D_1 (resp. D_2) is allowed to make a meeting relay drop only packets of S_{11} (resp. S_{22}) it already got. It is worth noting also that the binary split of the counter in between the replicated packet and that remaining in the sending node is not a constraint of the framework, and can be changed for any other counter sharing, such as that based on some utility and presented in [4].

3.2 Modeling dissemination

The goal is to predict the evolution over time of the different numbers of nodes describing the dissemination process and defined in Table 3.1. To do so, we resort to a mean-field approximation that allows to predict the mean behavior of a system, modeled as a Markov chain, made of a growing number of interacting objects. We do not provide a formal proof that such a mean-field approximation hold in our work, but rather only a sketch of the proof, and assess numerically the accuracy of the model in Section 3.2.3. By Theorem 3.1 of [99], the quantities X_{ic}, Y_{ic}, Z^1_{ic}, Z^2_{ic}, $\tilde{X}_{Ic}$, $\tilde{Y}_{Ic}$, $\tilde{Z}^1_{Ic}$, $\tilde{Z}^2_{Ic}$ defined in Table 3.1, that are random processes depending on the random mobility process, can be approximated by deterministic processes that are the solutions of certain coupled Ordinary Differential Equations (ODEs). These ODEs stem from the limit of the system dynamics (the “drift”) for large N and are called the fluid model. Let us now present the ODEs of the fluid model.

Below we present the main components of the model and in particular we highlight how the control of ISNC over the time and the communities is taken into account. As in the protocol description above, the notation in what remains corresponds to considering a node A that may send packets to a node B that is it meeting.

3.2.1 Evolution of the buffer occupancy distribution

The ODEs for X_{ic} and Z^1_{ic} write as:

$$dX_{ic}(t) = \beta_{s_1c}N_c\sum_{\mathbf{l}^B}P_j(c,\mathbf{l}^B)P_{gs11}(i-l^B_{11},\mathbf{l}^B,\mathbf{u}_{s_1c})\dots$$

$$+\beta_{cd_1}N_c\sum_{\mathbf{l}^B}P_j(c,\mathbf{l}^B)P_{ls11}(l^B_{11}-i,\mathbf{l}^B,\mathbf{l}^{D_1},\mathbf{u}_{cd_1})\dots$$

$$+N_c\sum_{e=1}^{C}\sum_{\mathbf{l}^A,\mathbf{l}^B}\beta_{ec}N_eP_j(c,\mathbf{l}^A)P_j(c,\mathbf{l}^B)P_{grs11}(i-l^B_{11},\mathbf{l}^A,\mathbf{l}^B,\mathbf{u}_{ec})\dots$$

1: Protocol with ISNC - Part 1

Data: a $(a, \mathbf{l}^A)$-node A (i.e., in community a with $\mathbf{l}^A$), and a node $(b, \mathbf{l}^B)$-node B,
$\mathbf{u}_{ab} = (u_{ab}^{11}, u_{ab}^{22}, u_{ab}^{31}, u_{ab}^{32})$, the number of packet transmission opportunities w
Result: How many and what packets generated by A to be stored at B
Let $Hd(\mathbf{u})$ be the distribution of a discrete random variable Y with 4 values, value i being taken with probability u_{ab}^i.
if $A == S_1$ *and* $B \neq D_1$ *and* $B \neq D_2$ **then**
  Draw q_{11} from a binomial distribution (w, u_{ab}^{11}).
  if *Num_of_RLCs_sent_by_*$S_1 < K_1'$ **then**
    Send $x = \min(q_{11}, K_1' -$ *Num_of_sent_RLCs_by_A*$, B - l^B)$ RLCs with indices from *Num_of_sent_RLCs_by_*$S_1 + 1$ up to *Num_of_sent_RLCs_by_*$S_1 + x$

else if $A == S_2$ *and* $B \neq D_1$ *and* $B \neq D_2$ **then**
  Draw q_{22} from a binomial distribution (w, u_{ab}^{22}).
  if *Num_of_RLCs_sent_by_*$S_2 < K_2'$ **then**
    Send $x = \min(q_{22}, K_2' -$ *Num_of_sent_RLCs_by_A*$, B - l^B)$ RLCs with indices from *Num_of_sent_RLCs_by_*$S_2 + 1$ up to *Num_of_sent_RLCs_by_*$S_2 + x$

else if $B == D_1$ **then**
  Drop the packets with indices of S_{11} already present at D_1. Update l_{11}^A accordingly.
  x=0;
  while $x \leq w$ **do**
    Draw y from $Hd(\mathbf{u})$.
    if y==*11* **then**
      Send 1 packet whose index is in S_{11} to D_1 and drop it from A.

    if y==*22* **then**
      Send 1 packet whose index is in S_{22} to D_1.

    if y==*31* **then**
      Send 1 packet whose index is in S_{31} to D_1. The payload is an RLC of all the packets of A.

    if y==*32* **then**
      Send 1 packet whose index is in S_{32} to D_1. The payload is an RLC of all the packets of A.

    x=x+1;
  end
else if $B == D_2$ **then**
  The same as above, replacing D_1 by D_2 and 11 by 22.

2: Protocol with ISNC - Part 2

/∗ Else $A \neq S_1$ and $A \neq S_2$ and $B \neq D_1$ and $B \neq D_2$ ∗/

else

x=0;

while $x \leq w$ **do**

Draw y from $Hd(\mathbf{u})$.

if y==*11* **then**

(1) Let $\mathbf{n}^{11}$ be the list of indices in S_{11} at A that are neither in S_{11} nor in S_{31} at B (according to def. 3.1.1) and whose counter is strictly greater than 1.

Let $\mathbf{p}$ be the set of packets at A corresponding to these indices.

if $\mathbf{n}^{11}$ *not empty and* $B - l^B \geq 1$ **then**

Send a packet q to B with:

$q.index = p_1.index = n_1^{11}$

$q.counter = \lfloor \frac{p_1.counter}{2} \rfloor$

$q.payload = p_1.payload$

Update

$l_{11}^B = l_{11}^B + 1$

$p_1.counter = \lceil \frac{p_1.counter}{2} \rceil$

Remove n_1^{11} from $\mathbf{n}^{11}$

if y==*22* **then**

Same steps from (1) as above, replacing 11 by 22 and M by Q.

if y==*31* **then**

(2) Let $\mathbf{n}^{31} = [v, \mathbf{n}^{11}]$, where $\mathbf{n}^{11}$ stems from step (1) and v is the list of indices in S_{31} at A that are neither in S_{11} nor in S_{31} at B (according to def. 3.1.1) and whose counter is strictly greater than 1.

Let $\mathbf{p}$ be the set of packets at A corresponding to these indices.

if $\mathbf{n}^{31}$ *not empty and* $B - l^B \geq 1$ **then**

Send a packet q to B with:

$q.index = p_1.index = n_1^{31}$

$q.counter = \lfloor \frac{p_1.counter}{2} \rfloor$

$q.payload =$ RLC(all packets at A)

Update

$l_{31}^B = l_{31}^B + 1$

$p_1.counter = \lceil \frac{p_1.counter}{2} \rceil$

if y==*32* **then**

Same steps from (2) as above, replacing 11, 31 and M by 22, 32 and Q.

x=x+1;

end

Go to the beginning of Algorithm 2, exchange A and B and perform again all the steps.

$$-\beta_{cs_1}N_c \sum_{\mathbf{l}^B:l_{11}^B=i,p_{11}>0} P_j(c,\mathbf{l}^B)P_{gs11}(p_{11},\mathbf{l}^B,\mathbf{u}_{s_1c})\ldots$$

$$-\beta_{cd_1}N_c \sum_{\mathbf{l}^B:l_{11}^B=i,p_{11}>0} P_j(c,\mathbf{l}^B)P_{ls11}(p_{11},\mathbf{l}^B,\mathbf{l}^{D_1},\mathbf{u}_{cd_1})\ldots$$

$$-N_c\sum_{e=1}^{C}\sum_{\substack{p_{11}\mathbf{l}^A,\\ \mathbf{l}^B:l_{11}^B=i}} \beta_{ec}N_eP_j(c,\mathbf{l}^A)P_j(c,\mathbf{l}^B)P_{grs11}(p_{11},\mathbf{l}^A,\mathbf{l}^B,\mathbf{u}_{ec})\ .$$

$$dZ_{ic}^1(t) = N_c\sum_{e=1}^{C}\sum_{\mathbf{l}^A,\mathbf{l}^B}\beta_{ec}N_eP_{grs31}(i-l_{31}^B,\mathbf{l}^A,\mathbf{l}^B,\mathbf{u}_{ec})\ldots$$

$$-N_c\sum_{e=1}^{C}\sum_{\substack{p_{31}>0,\mathbf{l}^A,\\ \mathbf{l}^B:l_{31}^B=i}}\beta_{ec}N_eP_{grs31}(p_{31},\mathbf{l}^A,\mathbf{l}^B,\mathbf{u}_{ec})\ .$$

The ODEs for Y_{ic} and Z_{ic}^2 can be deduced from those of X_{ic} and Z_{ic}^1, replacing 1 by 2 everywhere. The components of the above equations are defined as follows:

- $P_j(c,\mathbf{l})$: fraction of relay nodes that are $(c,\mathbf{l})$-nodes. $P_j(c,\mathbf{l})$ is computed such that the following constraints are satisfied: $P_j(c,\mathbf{l}) = 0$ for $l^B > B$, $\sum_{\mathbf{l}} P_j(c,\mathbf{l}) = 1$, $\sum_{\mathbf{l}:l_{11}=i} P_j(c,\mathbf{l}) = \frac{X_{ic}}{N_c}$, $\sum_{\mathbf{l}:l_{22}=i} P_j(c,\mathbf{l}) = \frac{Y_{ic}}{N_c}$, $\sum_{\mathbf{l}:l_{31}=i} P_j(c,\mathbf{l}) = \frac{Z_{ic}^1}{N_c}$ and $\sum_{\mathbf{l}:l_{32}=i} P_j(c,\mathbf{l}) = \frac{Z_{ic}^2}{N_c}$.
- Let $K_{S_1}(t)$ be the number of indices released by S_1 up to time t and P_{sc} be the average number of indices that S_1 gives around time t to community c. Then, $\frac{dK_{S_1}(t)}{dt} = \sum_{c=1}^{C}\beta_{s_1c}N_cP_{sc}$ where, $P_{sc} = \sum_{p_{11}}\sum_{\mathbf{l}^B} p_{11}P_{gs11}(p_{11},\mathbf{l}^B,\mathbf{u}_{s_1c})P_j(c,\mathbf{l}^B)$. The number of indices of S_{11} that D_1 has received until time t is denoted by $R_{11}(t)$; $\frac{dR_{11}(t)}{dt}$ can be expressed from $P_{ls11}(.)$ in the same way as $K_{S_1}(t)$.
- $P_{nic,e,c}\big(n_{11},\mathbf{l}^A,\mathbf{l}^B,K_{S_1}(t),\mathbf{v}(t)\big)$: probability that for a $(e,\mathbf{l}^A)$-node and a $(c,\mathbf{l}^B)$-node, there are n_{11} indices of S_{11} at node A n̲ot i̲n c̲ommon with $S_{11}\bigcup S_{31}$ at node B and whose corresponding spray-counters are still below M, when S_1 has already spread out $K_{S_1}(t)$ indices. The vector $\mathbf{v}_{11}(c,t)$ stores the occurrence probability of each index of S_{11} at time t in community c: $\mathbf{v}_{11}(c,t) = \big(\frac{\tilde{X}_{1e}}{\tilde{X}_c},\ldots,\frac{\tilde{X}_{K_{S_1}(t)c}}{\tilde{X}_c}\big)$ with $\tilde{X}_c = \sum_{I\in S_{11}}\tilde{X}_{Ic}$. Similarly, we define $\mathbf{v}_{31}(c,t)$ from the $\tilde{Z}_{Ic}^1$ and we take $\mathbf{v}(t) = \frac{l_{11}^A}{s}\mathbf{v}_{11}(c,t) + \frac{l_{31}^A}{s}\mathbf{v}_{31}(c,t) + \frac{j}{s}\mathbf{v}_{11}(c,t)$, $s = l_{11}^B + l_{31}^B + j$ and $p_s = \frac{\sum_{I\epsilon E}\tilde{X}_{Ie}(t)}{\sum_{I\epsilon T_c}\tilde{X}_{Ie}(t)}$, with $T_e = \{I \in S_{11} : \tilde{X}_{Ie} > 0\}$ and E be the set of indices of S_{11} that can still spread: $E = \{I \in S_{11} : 0 < \sum_{c=0}^{C}\big(\tilde{X}_{Ic} + \tilde{Z}_{Ic}^1\big) < M\}$. Then $P_{nic,e,c}\big(n_{11},\mathbf{l}^A,\mathbf{l}^B,K_{S_1}(t),\mathbf{v}(t)\big)$ and $P_{nicD,e,d_1}\big(n_{11},\mathbf{l}^A,\mathbf{l}^{D_1},K_{S_1}(t),\mathbf{v}(t)\big)$ (the probability that there are n_{11} indices of S_{11} at node B, not in common with S_{11} at D_1) are given by a combination of the above quantities with the function $S_z(.)$ defined hereafter. If S_e is a set of pairwise different elements from T_e whose cardinality is $|S_e|$, the probability to have exactly z

different elements occurring among a set of $K_{S_1}(t)$ elements, the i^{th} elements having an occurrence probability $\mathbf{v}_i(t)$, is $S_z\big(K_{S_1}(t), z, \mathbf{v}(t)\big) = \sum_{S_e \subset T_e : |S_e| = z} \prod_{i \epsilon S_e} v_i(t) \prod_{i \epsilon T_e \setminus S_e} \big(1 - v_i(t)\big)$. The probability that the number of indices not in common be greater than n_{11} between nodes A and B is defined as:

$$P_{cn,e,c}\big(n_{11}, \mathbf{l}^A, \mathbf{l}^B, K_{S_1}(t), \mathbf{v}(t)\big) = \begin{cases} 1 & \text{, if } n_{11} < l_{11}^A - (l_{11}^B + l_{31}^B) \\ 0 & \text{, elseif } n_{11} > l_{11}^A \\ \frac{\sum_{j=n_{11}}^{l_{11}^A} S_z\big(K_{S_1}(t), l_{11}^B + l_{31}^B + j, \mathbf{v}(t)\big)}{\sum_{j=0}^{l_{11}^A} S_z\big(K_{S_1}(t), l_{11}^B + l_{31}^B + j, \mathbf{v}(t)\big)} & \text{, otherwise} \end{cases}$$

$$P_{nic,e,c}\big(n_{11}, \mathbf{l}^A, \mathbf{l}^B, K_{S_1}(t), \mathbf{v}(t)\big) = \binom{l_{11}^A}{n_{11}} p^{n_{11}} (1-p)^{(l_{11}^A - n_{11})} \dots$$
$$\Big(P_{cn,e,c}\big(n_{11}, \mathbf{l}^A, \mathbf{l}^B, K_{S_1}(t), \mathbf{v}(t)\big) - P_{cn,e,c}\big(n_{11}+1, \mathbf{l}^A, \mathbf{l}^B, K_{S_1}(t), \mathbf{v}(t)\big)\Big) ,$$

where, $\mathbf{v}(t) = \frac{l_{11}^A}{s}\mathbf{v}_{11}(c,t) + \frac{l_{31}^A}{s}\mathbf{v}_{31}(c,t) + \frac{j}{s}\mathbf{v}_{11}(e,t)$, $s = l_{11}^B + l_{31}^B + j$ and $p = \frac{\sum_{I \epsilon E} \tilde{X}_{Ie}(t)}{\sum_{I \epsilon T_e} \tilde{X}_{Ie}(t)}$. When D_1 is the receiving node, the probability that there are n_{11} indices of S_{11} at node B, not in common with S_{11} at D_1 is given by $P_{nicD,e,d_1}\big(n_{11}, \mathbf{l}^A, \mathbf{l}^{D_1}, K_{S_1}(t), \mathbf{v}(t)\big) = \binom{l_{11}^A}{n_{11}}(1-p)^{n_{11}} p^{(l_{11}^A - n_{11})}$ where $p = \frac{\sum_{I \epsilon T_e} P_I^{11}(t)}{|T_e|}$ and $P_I^{11}(t)$ is the probability that index I of S_{11} has been received at D_1 by time t.

In what follows, q_{11} $\big(Q_{11}$ for the random variable (r.v.)$\big)$ denotes the number of draws of S_{11} out of the bandwidth realization, n_{11} (N_{11} for the r.v.) denotes the number of indices of S_{11} that are in node A but not in B, p_{11} denotes the number of indices of S_{11} given by A to B, and s (ζ for the r.v.) denotes the bandwidth realization. Hence we have:

$$Pr(Q_{11} = q_{11}) = \sum_{r \geq q_{11}} Pr(\zeta = r) \binom{s}{q_{11}} (u_{s_1c}^{11})^{q_{11}} (1 - u_{s_1c}^{11})^{r - q_{11}} .$$

$$Pr(N_{11} = n_{11}) = \begin{cases} P_{nicD,c,d_1}\big(n_{11}, \mathbf{l}^B, \mathbf{l}^{D_1}, K_{S_1}(t), \mathbf{v}(t)\big) , \\ \text{if } B = D_1 \\ P_{nic,e,c}\big(n_{11}, \mathbf{l}^A, \mathbf{l}^B, K_{S_1}(t), \mathbf{v}(t)\big) , \\ \text{otherwise} \end{cases}$$

$Pr(\zeta = r)$ is given by the network configuration and can be any (taken as Poisson in the numerical examples below). Similar quantities are defined for S_{22}, S_{31} and S_{32}.

- $P_{gs11}(p_{11}, \mathbf{l}^B, \mathbf{u}_{s_1c})$: probability that S_1 gives p_{11} indices (of S_{11}) to node B.

$$P_{gs11}(p_{11}, \mathbf{l}^B, \mathbf{u}_{s_1c}) = \sum_{q_{11}} Pr(Q_{11} = q_{11}) \Big(p_{11} == \min\big(B - l^B, K_1^{'} - K_{S_1}(t), q_{11}\big)\Big) .$$

- $P_{ls11}(p_{11}, \mathbf{l}^B, \mathbf{l}^{D_1}, \mathbf{u}_{cd_1})$: probability that node B, upon meeting with D_1, drops p_{11} indices of S_{11} that D_1 already has or that B hands over to D_1.

$$P_{ls11}(p_{11}, \mathbf{l}^B, \mathbf{l}^{D_1}, \mathbf{u}_{cd_1}) = \sum_{n_{11}, q_{11}} Pr(Q_{11} = q_{11}) \dots$$

$$Pr(N_{11} = n_{11})\big(p_{11} + n_{11} - l^B_{11} == \min(n_{11}, q_{11})\big) \; .$$

- $P_{grs31}(p_{31}, \mathbf{l}^A, \mathbf{l}^B, \mathbf{u}_{ec})$: probability that node B receives p_{31} indices of S_{31} from node A.

$$P_{grs31}(p_{31}, \mathbf{l}^A, \mathbf{l}^B, \mathbf{u}_{ec}) = (p_{31} \leq B - l^B) \sum_{\substack{s,\\ n_{11}, n_{31}}} F(p_{31}) \dots$$

$$Pr(S = s) Pr(N_{11} = n_{11}) Pr(N_{31} = n_{31}) \dots$$

$$\Big(l^A_{31} > 0 \; or \; \big(l^A_{31} = 0, l^A_{11} > 0 \; and \; (l^A_{22} \; or \; l^A_{32}) > 0\big)\Big) \; ,$$

that represents the condition for generating a ISNC packet (out of mixed S_{31} or unmixed packets in S_{11} and S_{22}). The r.v. S stands for the number of draws that elect a S_{11} or S_{31} index to be sent out, hence $Pr(S = s) = \sum_{r \geq s} Pr(\zeta = r)\binom{r}{s}(u_1)^s(1-u_1)^{r-s}$ with $u_1 = u^{11}_{s_{1c}} + u^{31}_{s_{1c}}$. Let $v_1 = u^{11}_{s_{1c}}/u_1$ and $v_3 = u^{31}_{s_{1c}}/u_1$. Specifically, $F(p_{31})$ captures the coding decision: when a contact occurs, at each transmission opportunity (below denoted by "a draw", the mean number of these being Bw), one of the four types of indices is drawn. If S_{31} is drawn, such an index is either directly one of the S_{31} indices at node A, or is one of the S_{11} if no S_{31} are yet available (the payload being forged by combining S_{11} and S_{22} or S_{32}). Hence, it leaves less S_{11} indices available for the subsequent draws of S_{11}. We have

$$F(p_{31}) = Pr(p_{31} \text{ packets of } S_{31} \text{received in } s \text{ draws}) = (p_{31} \leq n_{31}) f(p_{31}, s) + (p_{31} > n_{31}) \dots$$

$$Pr\Big(n_{31} \text{ of } S_{31} \text{ sent then } p_{31} - n_{31} \text{ sent from the } S_{11} \text{ until } s\Big),$$

with $f(p_{31}, s) = \binom{s}{p_{31}} v_3^{p_{31}} v_1^{s-p_{31}}$, and

$$Pr\Big(n_{31} \text{ of } S_{31} \text{ sent then } p_{31} - n_{31} \text{ sent from the } S_{11} \text{ until } s\Big) =$$

$$\begin{aligned} &\sum_{a=n_{31}}^{s} Pr\Big(\text{all } n_{31} \text{ of } S_{31} \text{ exhausted at draw } a\Big) \dots \\ &\sum_{b=a+p_{31}-n_{31}+1}^{\min(s, n_{31}+n_{11})} Pr\Big(\text{last packet in } S_{31} \text{ received at draw } b\Big) \dots \\ &\qquad Pr\Big(p_{31} - n_{31} \text{drawn in } b - a \text{ draws}\Big) \; , \end{aligned} \tag{3.1}$$

where, $Pr\Big(\text{all } n_{31} \text{ of } S_{31} \text{ exhausted at draw } a\Big) = \binom{a}{n_{31}} v_3^{n_{31}} v_1^{a-n_{31}}$,

$Pr\Big(\text{last packet in } S_{31} \text{ received at draw } b\Big) = v_1^{s-b} + (b - n_{31} == n_{11}) - (b - n_{31} == n_{11}) v_1^{s-b}$, and $Pr\Big(p_{31} - n_{31} \text{drawn in } b - a \text{ draws}\Big) = \binom{b-a}{p_{31}-n_{31}} v_3^{p_{31}-n_{31}} v_1^{b-a-(p_{31}-n_{31})}$.

- $P_{grs11}(p_{11}, \mathbf{l}^A, \mathbf{l}^B, \mathbf{u}_{ec})$: probability that node B gains p_{11} indices of S_{11} from node A.

$$P_{grs11}(p_{11}, \mathbf{l}^A, \mathbf{l}^B, \mathbf{u}_{ec}) = (p_{31} \le B - l^B) \dots$$

$$\begin{cases} \sum_{n_{11}, q_{11}} Pr(Q_{11} = q_{11}) Pr(N_{11} = n_{11}) \dots \\ \big(p_{11} == \min(n_{11}, q_{11})\big) \text{ , if } l_{31}^A = 0 \; and \; l_{22}^A = 0 \\ \sum_{s, n_{11}, n_{31}} Pr(S = s) Pr(N_{11} = n_{11}) Pr(N_{31} = n_{31}) \dots \\ G(p_{11}), \text{ otherwise} \end{cases}$$

with $G(p_{11}) = Pr(p_{11} \text{ packets of } S_{11} \text{received in } s \text{ draws}) =$

$$\sum_{p_{31}}^{n_{31}+n_{11}-p_{11}} (p_{31} \le n_{31}) A + (p_{31} > n_{31}) B \ ,$$

with $A = Pr\Big(p_{31} \text{ of } S_{31} \text{ drawn until } s \text{ and } p_{11} \text{ of } S_{11} \text{ sent}\Big) = \binom{s}{p_{31}} v_3^{p_{31}} v_1^{s-p_{31}} \big((s - p_{31} > n_{31})(p_{31} == n_{31}) + (p_{31} \le n_{31})(s - p_{31} == n_{31})\big)$ and $B = Pr\Big(n_{31} \text{ of the } n_{31} \text{ sent and then } p_{31} - n_{31} \text{ sent from the } S_{11} \text{ and } p_{11} \text{ of } S_{11} \text{ sent until } s\Big)$ is given by eq. 3.1 with $Pr(\text{last packet in } S_{31} \text{ received at draw } b)$ changed to: $Pr(\text{last packet in } S_{31} \text{ received at draw } b \text{ and } p_{11} \text{ sent in } s) = \big(a - n_{31} + b - a - (p_{31} - n_{31}) == n_{31}\big)\ Pr(\text{all } S_{11} \text{ exhausted in } b \text{ draws}) + Pr(\text{no } S_{31} \text{ drawn in } s - b)\ Pr(\text{exactly } p_{11} - b + p_{31} \text{ sent between draws } b \text{ and } s)$ the latter being obtained in a similar manner as for P_{grs31}.

3.2.2 Evolution of the index dissemination distribution

The ODEs for $\tilde{X}_{Ic}$ and $\tilde{Z}_{Ic}^1$ can be written as:

$$\frac{d\tilde{X}_{Ic}}{dt} = \sum_{e=1}^{C} \beta_{ce} N_e N_c A_{R11,e,c} + \beta_{s_1 c} N_c A_{S11,c} - \beta_{cd_1} \tilde{X}_{Ic} A_{D11,c} \ .$$

$$\frac{d\tilde{Z}_{Ic}^1}{dt} = \sum_{e=1}^{C} \beta_{ec} N_e N_c A_{R31,e,c} \ .$$

The ODEs for $\tilde{Y}_{Ic}$ and $\tilde{Z}_{Ic}^2$ can be deduced from those of $\tilde{X}_{Ic}$ and $\tilde{Z}_{Ic}^1$, replacing 1 by 2 everywhere. We have the following components:

- $A_{D11,c}$: fraction of nodes in community c that have I of S_{11} in their buffer and that drop it upon meeting with D_1.

$$A_{D11,c} = \sum_{\mathbf{l}^B} P_j(c, \mathbf{l}^B) \big(1 - ptnh_{11}(I, \mathbf{l}^B)\big) \Big[P_I^{11} + (1 - P_I^{11}) \dots$$

$$\sum_{q_{11},n_{11},p_{11}} \frac{p_{11}}{n_{11}} Pr(P_{11} = p_{11})\Big] .$$

- $A_{S11,c}$: fraction of nodes in community c that are infected by S_{11}.

$$A_{S11,c} = \big(K_{S_1}(t) \leq I < K_{S_1}(t) + B\big)\big(K_{S_1}(t) \leq K_1^{'}\big)c_2 u_{s_{1c}}^{11}$$

where, $c_2 = \sum_{\mathbf{l}^B} P_j(c, \mathbf{l}^B) pnth_{11}(I, \mathbf{l}^B) Pr\big(\zeta \geq I - K_{S_1}(t)\big)\big(B - l^B \geq I - K_{S_1}(t)\big)$.

- $pnth_{11,c}(I, \mathbf{l}^B)$: probability for node B not to have I of S_{11} in its buffer.

$$pnth_{11,c}(I, \mathbf{l}^B) = \frac{\sum_{j=l_{11}^B}^{B} S_{z,c}\big(K_{S_1}(t) - 1, j, \mathbf{v}_{T_c - \{I\}}(t)\big)}{\sum_{j=l_{11}^B}^{B} S_{z,c}\big(K_{S_1}(t), j, \mathbf{v}(t)\big)} ,$$

where $\mathbf{v}(t) = \mathbf{v}_{11}(c, t)$. We define $pnth_{31,c}(I, \mathbf{l}^B)$ similarly, replacing $\mathbf{v}_{11}(c, t)$ by $\mathbf{v}_{31}(c, t)$.

- $A_{R11,e,c}$: fraction of nodes in community c without index I of S_{11} that obtain I from a relay in community e.

$$A_{R11,e,c} = \begin{cases} 0 \quad \text{, if } \sum_{e=1}^{C} \big(\tilde{X}_{Ie} + \tilde{Z}_{Ie}^{1}\big) \geq M \\ \\ \sum\limits_{\mathbf{l}^A, \mathbf{l}^B} P_j(e, \mathbf{l}^A) P_j(c, \mathbf{l}^B) pnth_{11,c}(I, \mathbf{l}^B) pnth_{31,c}(I, \mathbf{l}^B) \ldots \\ \big(1 - pnth_{11,e}(I, \mathbf{l}^A)\big) \sum\limits_{s,n_{11},n_{31}} Pr(S = s) Pr(N_{11} = n_{11}) \ldots \\ Pr(N_{31} = n_{31}) \frac{p_{11}}{n_{11}} G(p_{11}), \text{ otherwise} \end{cases}$$

with $G(p_{11})$ is given in the above section. The term $\frac{p_{11}}{n_{11}} G(p_{11})$ is the probability that I is chosen to get forwarded given these conditions.

- $A_{R31,e,c}$: fraction of nodes in community c without index I of S_{31} that obtain I from another relay in community e.

$$A_{R31,e,c} = \begin{cases} 0 \quad \text{, if } \sum_{e=1}^{C} \big(\tilde{X}_{Ie} + \tilde{Z}_{Ie}^{1}\big) \geq M \\ \\ \sum\limits_{\mathbf{l}^A, \mathbf{l}^B} P_j(e, \mathbf{l}^A) P_j(c, \mathbf{l}^B) pnth_{11,c}(I, \mathbf{l}^B) pnth_{31,c}(I, \mathbf{l}^B) \ldots \\ \Big((l_{31}^A > 0) A_R^{Case1} + \big((l_{31}^A = 0)(l_{11}^A > 0) \ldots \\ (l_{22}^A \; or \; l_{32}^A) > 0)\big) A_R^{Case2}\Big) \text{ , otherwise .} \end{cases}$$

$$A_R^{Case1} = \big(1 - pnth_{31,e}(I, \mathbf{l}^A)\big) pnth_{11,e}(I, \mathbf{l}^A) \sum_{\substack{p_{31} \leq B - l^B, \\ s, n_{31}, n_{11}}} (p31 \leq n31) \frac{p_{31}}{n_{31}} H + pnth_{31,e}(I, \mathbf{l}^A) \ldots$$

$$\big(1 - pnth_{11,e}(I, \mathbf{l}^A)\big) \sum_{\substack{p_{31} \leq B - l^B, \\ s, n_{31}, n_{11}}} (p31 > n31) \frac{(p_{31} - n_{31})}{n_{11}} L ,$$

$$A_R^{Case2} = \big(1 - pnth_{11}(I, \mathbf{l}^A)\big) \sum_{\substack{p_{31} \le B - l^B, \\ q_{31}, n_{11}}} \frac{p_{31}}{n_{11}} M \ .$$

The expressions of H, L and M are easily derived from the decomposition of P_{grs31} in the above section.

- $P_I^{11}(t)$: probability that D_1 has received index I of S_{11} by time t.

$$\frac{dP_I^{11}(t)}{dt} = \sum_{c=1}^{C} \beta_{cd_1} N_c \big(1 - P_I^{11}(t)\big) A'_{D11,c} \ ,$$

where $A'_{D11,c}$ is the fraction of nodes in community c that hold I of S_{11} and that hand I over to D_1 provided that D_1 does not have I. We have:

$$A'_{D11,c} = \sum_{\mathbf{l}^B} P_j(c, \mathbf{l}^B)(1 - pnth_{11}(I, \mathbf{l}^B))H \ ,$$

with H the probability that I is selected into the forwarded indices: $H = \sum_{\substack{p_{11}, \\ n_{11}, q_{11}}} \frac{p_{11}}{n_{11}}$ $Pr(Q_{11} = q_{11})Pr(N_{11} = n_{11})\big(p_{11} = \min(n_{11}, q_{11})\big)$, the selection is uniformly at random amongst the eligible indices here, but can be with the lowest spray-counter as modeled in [103].

Decoding Criterion

Let $P_{S_1}(\tau)$ be the success probability at time τ. To account for the possible benefit brought by coding while keeping a simple criterion, we consider that D_1 can recover the K_1 packets sent by S_1 if (i) it receives at least K_1 indices of S_{11}, or (ii) if it receives non-coded and coded packets so that all the K_1 and K_2 packets are received. Note that case (ii) is pessimistic as the coding matrix can be inverted even though it is not met, but it is so in order to keep a tractable decoding criterion. Yet, it allows to account for a coding benefit. Hence we have: $P_{S_1}(\tau)$=$P_{S_1}^{(i)}(\tau)$+$P_{S_1}^{(ii)}(\tau)$.

$$P_{S_1}^{(i)}(\tau) = \sum_{k=K_1}^{K_1'} S_z\big(K_1', k, \mathbf{P}^{11}(\tau)\big) \ ,$$

$$\begin{aligned}
P_{S_1}^{(ii)}(\tau) = & \sum_{r=1}^{K_1} \sum_{s=0}^{K_2} Pr(K_1 - r \text{ of } S_{11}, \text{ at least } r \text{ of } S_{31}, \ldots \\
& K_2 - s \text{ of } S_{22}, \text{ at least } s \text{ of } S_{32}) \ , \\
= & \sum_{r=1}^{K_1} \sum_{s=0}^{K_2} S_z\big(K_1', K_1 - r, \mathbf{P}^{11}(\tau)\big) \ldots \\
& \left(\sum_{u=r}^{K_1'} S_z\big(K_1', u, \mathbf{P}^{31}(\tau)\big) \frac{\sum_{t=r}^{u} S_z(K_1', K_1 - r + t, \mathbf{v}_1)}{\sum_{t=0}^{u} S_z(K_1', K_1 - r + t, \mathbf{v}_1)} \right) \ldots
\end{aligned}$$

$$S_z\big(K_2', K_2 - s, \mathbf{P}^{22}(\tau)\big)\dots$$

$$\left(\sum_{u=s}^{K_2'} S_z\big(K_2', u, \mathbf{P}^{32}(\tau)\big)\frac{\sum_{t=s}^{u} S_z(K_2', K_2 - s + t, \mathbf{v}_2)}{\sum_{t=0}^{u} S_z(K_2', K_2 - s + t, \mathbf{v}_2)}\right),$$

where, $\mathbf{v}_1 = \frac{K_1 - r}{K_1 - r + u}\mathbf{P}^{11}(\tau) + \frac{u}{K_1 - r + u}\mathbf{P}^{31}(\tau)$ and $\mathbf{v}_2 = \frac{K_2 - s}{K_2 - s + u}\mathbf{P}^{22}(\tau) + \frac{u}{K_2 - s + u}\mathbf{P}^{32}(\tau)$.

3.2.3 Numerical validation

In this section, we assess the accuracy of the fluid model above, that captures the effect of the joint control of routing and ISNC on various quantities. We consider a synthetic contact trace on which we run the ISNC protocol described in Algorithm 1-2 thanks to a discrete event simulator written in Matlab. The simulation results are averaged over 30 runs and the 5% confidence intervals are plotted. The trace is made of $N = 1000$ nodes, $C = 1$ for the sake of clarity of the curves and $\beta = 5.10^{-4}$. The buffer size is set to $B = 2$ packets. The bandwidth is Poisson distributed with mean $Bw = 3$ packets. The communication settings of the two sessions are: $K_1 = K_2 = 1$, $K_1' = K_2' = 4$ and $M = Q = 50$. We set the control policy $\mathbf{u}$ to $u_{11} = u_{22} = 0.3$, $u_{31} = 0.4$ and $u_{32} = 0$. Figure 3.2 depicts the number of nodes infected with each type packets, namely $\sum_{i=1}^{B} X_i$, $\sum_{i=1}^{B} Y_i$, $\sum_{i=1}^{B} Z_i^1$ and $\sum_{i=1}^{B} Z_i^2$. We observe the relative good fit between analysis and simulation for both non-coded and coded type packets. Figure 3.3 represents the evolution of the number of DoFs of S_1 (resp. S_2) received by D_1. These numbers of DoFs are determined by a Gauss-Jordan elimination of the coding matrix in the simulation, and by the sum of the pairwise different received indices of S_{11} and S_{31} (resp. S_{22} and S_{32}) in the analytical model. We observe a good fit between the simulation results and the analytical prediction.

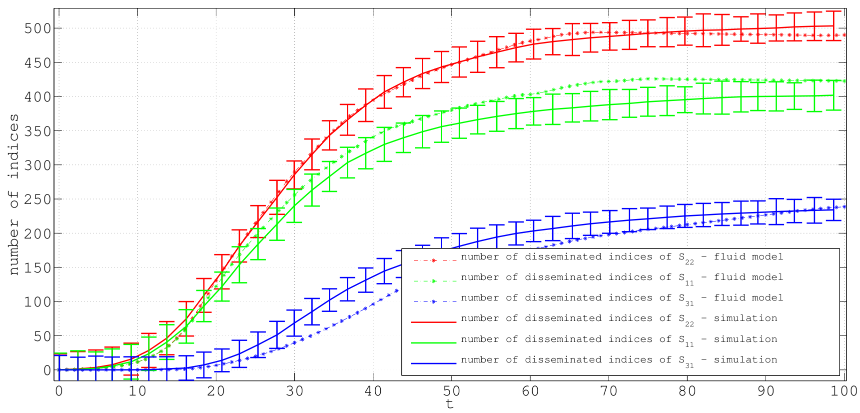

FIGURE 3.2: Evolution along time of the number of infected nodes with packets of different types.

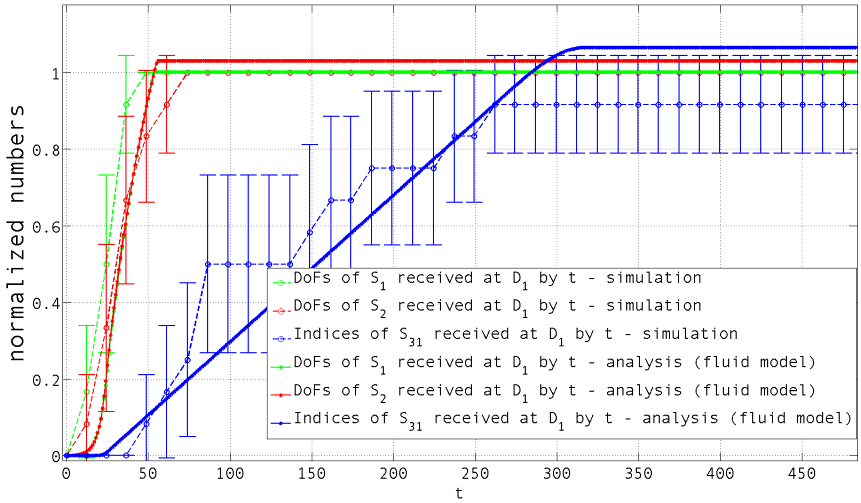

FIGURE 3.3: Evolution along time of the number of DoF of each source received by D_1.

3.3 The ISNC control problem

The problem we want to address thanks to (i) the introduced control policy of routing and ISNC and (ii) the fluid model that predicts the delivery probability, is that of control policy optimization under some energy or memory constraint.

3.3.1 Discussion on the optimization problem

Let $U(.)$ be any classical utility function, such as $\log(1+x)$ if x is a probability, and $P_1(\tau)$ (resp. $P_2(\tau)$) the probability that D_1 (resp. D_2) has obtained its K_1 information packets by τ. The problem of finding the optimal policy $\mathbf{u}$ which jointly controls routing and pairwise ISNC decisions under some energy constraint can be formulated as:

$$\max_{\{\mathbf{u_{ec}}\}_{\mathbf{e,c=1}}^{\mathbf{C}}} \quad obj(\tau) = U(P_1(\tau)) + U(P_2(\tau))$$

subject to $\mathbf{u}$ satisfying the energy constraint.

Note that other objectives, such as the mean completion delay for each session, can be considered. Optimizing ISNC decisions is a difficult problem in general, and in the social DTN scenario considered, this problem corresponds to a Markov decision process where at each time step, a central controller chooses an action so as to maximize the expected reward over a finite time horizon. It has been shown in [104] that when the system is made of N interacting objects and the occupancy measure is a Markov process (that has been discussed in Section 3.2), the optimal reward converges to the optimal reward of the mean field approximation of the system, which is given by the solution of an Hamilton-Jacobi-Bellman (HJB) equation.

Thanks to the ODEs presented in Section 3.2, that allow to get the fluid limits $P_{S_1}(\tau)$ and $P_{S_2}(\tau)$ of $P_1(\tau)$ and $P_2(\tau)$, the optimal ISNC policy for a finite N can hence be approximated by the asymptotically optimal policy built by solving the HJB equation for the associated mean field limit. However, owing to the intricacy of our model made of coupled ODEs, the HJB equation cannot be solved in a closed-form. We would hence need to resort to a numerical solver, but the dimension of the involved vectors prevents from using this kind of solvers (see, e.g., [105]). A feasible implementation of the optimization procedure is to use heuristic optimization methods, such as Differential Evolution [102]. Besides, let us specify that in DTN, a simple way of accounting for the energy consumption incurred by a routing policy is for example with the number of transmissions. This number can be easily extracted from the quantities modeled in Section 3.2, allowing to implement the energy constraint in the optimization process.

Investigating this optimization problem in social DTN thanks to the above fluid model is the subject of this thesis. In particular, in order to design a decentralized ISNC policy, the model will be adapted to powerful existing decentralized routing policies (such as SimBet [4]) in order to devise relevant local ISNC decisions. This is the subject of Chapter 5.

3.3.2 Numerical example

We now provide a numerical example that shows the relevance of the approach trying to get benefit from ISNC in social DTN. We consider the topology depicted in Figure 3.4 where the communities 1 and 3 are connected through another community 2. Community 1 (resp. 3) is that of the source node of session 1 (resp. 2) and of the destination node of session 2 (resp. 1) (these are 4 different nodes). This topology refers to the toy-example of two Wifi stations willing to exchange packets through an access point (AP) [31]. In this case, transmissions and hence time and throughput are saved if the AP combines the packets of the two stations. Whether this kind of ISNC advantage can exist in DTN is an open question, in particular when we do not consider nodes anymore but communities, that is when the source and destination nodes of both sessions are not exchanged but represent four different nodes. On the simple topology of Figure 3.4, we illustrate in Figure 3.5 that ISNC can be indeed beneficial with respect to intra-session NC. The intra-session NC policy we compare to is the best dissemination policy we found amongst those with varying values of $\mathbf{u}_{22}$ controlling spreading in community 2. Zhang et *al.* have shown empirically in [93] that uncontrolled ISNC of different source-destination pairs is not beneficial in general in homogeneous DTN. We illustrate in Fig. 3.5 that it can be beneficial in heterogeneous DTN. Specifically, it turns out that that the highest benefit is obtained when session mixing is performed at the side communities, and not at the relay community, as the direct analogy with connected networks would suggest. Further study is needed, allowed by the presented protocol and its analytical model, to investigate on what social graph topologies (amongst which undirected like in Fig. 3.4) and under what conditions ISNC can be beneficial. This is the subject of the next chapter.

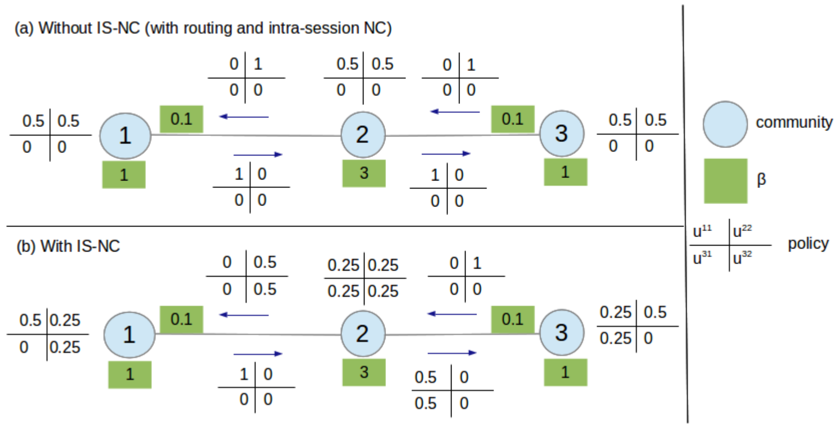

FIGURE 3.4: The community topology considered. Each community is made of 333 nodes and the inter-meeting intensities are in green. For both policies: $K_1 = K_2 = 2$, $K'_1 = K'_2 = 5$, $M = Q = 60$. (a) The non ISNC policy (that is, a policy where only intra-session NC is used). (b) The ISNC policy.

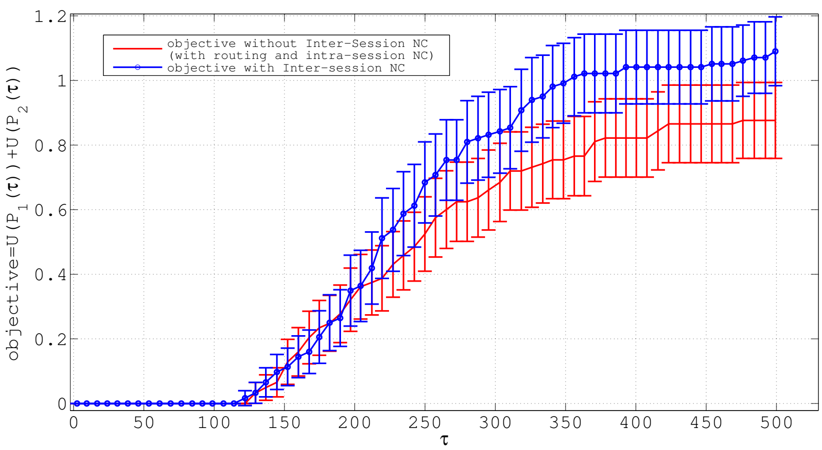

FIGURE 3.5: The obtained objective values for the above social DTN topology and policies.

3.3.3 Relevance to real-world traces

Finally, let us briefly show that even the simple topology of Figure 3.4 can arise in real-world social DTN. We consider as an example the MIT Reality Mining contact trace [70], corresponding to Bluetooth contacts collected with 100 smartphones distributed to students and staff at MIT over a period of 9 months. The people come into contact owing to the mobility and these contact patterns can reflect the social features such as the clustering of nodes into different communities, as analyzed in several studies such as [21]. We aggregated the contact trace into a weighted contact graph whose weights represent the tie strength (combining contact frequency and duration) between the nodes. We then applied Louvain community detection algorithm [106] to detect the communities in the contact graph and computed the β matrix describing the community structure. The following β matrix has been obtained with 7 communities detected:

$\beta_{MIT} =$

$$\begin{bmatrix} 2.12 & 0.09 & 0.03 & 0.00 & 0.01 & 0.05 & 0.01 \\ 0.09 & 5.89 & 0.52 & 0.17 & 0.73 & 0.33 & 0.12 \\ 0.03 & 0.52 & 1.71 & 0.14 & 0.47 & 0.27 & 0.09 \\ 0.00 & 0.17 & 0.14 & 1.97 & 0.19 & 0.13 & 0.07 \\ 0.01 & 0.73 & 0.47 & 0.19 & 12.00 & 0.64 & 0.21 \\ 0.05 & 0.33 & 0.27 & 0.13 & 0.64 & 10.87 & 0.17 \\ 0.01 & 0.12 & 0.09 & 0.07 & 0.21 & 0.17 & 6.48 \end{bmatrix}.$$

A similar topology as in Figure 3.4 arises when, for example, a node of community 1 and another node in community 4 want to exchange packets. In this case, a good route is to go through community 6, and the involved β_{ij} are then: $\beta_{11} = 2.1$, $\beta_{66} = 10.8$, $\beta_{44} = 1.9$, $\beta_{16} = 0.05$, $\beta_{64} = 0.13$.

3.4 Conclusion

We have devised a parameterized pairwise ISNC control policy and expressed the control optimization problem thanks to a performance model. The scheme is at the same time a routing and a coding policy that allows to optimize for a utility function defined over the two sessions, under some energy constraint. Our policy decides which nodes can mix the sessions based on their communities rather than on their individual properties, making the devised policy scalable with the number of nodes if the number of communities keeps limited. We have shown that numerical gains of ISNC over intra-session NC can indeed be obtained.

This chapter does not aim at presenting a self-contained decentralized ISNC protocol that can be confronted with existing routing policies in DTN. It aims at devising, modeling and proving the benefit of a centralized social-aware (community-based) pairwise ISNC policy. Specifically, the problem of grouping sessions by two is not investigated here. Detecting and selecting what pairs of sessions to be mixed is part of the decentralization problem. Moreover, further study is needed, allowed by the presented protocol and its analytical model, to investigate on what social graph topologies (amongst which undirected like in Fig. 3.4) and under what conditions (e.g., sizes K_1 and K_2 of the sessions and energy budget) ISNC can be beneficial. The next step after this work is to study numerically the optimization problem in order to extract heuristics to devise a decentralized ISNC policy for social DTN. In particular, the model will be adapted to powerful existing decentralized routing policies (such as SimBet [4]) in order to devise relevant local ISNC decisions.

Chapter 4

Experimental study of pairwise Inter-session NC in DTMSN

The previous chapter has presented a ISNC policy and an analytical model where the number of packets per session can be any, corresponding to the case where a file is split into several packets, and the metric (whether it be delay or delivery probability) is on the whole file. Also, 2 sessions are considered. In order to tackle the optimization problem of ISNC policy by reducing the parameter space and identifying sound heuristics, we first analyze in the present chapter the impact of various parameters on the ISNC performance. This chapter therefore aims at identifying, through progressive experiments on toy topologies, what can be the advantages of ISNC in DTMSN, specifically under what assumptions on routing, buffer management, and the network load.

In [94], NC is considered at some intermediate hub nodes, but only across packets destined to the same destination node. In [93], Zhang *et al.*, consider both intra- and inter-session NC in homogeneous DTN. For unicast sessions with different sources and destinations, uncontrolled ISNC is shown not to perform better than intra-session. In this thesis we tackle the more general problem of ISNC for unicast sessions with different destinations. When considering several unicast sessions, ISNC can bring throughput and fairness gains [2, 91] both on lossless and lossy links. However, the optimization problem of ISNC for multiple unicast sessions has been proven NP-hard [25], in particular because of the joint problems of subgraph selection and coding decisions, that can be solved independently for a single multicast session [90]. Therefore, all the works addressing the problem of ISNC target suboptimal, yet continually improved, methods [2, 29, 31, 91]. These approaches are not directly applicable to DTN as they assume fixed topologies and may incur heavy signaling. In particular, in [29], Eryilmaz and Lun introduced a routing-scheduling-coding strategy using back-pressure techniques. Modeling the coded flows as *poisoned* [2], the queue-length exchange is meant to determine the location of the encoding, decoding, and remedy generating nodes. Furthermore, all these works considered directed networks. However, there is a priori no reason for considering that two nodes can exchange packets in a single direction in DTN. Li and Li in [30] have shown theoretically what can be the maximum throughput improvement with intra-session NC in undirected networks, compared with what can be obtained with integral, half-integer and fractional routing. In particular, for the multicast problem, the throughput increase ratio is upper- bounded by two between NC and half-integer routing, or even less with fractional routing [30, 95]. However, the shared resources (buffer, contact bandwidth) in DTMSN make ISNC attractive as in wireless mesh networks [31], though these networks are undirected. Hence, to tackle the open problem of ISNC design in social DTN, one of the steps we choose to take is to study through experiments on simple topologies, made of homogeneous node communities connected together in a heterogeneous manner, what improvement can be brought by ISNC, and how.

The first section deals with a simple chain topology, while the second investigates the typical butterfly topology. In these topologies, the graph nodes do not represent network nodes anymore, but entire node communities.

4.1 The chain topology

As mentioned above, a well-known application of ISNC is with opportunistic routing in wireless ad hoc mesh networks, such as the COPE framework [31]. The very first example is that of Figure 4.1 where two stations are connected through a third one, such as an Access Point (AP), the station on one side being both a source and destination for 2 sessions established with the other side session. When only one station can send a packet at a time (like in WiFi if they are all three on the same channel), then ISNC at the AP allows to save one transmission over 4 needed otherwise without ISNC. In such a topology, the transmissions are bi-directional but ISNC is beneficial owing to the constrained resource sharing (the wireless medium is shared in time). Hence, this is the first topology we chose to investigate, as a simplistic DTMSN where ISNC might be beneficial.

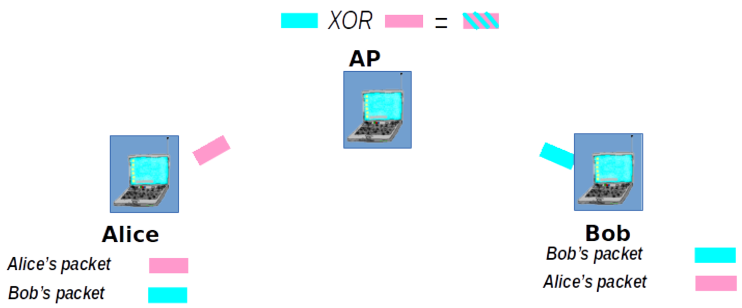

FIGURE 4.1: Chain topology with two stations connected through an Access Point (AP).

4.1.1 Starting point

Figure 4.2 represents a chain topology consisting of 3 communities, with the intra- and inter-meeting intensities β_{ij}, for $i, j \in 1, 3$ and buffer size of $B = 2$. Source node S_1 (resp. S_2) of session-1 (resp. 2) is in community $C_{S_1} = 1$ (resp. $C_{S_2} = 2$), and destination node D_1 (resp. D_2) of session-1 (resp. 2) is in community $C_{D_1} = 2$ (resp. $C_{D_2} = 1$). The buffer management for the protocol is presented in Table 4.1.

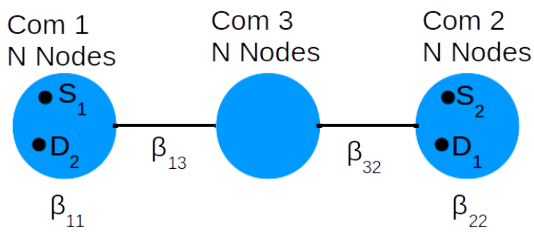

FIGURE 4.2: Chain topology with 3 communities.

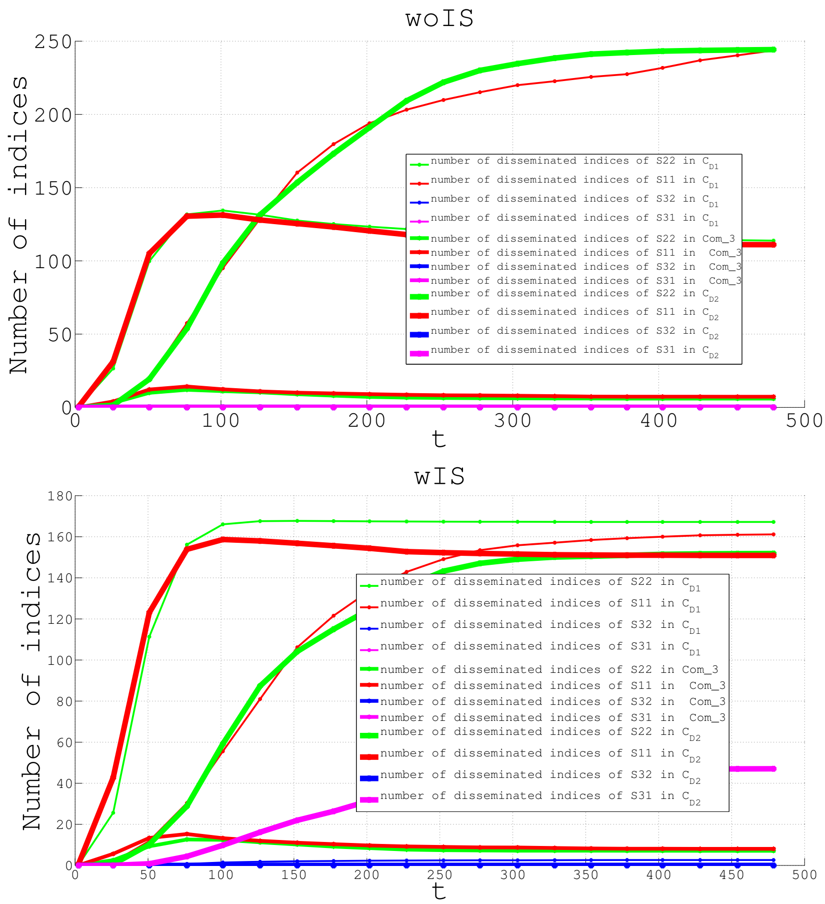

FIGURE 4.3: Evolution along time of the number of infected nodes with packet of different types taking $K = 2$.

As observed on the utilities in Figure 4.4, ISNC does not bring any gain here. Let us analyze the level of S_{22} packets in community $C_{D_2} = 1$. We verify from Figures 4.3 and 4.4 that the

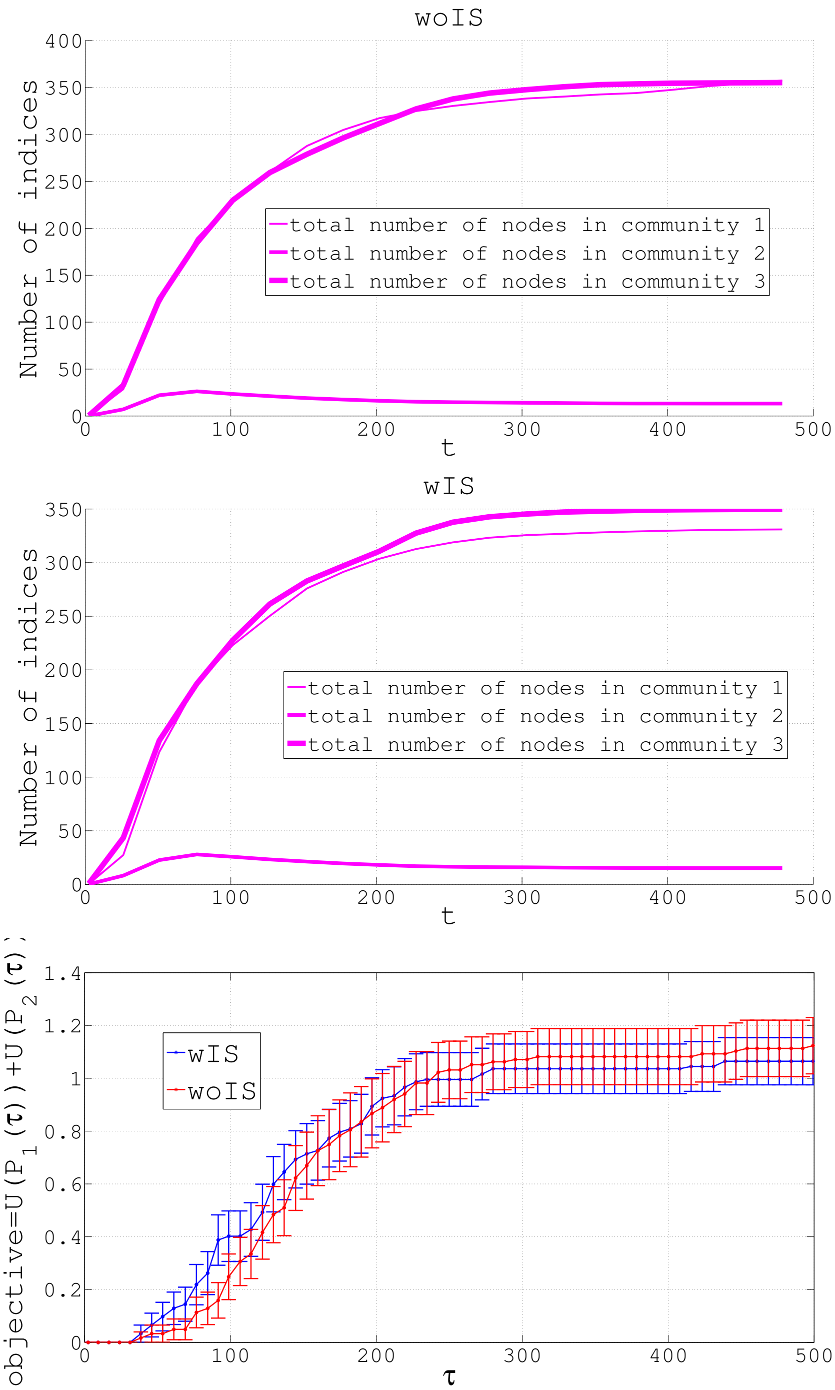

FIGURE 4.4: Simulation results for chain topology with 3 communities.

analyzed quantities are symmetric as expected, and the following observations hold for $C_{D_1} = 2$ as well. We can see in community $C_{D_2} = 1$ that the level of S_{22} is lower with ISNC (wIS) than without ISNC (woIS), but the fraction of S_{31} in wIS that replaces S_{22} does not help the decoding. Slightly changing the dissemination protocol or the buffer management choices (what packet can erase what other packet) yield different levels of S_{22} and S_{31}, but without higher gains for ISNC.

The constrained resource that is shared is the relays' buffers. So the performance impediment that prevents the destination node from receiving the desired packets is the occupancy of the relay buffers in the destination community. In the destination community, the buffer management can be such that the the packets not destined to this community are erased by those which are. However, doing so might prevent the packets originating in this community to spread enough so as to reach their destination community. In such a case, ISNC can be thought to bring some gain. In order to investigate properly what are the session and buffer management parameters for which such a phenomenon may appear, we perform the following analysis.

4.1.2 Analysis

Let us consider a more simple setup than above, with only 2 communities with 2 reverse sessions as above. We first consider the toy topology depicted in Figure 4.5(a), with 2 reverse sessions having source-destination nodes (S_1, D_1) and (S_2, D_2). Our goal is first to determine whether ISNC can improve the probability of message delivery at both destinations.

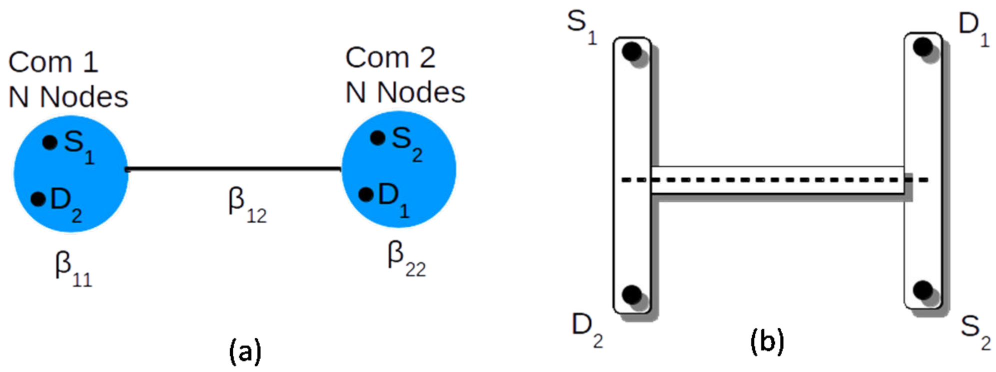

FIGURE 4.5: Bandwidth sharing in a network topology with 2 communities.

The constrained resource that is shared is the relays buffers. So what may impede the destination node from receiving the desired packets is the occupancy of the relays' buffers. In the destination community, the buffer management can be such that the the packets not destined to this community are erased by those which are. However, doing so might prevent the packets originating in this community to spread enough so as to reach their destination community. In such a case, ISNC can be thought to bring some gain.

Let us first analyze the buffer management problem without ISNC. A number of works have analyzed the buffer management (and scheduling for buffer sizes greater that 1) in DTMSN, such as [74, 78], and references therein. In particular, [78] derive an optimal drop policy that discriminates packets based especially on their number of copies and TTL, and for heterogeneous networks, based on the joint distribution of the latter and inter-meeting intensities. In our work, we chose not to implement this buffer refined management, thereby leaving more odds of outperforming a non ISNC policy to ISNC. We consider the general case where a message can be made of several packets, namely here $K = 20$ packets for a buffer size $B = 1$,

Packet to send	Conditions	No. of copies of an index to be stored at node n_2
S_{11}	$buf\_size < B$	$(C_{n1}, C_{n2}) = (1,1) \| (C_{n2}, C_{n1}) = (1,1) \| (C_{n1}, C_{n2}) = (3,3) \| (C_{n2}, C_{n1}) = (3,3)$: S_{11} with $\frac{nbcop\_n_1}{2}$ $(C_{n1}, C_{n2}) = (1,2) \| (C_{n1}, C_{n2}) = (2,3)$: $nbcop\_n_1 - 1$
	$buf\_size < B$ and $(C_{n1}, C_{n2}) = (2,3)$	if n_2 has S_{22}: create S_{32} with $nbcop\_n_1 - 1 + nbcop\_n_2$ if n_2 has S_{32}: mix packets and set index S_{32} with $nbcop\_n_1 - 1 + nbcop\_n_2$
S_{22}	$buf\_size < B$	$(C_{n1}, C_{n2}) = (3,3) \| (C_{n2}, C_{n1}) = (3,3) \| (C_{n1}, C_{n2}) = (1,1) \| (C_{n2}, C_{n1}) = (1,1)$: S_{22} with $\frac{nbcop\_n_1}{2}$ $(C_{n1}, C_{n2}) = (3,2) \| (C_{n1}, C_{n2}) = (2,1)$: $nbcop\_n_1 - 1$
	$buf\_size < B$ and $(C_{n1}, C_{n2}) = (2,1)$	if n_2 has S_{11}: create S_{31} with $nbcop\_n_1 - 1 + nbcop\_n_2$ if n_2 has S_{31}: mix packets and set index S_{31} with $nbcop\_n_1 - 1 + nbcop\_n_2$
S_{32}	$(C_{n1}, C_{n2}) = (3,3)$	if $buf\_size < B$: S_{32} with $nbcop\_n_1$
		if $buf\_size == B$: if n_2 has S_{22}, mix packets and set index S_{32} with $\frac{nbcop\_n_1}{2} + nbcop\_n_2$
	$(C_{n1}, C_{n2}) = (3,2)$	if $buf\_size < B$: S_{32} with $nbcop\_n_1$
		if $buf\_size == B$: if n_2 has S_{32}, mix packets and set index S_{32} with $\frac{nbcop\_n_1}{2} + nbcop\_n_2$
	$(C_{n1}, C_{n2}) = (2,2)$	if $buf\_size < B$: S_{32} with $nbcop\_n_1$
		if $buf\_size == B$: if n_2 has S_{32}, mix packets and set index S_{32} with $\frac{nbcop\_n_1}{2} + nbcop\_n_2$
	$(C_{n1}, C_{n2}) = (2\, or\, 1, 1)$	if $buf\_size < B$: S_{32} with $nbcop\_n_1$
		if $buf\_size == B$: if n_2 has S_{22}, mix packets and send index S_{32} with $\frac{nbcop\_n_1}{2} + nbcop\_n_2$
S_{31}	$(C_{n1}, C_{n2}) = (1,1)$	if $buf\_size < B$: S_{31} with $nbcop\_n_1$
		if $buf\_size == B$: if n_2 has S_{11}, mix packets and set index S_{31} with $\frac{nbcop\_n_1}{2} + nbcop\_n_2$
	$(C_{n1}, C_{n2}) = (1,2)$	if $buf\_size < B$: S_{31} with $nbcop\_n_1$
		if $buf\_size == B$: if n_2 has S_{31}, mix packets and set index S_{31} with $\frac{nbcop\_n_1}{2} + nbcop\_n_2$
	$(C_{n1}, C_{n2}) = (2,2)$	if $buf\_size < B$: S_{31} with $nbcop\_n_1$
		if $buf\_size == B$: if n_2 has S_{31}, mix packets and set index S_{32} with $\frac{nbcop\_n_1}{2} + nbcop\_n_2$
	$(C_{n1}, C_{n2}) = (2\, or\, 3, 3)$	if $buf\_size < B$: S_{31} with $nbcop\_n_1$
		if $buf\_size == B$: if n_2 has S_{11}, mix packets and set index S_{31} with $\frac{nbcop\_n_1}{2} + nbcop\_n_2$

TABLE 4.1: Buffer management between two meeting nodes n_1 and n_2.

$u_{11} = \begin{bmatrix} 0.5 & 1 & 0 \\ 0 & 0.5 & 1 \\ 0 & 0 & 0.5 \end{bmatrix}$	$u_{22} = \begin{bmatrix} 0.5 & 0 & 0 \\ 1 & 0.5 & 0 \\ 0 & 1 & 0.5 \end{bmatrix}$	$u_{31} = u_{32} = \begin{bmatrix} 0.25 & 0 & 0 \\ 0 & 0.25 & 0 \\ 0 & 0 & 0.25 \end{bmatrix}$

TABLE 4.2: The values of u_{ij} considered for chain topology with 3 communities.

$\beta_{11} = \beta_{22} = 0.05$, $N_c = 50$, p is the probability that a packet of session i replaces a packet of session $1 - i$ in community C_{D_i}. Figure 4.6 shows the probability that each destination has received at least 10 (different) information packets by time $500s$, when the the meeting intensity β_{12} between both communities varies. This quantity is obtained by the fluid model described below (with the same notation as in the previous chapter, and matches well the simulations of this simple setting).

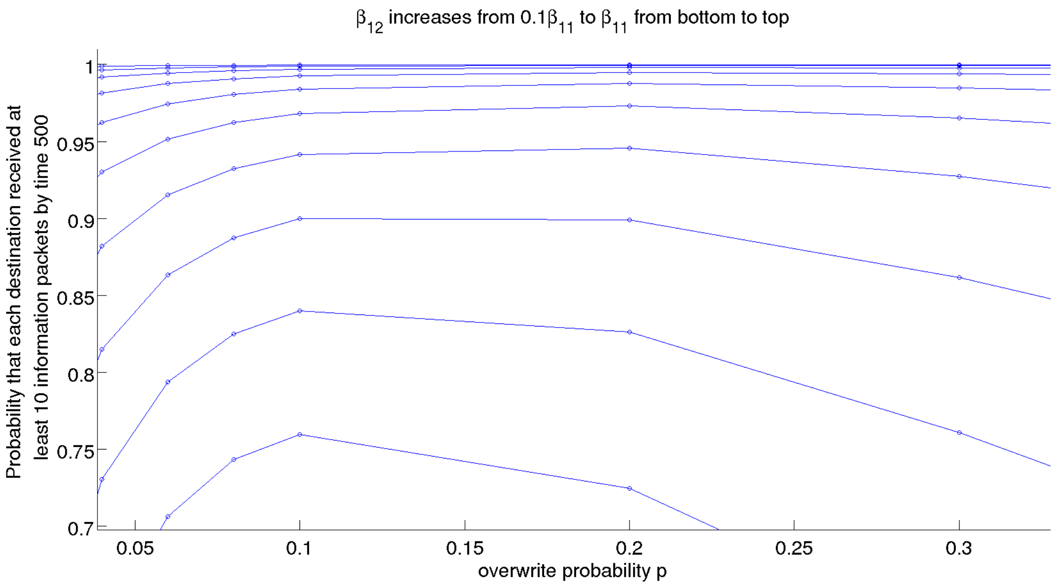

FIGURE 4.6: Performance analysis for network topology with 2 communities.

$$\frac{dX_{k,1}}{dt} = \beta_{11}\Big(pois_{k,1}(t) + X_{k,1}\Big)\Big(N - \sum_k X_{k,1} - \sum_k Y_{k,1}\Big)\dots$$

$$-pX_{k,1}\Big(\beta_{11}\sum_k Y_{k,1} * +\beta_{12}\sum_k Y_{k,2}\Big),$$

$$\frac{dX_{k,2}}{dt} = \Big(\beta_{12}\big(pois_{k,1}(t) + X_{k,1}\big) + \beta_{22}X_{k,2}\Big)\Big(N - \sum_k X_{k,2} - \sum_k Y_{k,2} + p\sum_k Y_{k,2}\Big),$$

$$\frac{dY_{k,1}}{dt} = \Big(\beta_{12}\big(pois_{k,2}(t) + Y_{k,2}\big) + \beta_{11}Y_{k,1}\Big)\Big(N - \sum_k Y_{k,1} - \sum_k X_{k,1} + p\sum_k X_{k,1}\Big),$$

$$\frac{dY_{k,2}}{dt} = \beta_{22}\Big(pois_{k,2}(t) + Y_{k,2}\Big)\Big(N - \sum_k Y_{k,2} - \sum_k X_{k,2}\Big)\dots$$

$$-pY_{k,2}\Big(\beta_{22}\sum_k X_{k,2} * + \beta_{12}\sum_k X_{k,1}\Big).$$

We observe that the optimal value of the overwrite probability p varies from 0.3 to 0.1 when β_{12} decreases. This is explained by the fact that the lower β_{12}, the slower the propagation of packets from a community to the other. So once a packet makes it successfully to the destination community, it must not propagate too fast in the latter so as to give time to all the information packets generated and hold by this community to cross to their destination community, before getting erased.

This corresponds to bandwidth sharing between the two sessions, as depicted in Figure 4.5(b) and readily seen from a flow perspective. In order to maximize a utility defined as a sum of concave functions of the delivery probabilities of each session, the best is indeed to equally share the common bandwidth so as to equalize the delivery probabilities. One can then wonder whether ISNC can help share the bandwidth between both reverse sessions in a more efficient way, that is, if the buffer management was given an extra choice that is instead of replacing packets, mixing them and thereby generating ISNC packets, would that bring any gain in utility (by serving both communities with the same packet)? Or if p optimal p is not known. More specifically, let us consider that the node buffer size is 1 packet, that a 2 community-1 nodes ν_1 and ν_2 meet, ν_1 already holds P_1, a session-1 packet, while ν_2 holds P_2, a session-2 packet. Instead of having ν_2 overwriting P_1, ν_1 can store P_3, the XOR (or any finite field sum) of P_1 and P_2, so that: (i) if ν_1 then meets with D_2 which is likely to have received P_1, D_2 can recover the P_2 it is interested in, (ii) if ν_1 then meets with D_1 which is likely to have received P_2, D_1 can recover P_1 it is interested in.

In order to investigate this question, we perform the following analysis to identify in what conditions ISNC can bring some gain. Let us consider a single packet per session and no packet drop leaving more odds of outperforming a non ISNC policy to ISNC.

Lemma 4.1.1. *Let Y_2 denote the fraction of nodes holding P_2 in community-2 when no ISNC is employed. A necessary condition for ISNC to outperform no ISNC is:*

$$\frac{Y_2}{N} > \frac{3+\alpha}{4}\ , \tag{4.1}$$

where, $\alpha = \beta_{12}/\beta_{22}$.

Proof: Without ISNC:

$$Pr\{S_1 \text{ be useful to } D_1 \text{ within } \Delta t | \text{a node holds } S_1\} Pr\{\text{a node holds } S_1\} =$$

$$\beta_{22}\Delta t(1 - \frac{Y_2}{N}),$$

$$Pr\{S_1 \text{ be useful to } D_2 \text{ within } \Delta t | \text{a node holds } S_1\} Pr\{\text{a node holds } S_1\} = 0,$$

With ISNC:

$$Pr\{S_3 \text{ be useful to } D_1 \text{ within } \Delta t | \text{a node holds } S_3\} Pr\{\text{a node holds } S_3\} \leq$$
$$\beta_{22} \Delta t \frac{Y_2}{N}(1 - \frac{Y_2'}{N}),$$
$$Pr\{S_3 \text{ be useful to } D_2 \text{ within } \Delta t | \text{a node holds } S_3\} Pr\{\text{a node holds } S_3\} \leq$$
$$\beta_{12} \Delta t \frac{X_1}{N}(1 - \frac{Y_2'}{N}),$$

where, assuming Δt low enough:

- $\beta_{22}\Delta t$ represents the probability that this coded packet hits D_1 within Δt from the time it has been generated in community-2, either by a combination of S_1 and S_2 or by replication.
- $\frac{Y_2}{N}$ (resp. $\frac{X_1}{N}$) represents the probability that D_1 (resp. D_2) has received S_2 (resp. S_1). This is an upper-bound on the actual probability as, assuming that the number of community-2 nodes infected (that have met) with session-2 packets is $\frac{Y_2}{N}$ at the end of Δt, $\frac{Y_2}{N}$ is the probability that a community-2 node holds at least one packet of session-2, not necessarily S_2. Packets are indeed sent continuously by both sessions.
- Y_2 (resp. Y_2') denotes the steady-state fraction of nodes holding S_2 in community-2 when no ISNC is employed (resp. when it is), and
$Pr\{\text{a node holds } S_1\} = X_2/N = (1 - Y_2/N)$ is the fraction of community 2 nodes holding S_1 in the steady-state (it is optimal for the success probabilities to occupy all the network nodes in the steady state). With ISNC however, community-2 nodes can carry either S_1, S_2 or S_3, therefore $(1 - Y_2'/N)$ is an upper-bound for $Pr\{\text{a node holds } S_3\}$.

Then,

$$Pr\{S_1 \text{ be useful to } D_1 \text{ or } D_2 \text{ within } \Delta t\} \leq \beta_{22}\Delta t(1 - \frac{Y_2}{N}),$$
$$Pr\{S_3 \text{ be useful to } D_1 \text{ or } D_2 \text{ within } \Delta t\} \leq \beta_{22}\Delta t \frac{Y_2}{N} + \beta_{12}\Delta t \frac{X_1}{N} \dots$$
$$-\beta_{12}\beta_{22}\frac{Y_2}{N}\frac{X_1}{N}(1 - \frac{Y_2'}{N})(\Delta t)^2.$$

Assuming that the last term is neglected as each component factor is by assumption lower than 1, we can translate the condition

$$Pr\{S_3 \text{ be useful to } D_1 \text{ or } D_2 \text{ within } \Delta t\} > Pr\{S_1 \text{ be useful to } D_1 \text{ or } D_2 \text{ within } \Delta t\}, \quad (4.2)$$

into the necessary condition:

$$(\beta_{22}\Delta t\frac{Y_2}{N}+\beta_{12}\Delta t\frac{X_1}{N})(1-\frac{Y_2'}{N})>\beta_{22}\Delta t(1-\frac{Y_2}{N}),$$

that is

$$-(\frac{Y_2'}{N})^2+\frac{Y_2'}{N}>\frac{1}{1+\alpha}(1-\frac{Y_2}{N}),$$

where, $\alpha=\beta_{12}/\beta_{22}$. Owing to the fact that $-x^2+x\leq 0.25$ for all $x\in\mathbb{R}$, a necessary condition to satisfy the above inequality is:

$$\frac{Y_2}{N}>\frac{3+\alpha}{4} \tag{4.3}$$

◇

When the communities 1 and 2 are merged, that is $\alpha=1$, no gain can be expected and we get back the result of Zhang *et al.* in [93]. There might exist values of $\alpha<1$ that lead to a gain of ISNC. In practice, we have not been able to find out values of α for which ISNC carried gain in utility compared to without ISNC with proper buffer management, for 1 and more packet per session. In particular, the above lemma may hold for several packets per session if we replace the definition of Y_2 by the total number of session-2 packets with intra-session NC, and similarly for the other quantities involved in the lemma. If that lemma would hold, to show that a ISNC gain is indeed possible, we would need the optimal value of $\frac{Y_2}{N}$ as a function of α, optimality in the sense that it maximizes the objective $U(P_1(\tau))+U(P_2(\tau))$. This might be obtained numerically using the fluid model (stemming from the mean-field approximation) presented above, letting the number of packets per session tend to infinity, finding the best parameter p controlling buffer management as described above, then getting back optimal Y_2/N. The computational intensity of such a computation gets quickly prohibitive (above $K=50$), and would be needed to be carried out for each α value, to see whether the so-obtained Y_2/N indeed verifies inequality-4.3.

Having identified that ISNC is likely not to bring gain in such a 2-community topology, we present next a topology closer to COPE.

4.1.3 A chain with a hub node

Such a topology is represented in Figure 4.7. It is made of two communities, whose nodes cannot meet except through a hub node. There are still two reverse sessions whose source and destination nodes are in opposite side communities.

The rationale for considering this topology as a good candidate to enable ISNC gain is the following:

- if the hub node buffer size is high enough, then there is no competition for buffer size access between both sessions, and ISNC is not expected to bring any advantage;

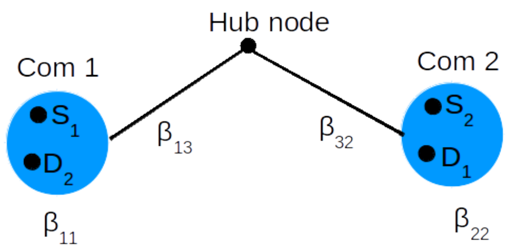

FIGURE 4.7: Chain topology with a hub node.

- if the hub node buffer size is constrained, say equal to one packet in the extreme case, then it may be beneficial for both sessions to allow the hub node to carry a coded packet that may serve both sessions, depending on what community the hub node meets next.

In order to investigate the possible gain, we carry out simulations on this topology. We consider $K = 1$ (each source generates a single packet), the buffer size is still 1 at all relay nodes. Let a packet destined to community i, for $i \in \{1, 2\}$, be denoted as a i-packet. We consider the following buffer management:

- when no relay nodes is allowed to mix both session packets, each time the hub node gets a copy of any packet, it overwrites that it may have held, and a i-packet overwrites in community i all the copies of the $(1 - i)$-packet.

- when the hub node holds the i-packet and meets with the $(1 - i)$-packet, then it stores the XOR of both packet (that is, a coded packet). This coded packet overwrites in community i the copies of the $(1 - i)$-packet, but not those of the i-packet.

We consider both cases, of unidirectional and bidirectional contacts: (i) in the unidirectional case, when two nodes meet, only one of them can transmit to the other one; (ii) in the bidirectional case, both meeting nodes can transmit to each other. Figure 4.8 (resp. Figure 4.9) shows the utility value obtained in the bidirectional (resp. unidirectional) case without and with ISNC. We observe that ISNC brings some gain only in the unidirectional case. This is explained as follows. At each meeting in the bidirectional case, if ISNC can be invoked, then that means that the hub node holds the i-packet and meets with a node of community $(1 - i)$ that holds the $(1 - i)$-packet. In such a case, the hub delivers (overwrites) the i-packet and gets the $(1 - i)$-packet. Therefore, ISNC is not useful. In this topology, ISNC can bring some gain only in the number of transmissions, if one transmission can serve both sessions. However, that is not possible in this framework, as sessions are in reverse directions and transmissions towards two communities cannot be simultaneous owing to the low density of the DTN. That would be the same problem if there were several relay hubs assembled within a daisy chain.

So we need to move on to another topology to make ISNC gain appear in a general case (not in a case restricted to unidirectional transmissions). As the transmissions are not simultaneous

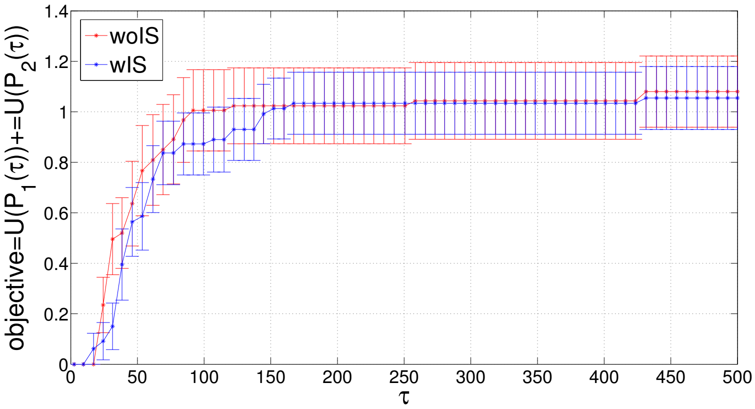

FIGURE 4.8: bidirectional, $K = 1$, $M = 200$.

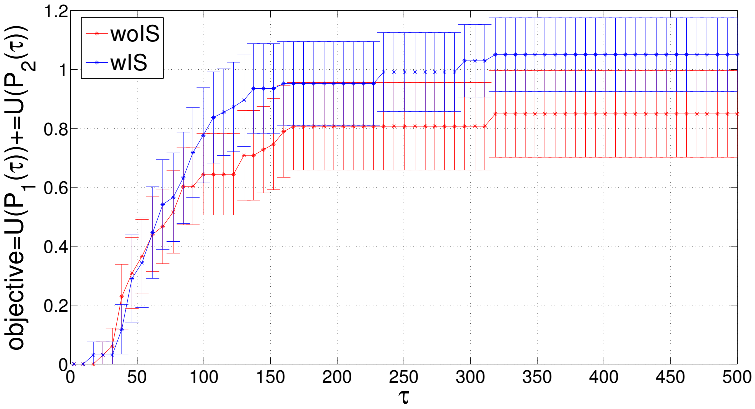

FIGURE 4.9: unidirectional, $K = 1$, $M = 200$.

in two different communities, as mentioned above, ISNC saving is in terms of occupied buffer space, so we need a transfer in the same buffer in the same direction for both sessions. This implies that S_1 and D_2 nodes cannot be in the same community anymore.

4.2 The butterfly topology

4.2.1 Preamble: ISNC and epidemic routing

Our goal is first to identify a topology where ISNC can be beneficial compared with no ISNC, then we study the impact of various network or client parameters on the performance (under what range of each parameter is ISNC beneficial or detrimental). To remedy the impediments of the above topologies in allowing ISNC gains, we consider a topology susceptible to alleviate these problems. We hence consider the well-known butterfly topology depicted in Figure 4.10, where there are 2 sessions with source-destination pairs as (S_1, D_1) and (S_2, D_2) for session-1 and session-2, respectively. In the remainder of this chapter, we consider only messages made of a single packet, that is $K = 1$ (and a session is by definition still made of a single message). So the terms packet, message and file are used interchangeably.

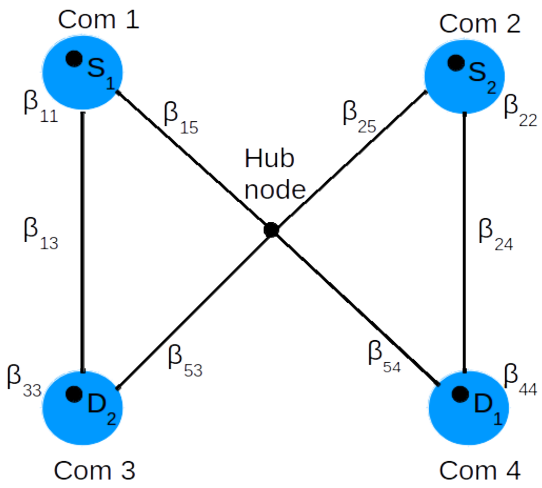

FIGURE 4.10: Butterfly topology: the side (blue-filled) communities are made each of $N = 250$ nodes, while the middle node is a single hub node. All nodes have buffer size 1, unless mentioned otherwise.

We run experiments, with $M = 200$, and we recall that M is the number of copies per message. In order to favor ISNC, we take the inter-meeting intensities as depicted in Figure 4.11, with $x = 5.10^{-3}$, the number of nodes per community is $N = 250$ and the intra-meeting intensity inside each community is $10x$. The ratios between inter-meeting intensities are such that the link between the hub node and the destination communities is the bottleneck, while it is easy for the destinations to get the remedy packets thanks to the high inter-meeting intensities on the side links.

The numerical results, we do not include here, show no improvement (in utility function computed over the two sessions, as presented in Chapter 3) with ISNC. The reason is as follows. The equivalent static graph of communities on which ISNC is used is undirected in the present case, meaning that the packet can flow in either ways between two meeting nodes of different communities. This is especially allowed by the chosen routing strategy: epidemic flooding here. Therefore, the routes taken by the packets are those depicted in Figure 4.11(a). Then ISNC cannot help even with a constrained buffer size (of 1 packet) at the hub node because the main paths taken by both sessions are opposite, and we get back to the situation on the daisy chain with reverse sessions described at the end of the last subsection. This is inline with the results of Li and Li in [30] who showed that the throughput increase brought by NC in undirected networks vanishes in front of fractional routing. Note that their results are obtained for multicast sessions and intra-session NC, but they are readily transposable to inter-session NC on unicast sessions, especially on the butterfly network. If only the routes depicted in Figure 4.11(b) were taken then ISNC would be useful.

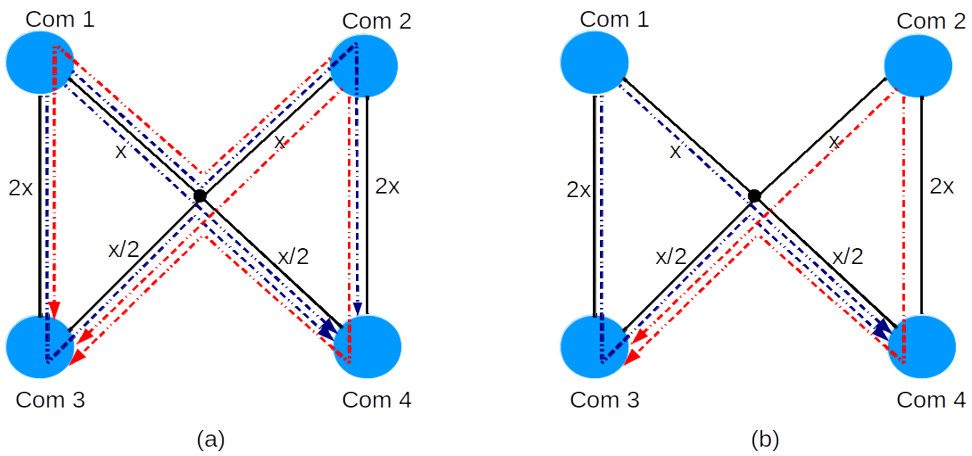

FIGURE 4.11: (a) All possible routes are depicted in blue and red for each session 1 and 2, respectively. (b) The only routes each session should take so as to get a gain with ISNC.

It hence turns out that the possible advantage of ISNC in this synthetic social DTN, whose communities form a butterfly graph but where contacts are inherently bidirectional, is closely tied to the routes the packets flow through in such mobile DTN. However, the way of operating such DTN is usually not to flood the packets to all possible nodes, for the sake of energy and memory consumptions, but rather to focus the copies to certain well-chosen relay nodes. Thus, routing in mobile social networks does not allow to take all the possible routes, such as depicted in 4.11(a). The next section therefore investigates the interplay between ISNC and social routing. In particular, we are going to show that on the set of routes selected by the SimBet routing algorithm [4], ISNC can be beneficial.

4.2.2 Combining ISNC and social routing

We now study how to enable ISNC when the routes are chosen by a social routing algorithm (such as BubbleRap [23], SimBet [4] or PeopleRank [64]). To do so, two distinct but correlated problems arise:

P1 Where (and when) must the coding be performed?

P2 Must remedy packets be sent, and if so, how to inform the source nodes?

Problem P1 relates to the choice of what nodes are allowed to mix messages of different sessions. These nodes can be either set fixed, and this is going to be our first approach to obtain ISNC gains on the butterfly topology with SimBet routing as a proof of concept, or decided online based on local decisions, and this will be investigated in the next chapter. P1 is related to P2 because, depending on if and when the remedy packets can reach the destination nodes, the nodes allowed to send mixed packets can differ.

Problem P2 can be solved in two different ways: (i) either a signaling mechanism is set up, such as that of Eryilmaz and Lun [29] or Kreishah *et al.* [91], to explicitly inform the source nodes to send their packets to destinations they are not destined to, (ii) or no signaling is performed and only the remedy packets that are inherent overhead produced by the social routing algorithm are exploited to avoid additional signaling while still allowing ISNC. Let us note that, doing so, not all coding opportunities can be leveraged, but there is a trade-off between the additional complexity and the ISNC gain. In this work, we have chosen the latter approach (ii) for the sake of simplicity. We get back to the discussion on the relevance of this choice at the end of the next chapter.

In this work, the social routing algorithm considered is SimBet [4]. This choice is due to the detailed utility components at play in SimBet, that allow for a modular usage when later designing decentralized coding criterion in the next chapter. The work is structured as follows:

Step 1 We consider a synthetic trace corresponding to the butterfly topology, and two sessions, as depicted in Figure 4.10. We set that the only node allowed to mix packets is the hub node, and it does so as soon as it can. The buffer management is discussed next in Section 4.2.3.1. We show that with such a systematic coding at the hub node in the butterfly topology operated with SimBet, ISNC outperforms single routing (or intra-session NC, but messages here are made of a single packet).

Step 2 We then consider the same framework, except that we devise a decentralized coding criterion. It allows each node that is presented with a coding opportunity to decide, based only on the local information it has access to, whether to code or not. We verify that within a fully decentralized implementation, the gains obtained in Step 1 can be gotten back on the butterfly topology.

Step 3 The so-designed ISNC protocol is tested on real-world traces so as to compare its performance with social routing algorithms, in terms of various metrics, in particular successful deliveries and supported load.

Steps 2 and 3 are the topic of the next chapter, while below Step 1 is developed in details.

4.2.3 The case of 2 single crossed sessions

4.2.3.1 Buffer and Copy Counter Management

The features described below apply symmetrically to both sessions: session-1 and session-2 with the source-destination pair (S_1, D_1) and (S_2, D_2), and packets P_1 and P_2, respectively. The packet resulting from the XOR between P_1 and P_2 is denoted by P_3. We consider in the rest of this thesis that P_3 can be generated at node B only if:

- B already holds P_2, and has no more room in its buffer,

- B meets A holding P_1 (or the other way around),
- the SimBet protocol (based on its own utilities as described in Section 1.2.2) triggers the transmission P_1 to B, if room was available at B,
- the buffer management below (and coding criterion in the next chapter) allow the replacement of P_2 at B by P_3

The copy counter assigned to P_3 is then the sum of the counter of the replaced packet and the copy budget handed over by A, determined by the copy share of SimBet ([4]).

We consider a buffer management which cannot favor ISNC compared with single routing and bias the results. It is detailed below.

F1 Destination node D_1 can erase P_1 from all nodes (in any community) but cannot erase P_2. A packet p_1 (ISNC or non-ISNC) can erase another packet p_2 in a community c only if p_1 is destined to c and p_2 is neither destined to c nor has its source in c. Also, P_1 can replace S_3 in the community of destination D_1.

F2 Keep on spreading the energy budget even though the payload of the already-there packets does not change. For example, when a node with P_1 meets a node with P_3 in any community, provided that the utility of later node is higher, the copies get transferred based on SimBet utility from the former node to the later one, although the payload remains the same. Similarly, the copies can spread when a node with P_2 meet a node with P_3. This feature, allowed by the use of ISNC, allows to re-focus the copy budget through coded packets to better serve both sessions. When the spray counter of a copy drops to zero, it is dropped.

F3 Destination node D_1: (i) erases P_1 from all nodes (in any community) but cannot erase P_2, (ii) erases P_3 upon reception from a node in community of D_1, and (iii) signals to the nodes of outside communities that it has received P_3 and/or recovered P_1.

Let us specify that the above items constitute buffer management choices aimed at fairly comparing ISNC with non-ISNC policies. We could also consider the most up-to-date scheme proposed by Krifa et al. in [78], and already described in the State-of-the-Art chapter as well as in the beginning of this chapter. In particular, the optimal drop policy they derive discriminates packets based especially on their number of copies and TTL, and for heterogeneous networks, it would require the joint distribution of the latter and inter-meeting intensities (i.e., communities). We can envision such a policy to replace F1, and hence apply both to no ISNC and ISNC policies. Owing to the lack of time, we did not implement this optimal drop policy, but we believe that, if implemented both for ISNC and non-ISNC, equivalent gains of the former should be obtained as with F1 (in particular in such a symmetric scenario with only 2 sessions).

It is worth noting that the last item F3 is not redundant or contradictory with item F1 which allows to replace p_2 by p_1 only in community c which is the destination community of p_1. The above item F3 extends it to other communities (amongst which the hub). We now explain how we implement signal in case of $F3$. Each packet can store a signal on the packet header. A packet header can have values in $\{0, \ldots, 3\}$:

0 : no destination of the pure packets (S_1, S_2) has received S_3, and/ or recovered its pure packet,

1 : only destination D_1 has received S_3, and/or recovered S_1,

2 : only destination D_2 has received S_3, and/or recovered S_2,

3 : both destinations have received S_3, and/or recovered pure packets.

Then, if this header field is

0 : the packet is not overwritten,

1 : the packet can be overwritten by S_2,

2 : the packet can be overwritten by S_1,

3 : the packet can be overwritten by both S_1 and S_2.

4.2.3.2 Simulation Settings

The exact topology on which the experiments are run is depicted in Figure 4.12. The trace is made of $N = 1001$ and $C = 5$ communities with $\beta_{i,j} = 5.10^{-4} x_{i,j}$, $(i, j) \in \{1, \ldots, C\}$, with the $x_{i,j}$ values as indicated in Figure 4.12. The buffer size is set to $B = 1$ packet. The bandwidth is set fixed to 1 packet in each direction upon each meeting. We consider the intuitive favorable case for ISNC, where the two session are crossed: session 1 (resp. 2) has source-destination pair (S_1, D_1) (resp. (S_2, D_2)) in communities 1 and 4 (resp. 2 and 3). The maximum number of copies per packet for both sessions varies as specified in the figures. The number of nodes in each community is $N_c = 250$, $\forall c \in \{1, \ldots, C\}$. Community 5 is the single hub node where a coded packet P_3 is allowed to be created if the hub node already holds P_2 and meets a node with P_1, or vice-versa. The simulations are performed thanks to a discrete event simulator written in Matlab. We use 15% of the simulation duration as warm-up phase to provide an opportunity to gather information about the nodes within the network, as in [4]. After the warm-up phase, the messages are allowed to disseminate in the network. The simulation results are averaged over 30 runs and the 5% confidence intervals are plotted.

4.2.3.3 Simulation Results

We plot in Figure 4.13 the objective function defined in the previous chapter (we recall that $P_1(\tau)$ and $P_2(\tau)$ are recovery probability of both session by time τ at their respective destinations). In order to assess the impact of each component F1, F2 and F3, the results incrementally

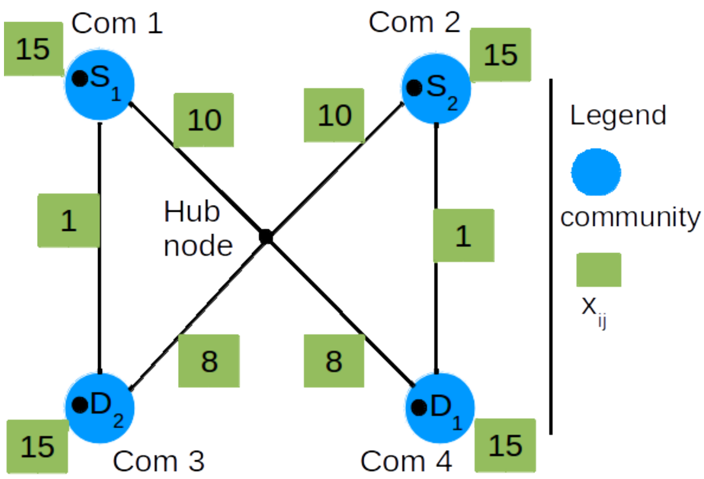

FIGURE 4.12: Butterfly topology considered for experiments.

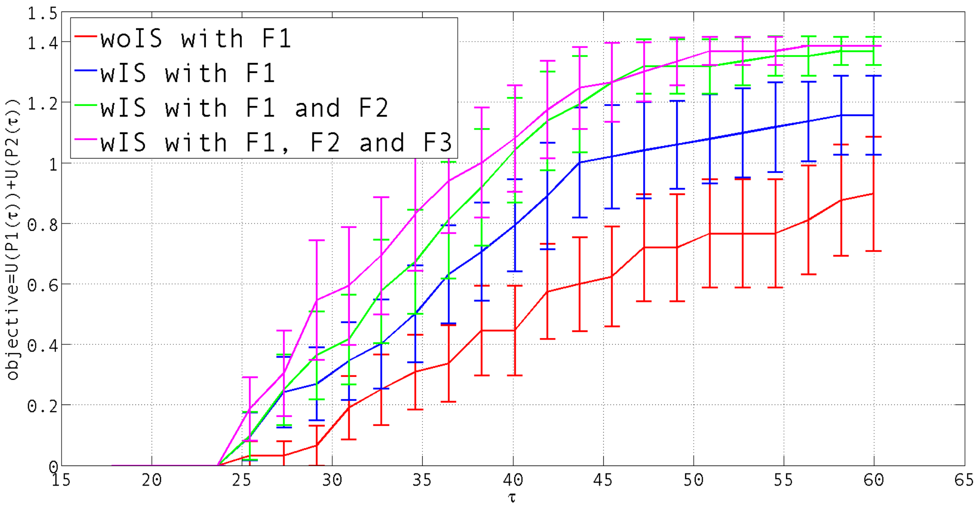

FIGURE 4.13: Simulation results for 2 crossed sessions.

adding each are shown. The curve label "woIS" referred to the case where SimBet routing alone is used, without session any mixing. In this case, only F1 can apply. We recall that here (unlike in the previous chapter) intra-session NC is not mentioned as messages are made of a single packet. The remaining 3 curves (labeled "wIS") show that without specific additional buffer or copy counter management, ISNC bring some gain, incrementally improved over F1, F2 and F3.

In order to ensure the origin of the gain brought by ISNC, we analyze the routes taken by the packets. Figure 4.14(a) depicts the evolution of the number of packets of each time in each community. Both sessions are fully symmetric, so is the network, hence we comment hereafter only for session 1. Packet P_1 (solid line) first spreads inside its source community 1, then reaches its destination community 4 through the mostly through the hub node as we see that the increase in P_1-infected nodes in $C = 4$ precedes the increase in $C = 2$, while the hub nodes

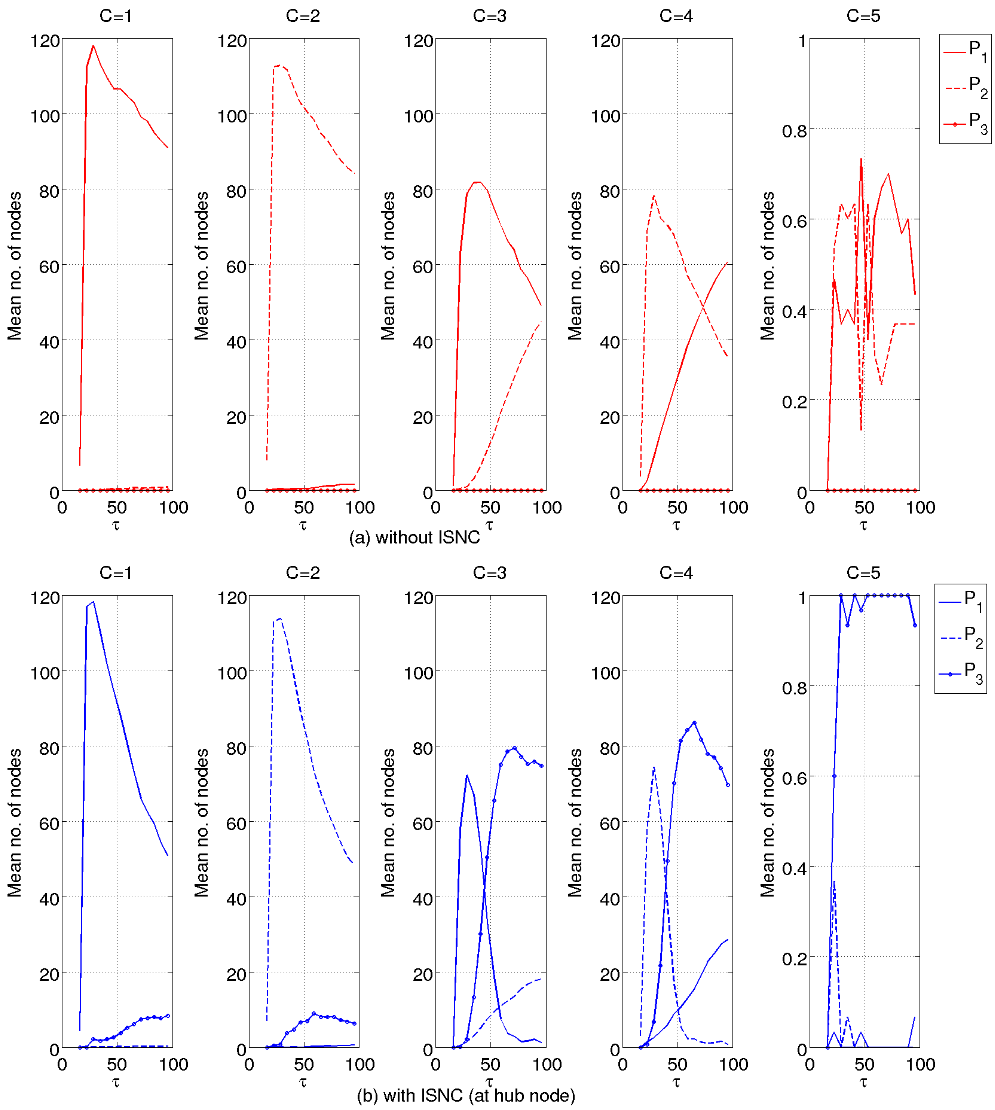

FIGURE 4.14: Mean number of nodes infected with each type of packets in each community.

gets readily infected. Community $C = 2$ remains almost uninfected by P_1. This shows that the routes taken by P_1 and governed by SimBet routing are very close to those in Figure 4.11(b), identified as the routes susceptible to benefit from ISNC, thereby explaining the gain with ISNC. In particular, in Figure 4.14(b), we can see that the hub gets occupied by a coded packet (P_3) systematically. This hub node is indeed the point of congestion, as it is at crossroads and has buffer size of 1. ISNC then gets beneficial in this setting.

Furthermore, these routes are decided by the three utilities components at play in SimBet, namely similarity, betweenness and tie-strength. Figure 4.15 depicts the evolution of each of these per community. As described in Section 1.2.2, a node hands over or copy the packet to another one based on comparison of the combination of these utility components. All the nodes in the same community converge to the same utilities values. Based on the utlities definitions and the knowledge of the network topology, we can predict the steady values of the utilities:

- Similarity between two nodes: number of common neighbors (met at least once).

$$Sim_x(D_1) = \begin{cases} 1, \text{if } x \in \text{ com } 1 \ , \\ 1, \text{if } x \in \text{ com } 3 \ , \\ N_2 + N_4, \text{if } x \in \text{ com } 2 \ , \\ N_2 + N_4, \text{if } x \in \text{ com } 4 \ , \\ N_2 + N_4, \text{if } x = \text{ hub / com } 5 \ . \end{cases}$$

- Betweenness of a node: number of pairs of nodes for which it is on the shortest path.

$$Bet_x = \begin{cases} (N_1 + N_3)(N_2 + N_4), \text{if } x = \text{ hub / com } 5 \ , \\ 0, \text{otherwise} \ . \end{cases}$$

- Tie-strength between two nodes: combination of meeting frequency (given by $\beta_{i,j}$), duration and recency. Its exact computation can be performed with our simple multi-community model.

$$TS_x(D_1) = \begin{cases} 0, \text{if } x \in \text{ com } 1 \ , \\ 0, \text{if } x \in \text{ com } 3 \ , \\ > 0, \text{otherwise} \ . \end{cases}$$

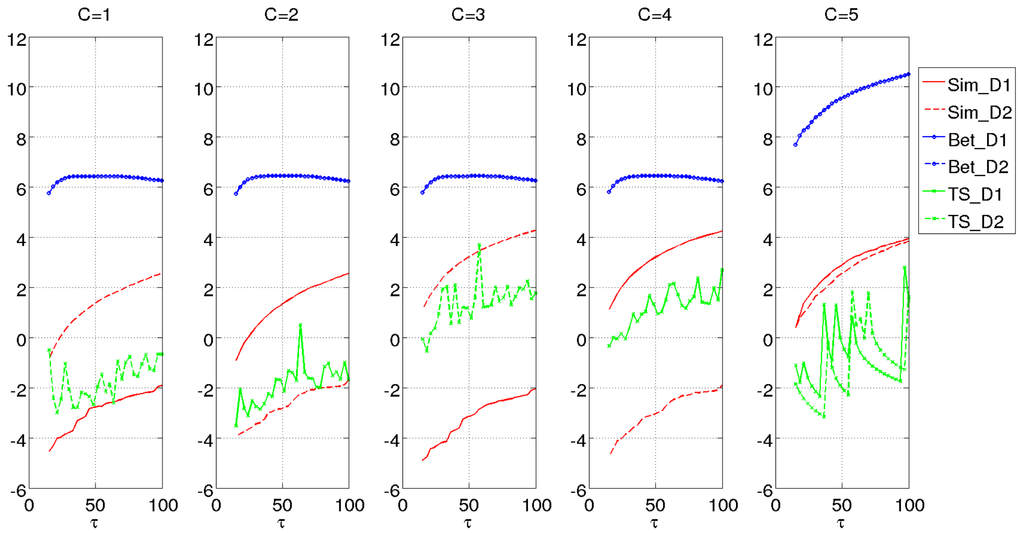

FIGURE 4.15: Similarity, Betweenness and Tie-strength utilities averaged over each community, plotted in log scale.

We now analyze how the ISNC gain varies with the energy/memory constraint, that is with the maximum number of copies per packet M. Figures 4.16 and 4.17 depicts the performance in terms of different metrics, to better compare thereafter with the results presented in the original SimBet paper [4]. They still show the performance of two crossed sessions, but with $M = 200$ and $M = 40$, respectively.

When there is no ISNC, delay for successful delivery of a pure packet is the time when destination first receives the pure packet while with ISNC, delay for the successful delivery of a packet is the time when a pure packet can be recovered either receiving directly the pure packet or by receiving coded packet with its remedy packet. When the successful delivery is via second case, delay is calculated as the maximum of the time when both coded and remedy packets are received. Average number of hops per message represent the average number of hops a message must take in order to reach the destination and for the delivery of a packet with coding, it is calculated as the maximum of the number of hops taken by coded packet and its remedy packet.

We observe in the upper-left Figure 4.16 the performance corresponding to that of Figure 4.13. For lower M, the gain of ISNC decreases, owing to the lower probability for the destination node to get the remedy packet. Hence, when desiging a decentralized coding criterion in the next chapter, we will need to pay specific attention to the constraint on M to decide whether to trigger session mixing.

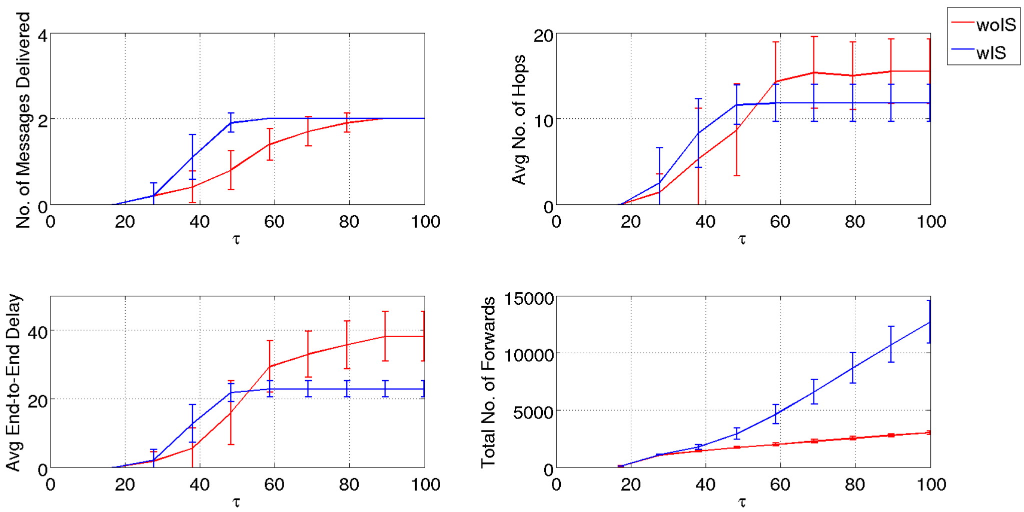

FIGURE 4.16: 2 crossed sessions, $M = 200$

4.2.4 The case of any number of concurrent sessions

We now analyze the impact of the network load on ISNC gain. In this section, the network load is considered to be the number of concurrent sessions. Let us first present the direct extensions of the above results, where the number of crossed sessions is not 2 anymore but 4 now. Figures 4.18 and 4.19 show that even for higher M, the ISNC gain vanishes as the number of sessions increases, owing to the limited buffer space per session that again impedes the remedy packet to make it early enough to the destination node. As explained below, we still allow only pairwise coding, meaning that a coded packet mixes at most two sessions.

We consider that there are multiple concurrent sessions running in the network. If the node buffers can store only one packet, then each time a node has a coding opportunity, it is between

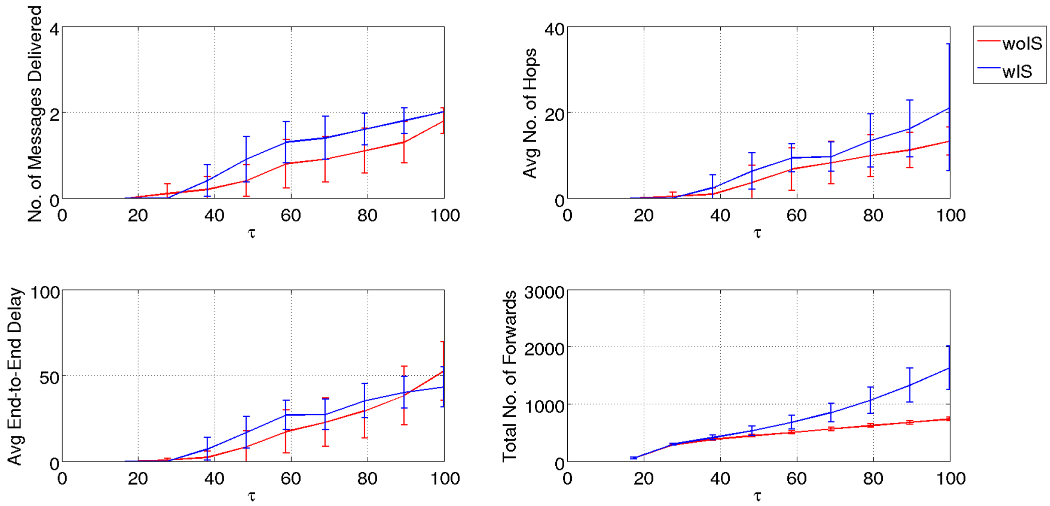

FIGURE 4.17: 2 crossed sessions, $M = 40$

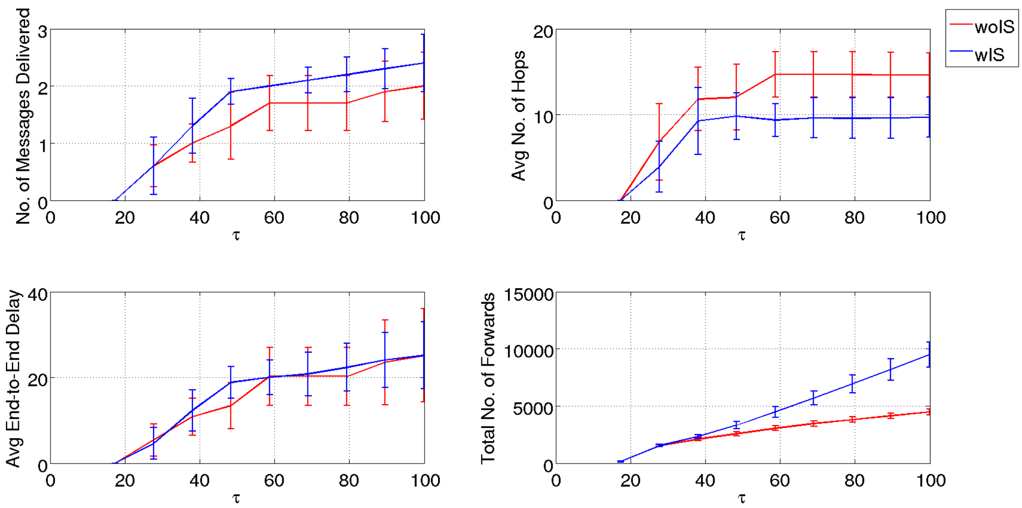

FIGURE 4.18: 4 crossed sessions, $M = 200$

only 2 sessions (that whose it already holds the packet, and that it is meeting the packet). We still consider that each message of each session consists of a single packet. If there are more than 2 sessions running in the network, then a node with buffer size 1 does not have to choose between what session to code. We present how to handle the case of larger buffer and hence wider coding choices, in the next chapter. We forbid that not more than 2 sessions be coded together.

4.2.4.1 Buffer Management

When the number of sessions is increased beyond 2 or 4, to improve the probability of delivery success of all messages, we allow the packets to be dropped with additional feature. In

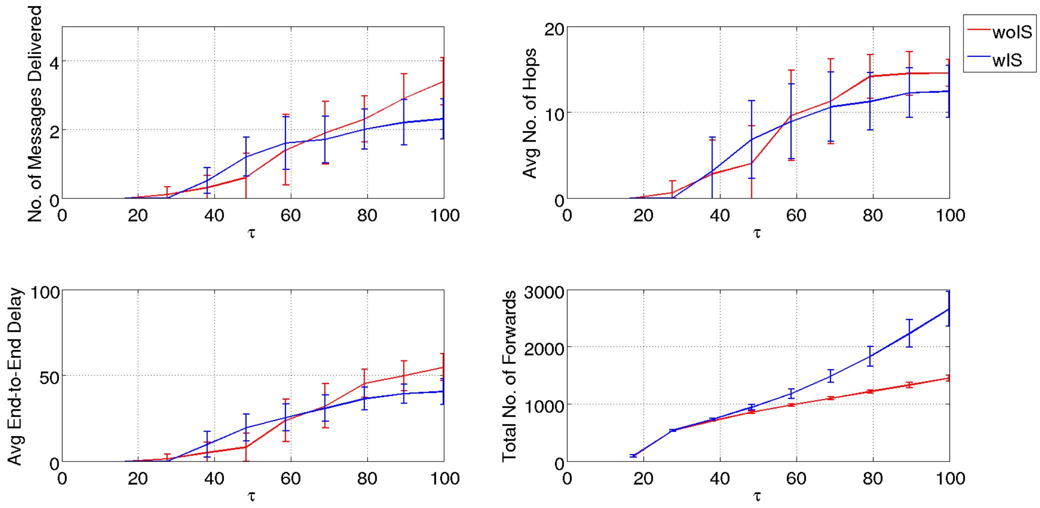

FIGURE 4.19: 4 crossed sessions, $M = 40$

addition to the buffer management policies F1-F3 explained in Section 4.2.3.1, we also implement policy F4 which we explain below:

F4 When two nodes I, with packet A, and J with packet B meet and have full buffers, then if A (resp. B) is in its destination community and B is not (resp. A), then we do not take any action, otherwise we pick up a random variable X such that

$$Pr(X = 0) = \frac{\alpha * n_A}{n_A + n_B},$$

$$Pr(X = 1) = \frac{\alpha * n_B}{n_A + n_B},$$

$$Pr(X = 2) = 1 - \alpha.$$

where n_A and n_B are the copy counters of packets A and B respectively, α is constant such that $\alpha : [0, 1]$. Then, we take following action based on different values of X:

$X = 0$: A overwrites B, while B cannot overwrite A,

$X = 1$: B overwrites A, while A cannot overwrite B,

$X = 2$: there is no exchange of packets and no overwrite is performed.

Furthermore, each message has a Time-To-Live (TTL) value, after which the message is dropped to allow new messages that arrive at a node to occupy the buffer space. If $L_n >= T\_start + V$, drop the packet P_1 from node's buffer, where T_start is the time when a copy of packet P_1 is received, V is TTL value and L_n is the current time when a decision is to be taken if the packet is to be dropped or not. We can select the value of V in following 3 different ways:

1. Constant: TTL is considered as a constant value; $V = V_0$.

2. Exponential: TTL is exponentially distributed with some constant mean, $meanV$ such that $V = exprnd(meanV)$ where $meanV = V_0$.

3. Exponential and dependent on copy counter: TTL is exponentially distributed with some mean value that is dependent on the number of copies yet to spread, $current\_counter$ such that $meanV = \frac{V_0 * current\_counter}{M}$.

4.2.4.2 Simulation Results

We consider the network topology of Figure 4.12 and the simulation setup is same as we presented for 2 sessions above in Section 4.2.3.2 with buffer size $B = 1$ and $Bw = 1$ but now we increase the number of sessions taking $N_S = 10$. The additional overwrite policy F4 is implemented with $\alpha = 0.8$ and "Exponential" drop policy with mean $V_0 = 100$. We obtain results for two sets: i) cross-sessions: first half of the 10 sessions has source-destination pairs in the community $(1, 4)$ while the remaining half in community $(2, 3)$, and ii) homogeneous sessions: source-destination pairs for any packet can be from any of the communities, which is closer to the real world scenario.

Figures 4.20 and 4.21 plot the simulation results for 10 crossed and 10 homogeneous sessions, respectively. We observe that ISNC performs much worse than no coding in the first case, while the degradation dwindle when the sessions are chosen uniformly at random over the communities.

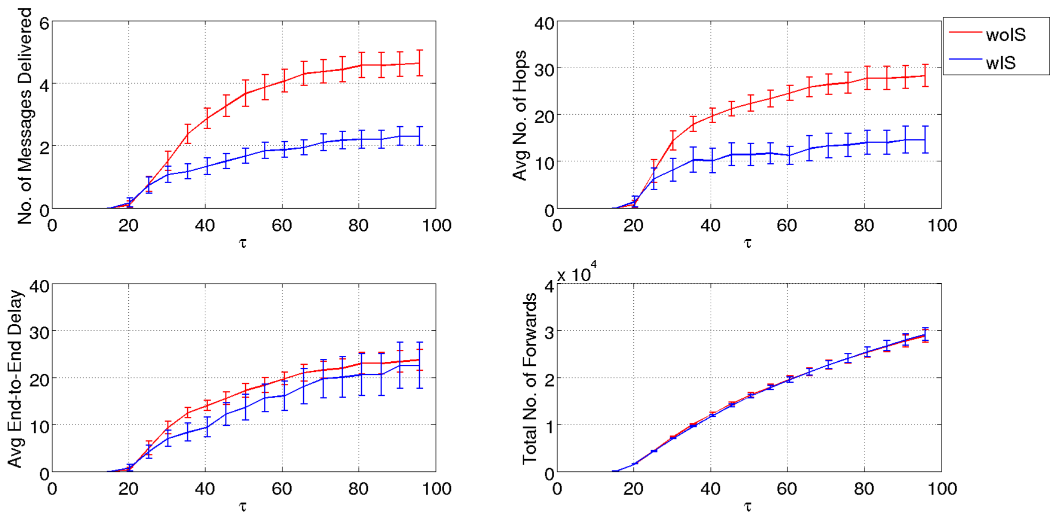

FIGURE 4.20: Simulation results for 10 crossed sessions, $M = 200$.

4.3 Conclusion

In this chapter, we have analyzed the performance of ISNC policy by carrying out experiments on a number of toy topologies such as network topology with 2 communities, chain

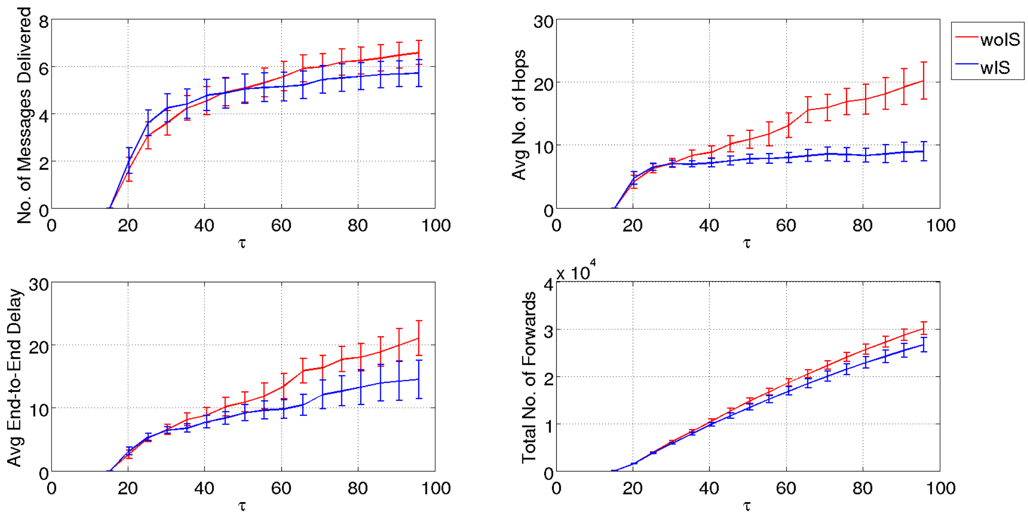

FIGURE 4.21: Simulation results for 10 homogeneous sessions (source-destinations pairs picked up uniformly at random in the different communities), $M = 200$.

topologies and butterfly topology, to identify when ISNC can be advantageous or detrimental to mobile social DTN. We have studied the impact of various parameters such as the maximum number of copies per packet, the network load, the buffer size and the buffer management policies. We conclude that when ISNC is implemented with social routing protocols such as SimBet, it can be beneficial. In the next chapter, we design decentralized ISNC coding criteria where coded packets are generated without the knowledge of network topology but with only local information available in the network.

Chapter 5

Designing online ISNC policies for DTMSN

In this chapter, we address the problem of deciding whether to trigger ISNC, if so, where and when in the network, that is mixing sessions based only on local information gathered at the nodes. We present our approach towards this goal, presenting the incrementally improved designed criteria.

5.1 Preamble: Transmission protocol with several concurrent sessions

We consider a general framework where there are multiple concurrent sessions running in the network. Let $Z_n = (S_n, D_n)$ denote the source-destination pair of session n. The node generates a coded packet if a certain condition $Q(Z_1, Z_2) >$threshold is satisfied. The expressions of $Q(.)$ and threshold are the subject of the next section.

As in the previous chapters, the sending node schedules the packets to send out in the decreasing order of their respective copy counters.

At the receiving node, if no room is left, then the received packet of session Z_A is checked for generating a coded packet by mixing it with an already present packet of session Z_B, where Z_B is chosen such that it maximizes $Q(Z_A, Z_B)$ over all Z_B present at B.

In Appendix B, we detail the ISNC transmission algorithm used in the multi-session case with arbitrary buffer size, considering that a node A may send packets to a node B that is it meeting.

5.2 Design of the decentralized coding criteria

5.2.1 General principle

We consider the decision problem of mixing session $Z_1 = (S_1, D_1)$ with session $Z_2 = (S_2, D_2)$ at node n_c. As before, P_1, P_2 and P_3 denote the session packets and the XoR of both, respectively. It can be ensured that ISNC does not perform worse than no ISNC only if:

$$\begin{cases} \text{Delay for } D_1 \text{ to get } P_2 < \text{Delay for } n_c \text{ to get } P_1 \text{ and } P_2 + \text{Delay for } P_3 \text{ to reach } D_1 \text{ from } n_c \\ \text{AND} \\ \text{Delay for } D_2 \text{ to get } P_1 < \text{Delay for } n_c \text{ to get } P_1 \text{ and } P_2 + \text{Delay for } P_3 \text{ to reach } D_2 \text{ from } n_c. \end{cases} \quad (5.1)$$

which corresponds to coding only if each remedy packet could make it to both destinations by the time the coded packet arrives. Another necessary condition for ISNC not to perform worse than sheer routing is to ensure that the coded packet does not reach the destination slower than a non-coded packet. However, we do not explicitly include this constraint in the coding criterion because we consider it is ensured by the chosen buffer management. We abbreviate the above quantities by:

$$\begin{cases} Delay(S_2 \to D_1) < Delay(S_1, S_2 \to n_c) + Delay(n_c \to D_1) \\ \text{AND} \\ Delay(S_1 \to D_2) < Delay(S_1, S_2 \to n_c) + Delay(n_c \to D_2). \end{cases}$$

5.2.2 Approximation framework

In order to estimate each of the three quantities involved in the above inequalities, we make the following choices.

- Owing to the difficulty of its estimation, $Delay(n_c \to D_1)$ is removed (considered to be 0), thereby constraining more the coding criterion (and hence possibly skipping coding opportunities).
- $Delay(S_1, S_2 \to n_c) = \max\big(Delay(S_1 \to n_c), Delay(S_2 \to n_c)\big)$, where the two arguments of the maximum are estimated by time counters kept in the packet headers.
- The estimation of $Delay(S_2 \to D_1)$ is done in different ways leading to the different coding criteria of the next section. The estimation is based only on local information gathered at the nodes, and takes into account different parameters.

Owing to the results of the previous chapter, we consider as routing policy SimBet, described in chapter State-of-the-Art, and make use of the utility components defined therein to approximate to quantity discussed in the last above item.

5.2.3 Estimation of quantities

5.2.3.1 Criterion I

$Delay(S_2 \to D_1)$ is the time for S_2 to hit destination node D_1. It is estimated with:

$$Delay(S_2 \to D_1) = \frac{1}{SimTS(S_2 \to D_1)\frac{U(D_1 \to D_2)}{U(D_1 \to D_2)+U(S_2 \to D_2)}}$$

where $SimTS(S_2 \to D_1)$ is meant to approximate the meeting frequency (at most 2 hops in communities) of S_2 with D_1, and $\frac{U(D_1 \to D_2)}{U(D_1 \to D_2)+U(S_2 \to D_2)}$ accounts for the number of P_2 copies disseminated in community of D_1. The above quantities are expressed thanks to the SimBet utility components: $SimTS(S_2 \to D_1) = Sim_{S_2}(D_1) + TieStrength_{S_2}(D_1)$ and $U(D_1 \to D_2) = Sim_{D_1}(D_2) + Bet_{D_1} + TieStrength_{D_1}(D_2)$ with the metrics Sim, Bet, $TieStrength$ as explained in Chapter 2. We obtain $Delay(S_1 \to D_2)$ similarly.

The impact of Criterion I on signaling as compared with SimBet is that each node needs to know the $SimTS$ component of all other pairs of nodes. So upon meeting, in the same way the ego matrix of size $N \times N$ is exchanged ([4], N is the number of network nodes), N^2 more values are exchanged. However, if the nodes are able to maintain a community-structure map (based, e.g., on the employed Louvain decentralized community detection algorithm [106]), then the number of values may only be C^2 (considering homogeneous communities). Also, the way to update the value of these exchanged components can be merely with some (exponentially) weighted moving average. Addressing properly these issues by testing the resilience of our algorithm to the online estimation could not been done in this thesis, where we have resorted for the tests below to a centralized knowledge kept over time in the simulations. We discuss more on that in the next section presenting Criterion II.

Figures 5.1, 5.2 and 5.3 show the performance of this criterion in comparison with systematic ISNC at hub node (studied in the previous chapter) and no ISNC. In the simplistic case of 2 single crossed session in figure 5.1, we verify that Criterion I indeed achieves a tradeoff between no ISNC and systematic ISNC at hub. More detailedly, figure 5.4 shows that initially the number of nodes with coded packet in hub community is less in case of decentralized wIS but with time there are as many coded packets as compared to centralized case.

For 10 crossed sessions then, Figure 5.2 details the performance in terms of the 4 metrics detailed in the previous chapter, Section 4.2.3.3. This case is where systematic coding performed bad, because we have identified in the previous chapter that it codes although the remedy packet is not likely to hit the destination node early enough. We observe that Criterion I is not able to perform as well as no ISNC is this case (meaning that it should disable ISNC, but does not completely), though it still performs better than systematic ISNC. Secondary, we also observe that Criterion I achieves a performance tradefoff in terms of number of messages delivered by some deadline, but allows a lower number of hops and no higher number of forwards, thereby saving node occupancy. Finally, Figure 5.3 shows that Criterion I performs roughly the same as systematic ISNC for 10 homogeneous sessions, allowing the same performance as no ISNC, with a much lower number of hops.

Despite the relative good results of this criterion shown in Figure 5.3, this criterion has two major problems. First, it does not perform as well as no ISNC in the case of 10 crossed sessions. Even though this case might seem intense, it however underlies a weakness of Criterion I as a strong constraint to the design of ISNC policies is to ensure that they do not degrade the performance as compared to routing, and ensure at least the same performance if not able to always bring some gain. The second problem in Criterion I is that the approximation of the meeting frequency of S_2 with D_1 by $SimTS(S_2 \to D_1)$ is too rough as, given the definition of the Sim and TS components, the $SimTS$ quantity does not scale as $\beta_{C_{S_2}C_{D_1}}$ or a combination of the β_{ij} that would lead to an estimation of the mean delay to travel from S_2 to D_1. That is why the next section investigates another criterion to address these issues.

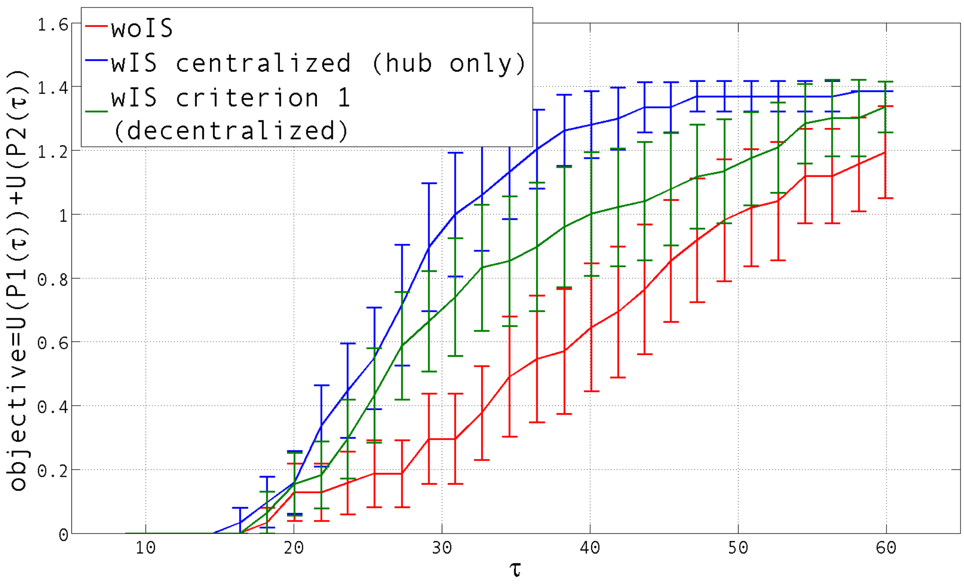

FIGURE 5.1: Simulation results for 2 crossed sessions, criterion I.

5.2.3.2 Criterion II

Having the right-hand side terms of condition B.1 given by time counters stored in the packet headers, the key problem is still the estimation of the left-hand side term $Delay(S_1 \to D_2)$. As the dissemination within a given community (say, the source community), is epidemic up to a certain point with SimBet (similar to Spray-and-Wait), we build on the result of [93] that expressed the mean packet delay under epidemic routing: for a homogeneous DTN with N nodes and meeting intensity β, the mean delay for a packet to hit a certain node is

$$\frac{\log(N)}{\beta N} . \tag{5.2}$$

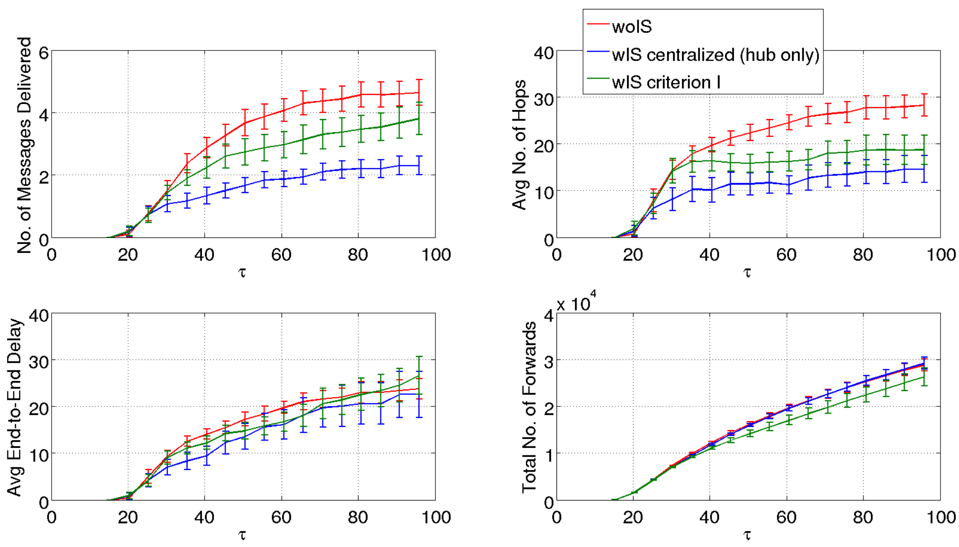

FIGURE 5.2: Simulation results for 10 crossed sessions, criterion I.

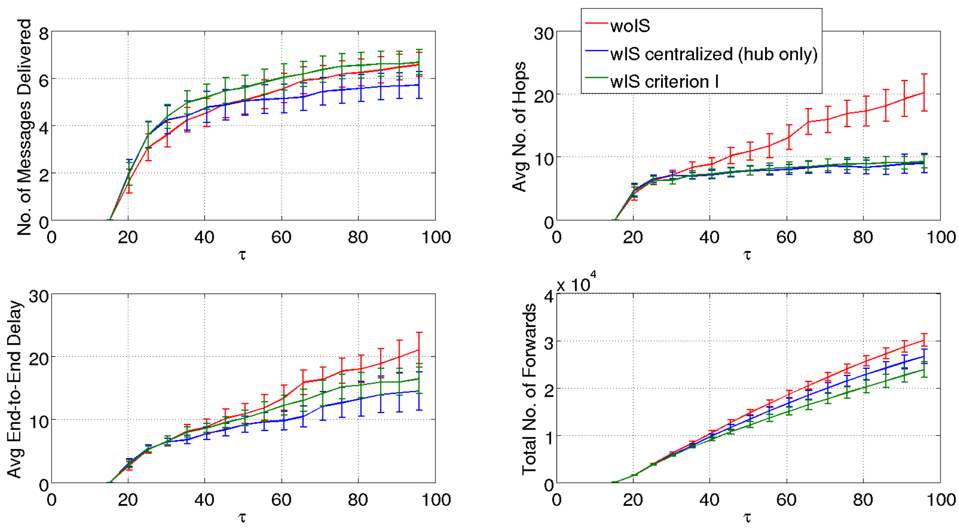

FIGURE 5.3: Simulation results for 10 homogeneous sessions, criterion I.

So we approximate as follows the quantity of interest:

$$Delay(S_1 \to D_2) = \frac{\log(N_{eff})}{\beta N_{eff}}$$

with the rationale of considering the dissemination governed by the available resources:

- N_{eff} stands for the number of effective nodes taking part into the spraying of packet P_2. We consider

$$N_{eff} = \frac{U(D_2 \to D_1)}{U(D_2 \to D_1) + U(S_1 \to D_1)} M \min(1, \frac{BN_{total}}{MN_s})$$

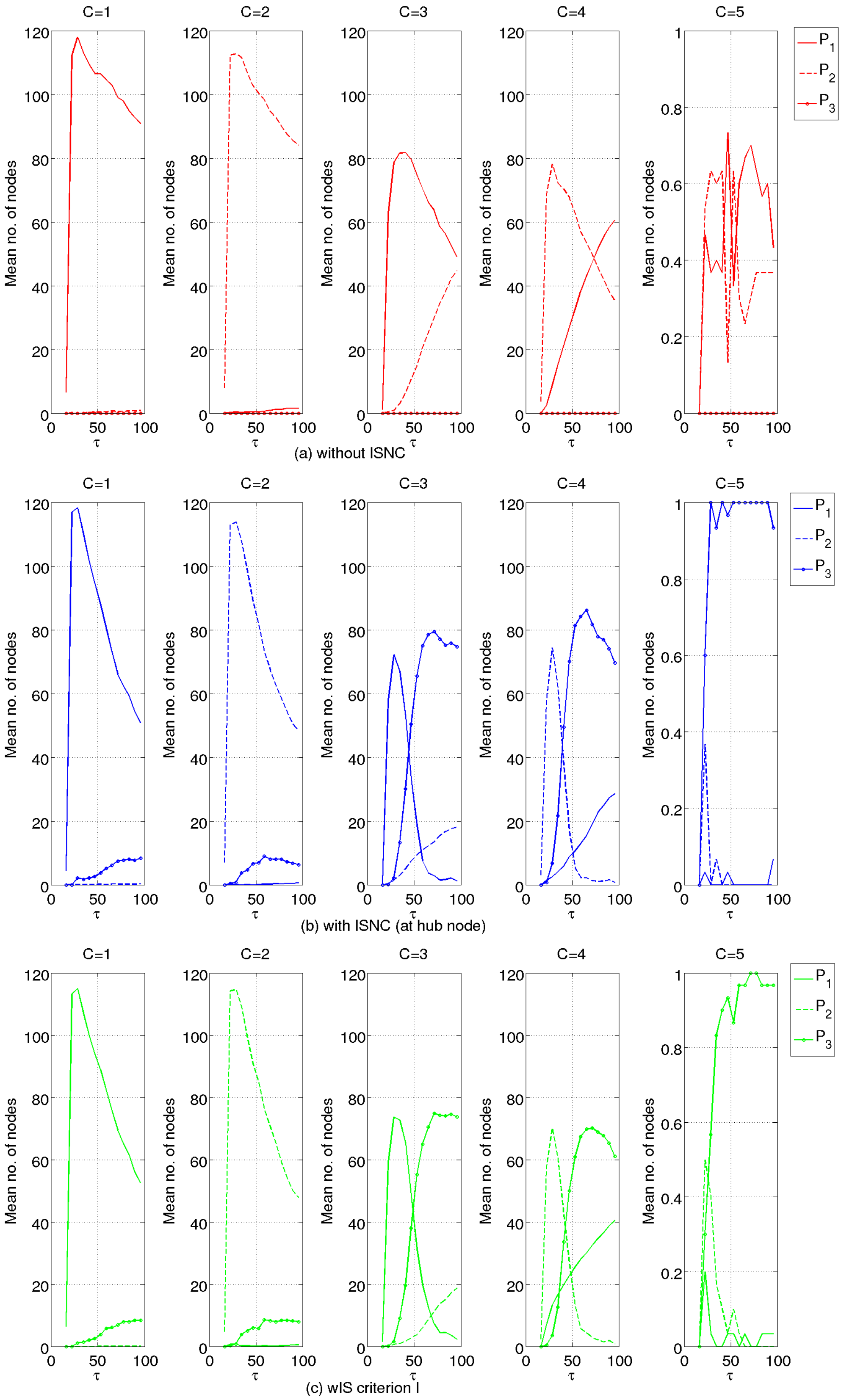

FIGURE 5.4: Mean number of nodes infected with each type of packets in each community.

where, over the total possible M of P_2-infected nodes, only a fraction $\frac{U(D_2 \to D_1)}{U(D_2 \to D_1)+U(S_1 \to D_1)}$ is actually created in the community of D_2. Then N_s stands for the number of running sessions in the network, so the last $\min$ term represents the fraction of buffer available at the nodes, given the number of copies of each message, the number of sessions (i.e., messages), and the total number of nodes in the network.

- To substitute the β parameter of equation 5.2, our first rough attempt is with $\beta = \frac{\beta_{11}+\beta_{13}}{2}$, with β_{11} the intra-meeting intensity inside community of S_1, and β_{13} the inter-meeting intensity between community of S_1 and that of D_2.

The above criterion still needs to be refined, but unfortunately time did not permit to. Indeed, the sheer fact to consider the same delay expression for homogeneous and heterogeneous network is loose. This especially shows through the definition of β above. Considering a network model to get this estimation while not knowing the outcome of the coding criterion is a research direction, so is taking into account the existing works such as that of Picu *et al.* [56, 69] We have briefly considered the possibility to use the model of Chapter 2, that is that of [18] but could not determine yet how to implement and estimate the outcome of the model online by a node.

Yet, below we assess numerically the performance of this criterion, because it suggests that it is the right way to keep investigating. Figure 5.5 shows that in the case where Criterion I performed bad, with 10 crossed sessions, Criterion II is able to achieve closer to the case of no ISNC, while still maintaining the benefits of ISNC in the case of homogeneous sessions of Figure 5.6.

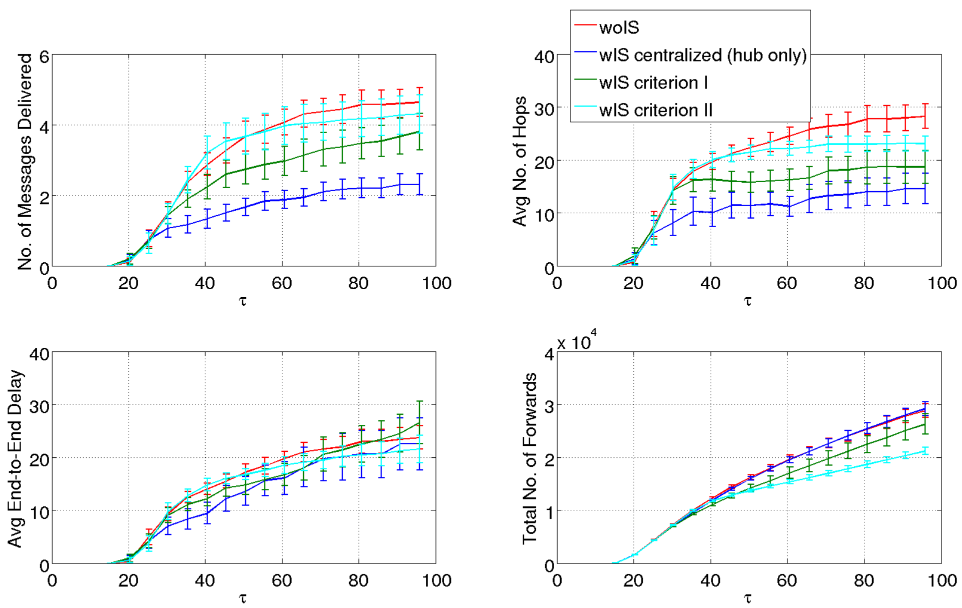

FIGURE 5.5: Simulation results for 10 cross sessions.

packets

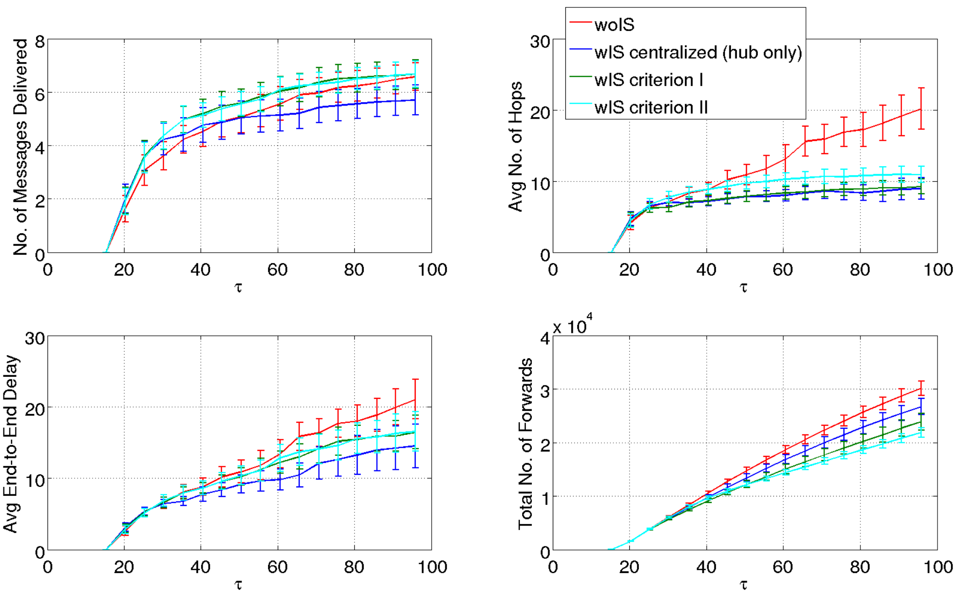

FIGURE 5.6: Simulation results for 10 homogeneous sessions.

5.2.4 Real-world traces and comparison on several metrics

5.2.4.1 Experimental datasets

We now test our decentralized ISNC coding criteria on some real-world traces. We present in Table 5.1, brief summary of the datasets collected from CRAWDAD [5]. These traces have already been used to test the routing protocols in DTN such as SimBet [4].

	Cambridge	Infocom05
participants	12	41
duration (days)	5	3
device	iMote	iMote
mobility patterns	Bluetooth contacts	Bluetooth contacts
scanning interval (seconds)	120	120
total devices	223	264

TABLE 5.1: Characteristics of experimental datasets [4, 5, 6].

In **Cambridge** trace, the imotes were distributed to doctoral students and faculty comprising a research group at the University of Cambridge Computer Laboratory [4, 5]. The experiment lasted for five days and a total of 223 devices, including the participants, were encountered during the experiment. The **Infocom05** dataset was collected from experiment conducted during the IEEE INFOCOM 2005 conference in Miami where 41 imotes equipped with Bluetooth were distributed among conference attendees [4, 5, 6]. The participants were chosen in order to represent a range of different groups belonging to different organizations and were asked to carry the devices with them for the duration of conference.

5.2.4.2 Community detection

Before using these traces to test decentralized ISNC coding criteria, we analyze the mobility pattern in these datasets. There are many community detection methods proposed and examined in the literature e.g., weighted network analysis (WNA) by Newman [23, 107], and Louvain [106] community detection algorithms. We present below WNA community detection algorithm that we use to examine the traces.

Weighted Network Analysis: WNA can work on weighted graphs directly though it cannot detect overlapping communities like in K-CLIQUE [108]. The modularity of a partition is a scalar value between -1 and 1 that measures the density of links inside communities as compared to links between communities. One can compute the modularity value for each community partitioning of a network in WNA using following definition of $modularity(Q)$:

$$Q = \sum_{vw} [\frac{A_{vw}}{2m} - \frac{K_v k_w}{(2m)^2}] \delta(c_v, c_w).$$

where, A_{vw} is the value of the weight of the edge between vertices v and w, if such an edge exists, and 0 otherwise; the δ-function $\delta(i, j)$ is 1 if $i = j$ and 0 otherwise; $m = \frac{1}{2} \sum_{vw} A_{vw}$; k_v is the degree of vertex v defined as $\sum_w A_{vw}$; and c_i denotes the community of which vertex i belongs to. The term in the formula $\frac{\sum_{vw} A_{vw}}{2m} \delta(c_v, c_w)$ is therefore equal to $\frac{\sum_{vw} A_{vw} \delta(c_v, c_w)}{\sum_{vw} A_{vw}}$, and this represents the fraction of the edges that fall within communities. The difference between this fraction and, the fraction of the edges that would be expected to fall within the communities if the edges were assigned randomly but keeping the degrees of the vertices unchanged is called $modularity$ [23]. This algorithm is a genetic algorithm, using the modularity as the measurement of fitness. It enumerates all possible merges of any two communities in the current solution, evaluates the relative fitness of the resulting merges, and chooses the best solution as the seed for the next iteration. Table 5.2 shows the number of communities detected from the traces on using WNA algorithm.

Trace/Model	Community detection method	No. of community	Q
Cambridge	WNA	3	0.28
Infocom05	WNA	8	0.1

TABLE 5.2: Number of communities and modularity (Q).

Then, we computed β defining the intra and inter-meeting intensities between the communities. We are able to identify that the butterfly topology pattern existed in the Infocom05 trace as shown in Figure 5.7.

$\beta_{Infocom05}(per\ hour) =$

$$\begin{bmatrix} 1.64 & 0.71 & 0.55 & 1.15 & 1.43 & 1.33 & 1.13 & 0.22 \\ 0.71 & 1.41 & 0.42 & 0.79 & 0.95 & 0.99 & 0.69 & 0.07 \\ 0.55 & 0.42 & 0.66 & 0.63 & 0.94 & 0.95 & 0.84 & 0.06 \\ 1.15 & 0.79 & 0.63 & 4.97 & 1.60 & 1.75 & 1.16 & 0.07 \\ 1.43 & 0.95 & 0.94 & 1.60 & 2.72 & 2.12 & 1.77 & 0.22 \\ 1.33 & 0.99 & 0.95 & 1.75 & 2.12 & 6.52 & 1.69 & 0.13 \\ 1.13 & 0.69 & 0.84 & 1.16 & 1.77 & 1.69 & 5.44 & 0.35 \\ 0.22 & 0.07 & 0.06 & 0.07 & 0.22 & 0.13 & 0.35 & 24.79 \end{bmatrix} \times 10^{-2} .$$

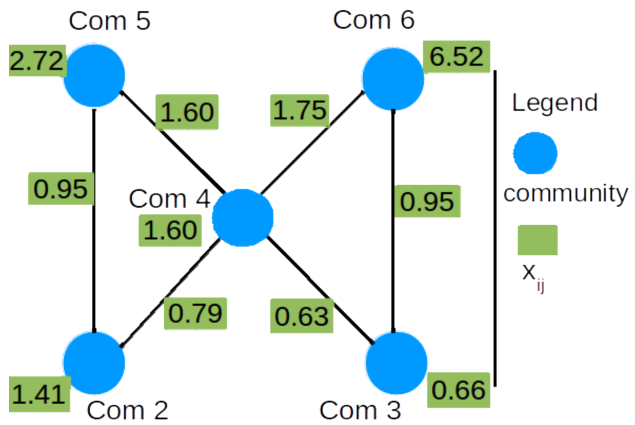

FIGURE 5.7: β value for identified butterfly topology inside Infocom05 trace.

5.2.4.3 Simulation results

Having identified that the heterogeneous mobility pattern and multi-community model exists in the real-world traces, we test our decentralized ISNC coding criteria on these: traces Cambridge and Infocom05.

We consider the buffer size is set to $B = 1$ packet, the bandwidth is set fixed to 1 packet in each direction upon each meeting, and the number of concurrent sessions to 50. The maximum number of copies per packet for each session is 40. The simulation results are averaged over 30 runs and the 5% confidence intervals are plotted.

We observe from Figures 5.8 and 5.9 that the online ISNC policy with SimBet routing perform as good as SimBet alone routing in Cambridge dataset while the large throughput gain is observed in Infocom05 dataset. We used the β value computed with the number of communities detected and frequency of meeting of the nodes as we discussed in above section for the decentralized ISNC coding criteria II. The heterogeneous mobility pattern, multi-community

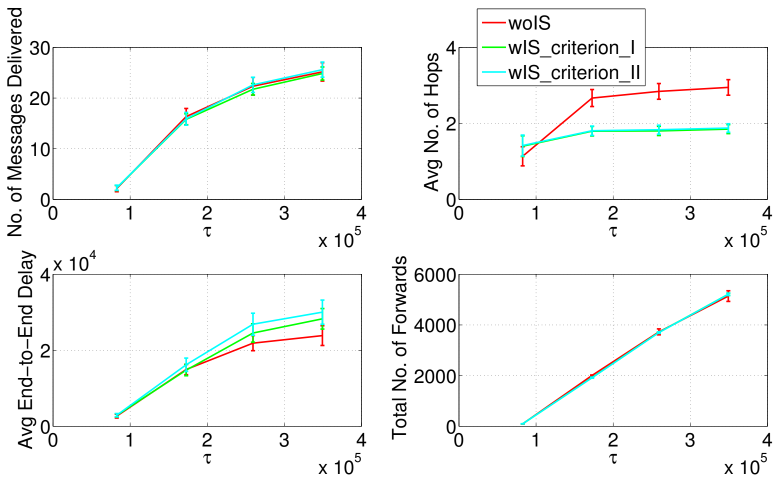

FIGURE 5.8: Simulation results for Cambridge dataset with 50 concurrent sessions, each with 40 copies.

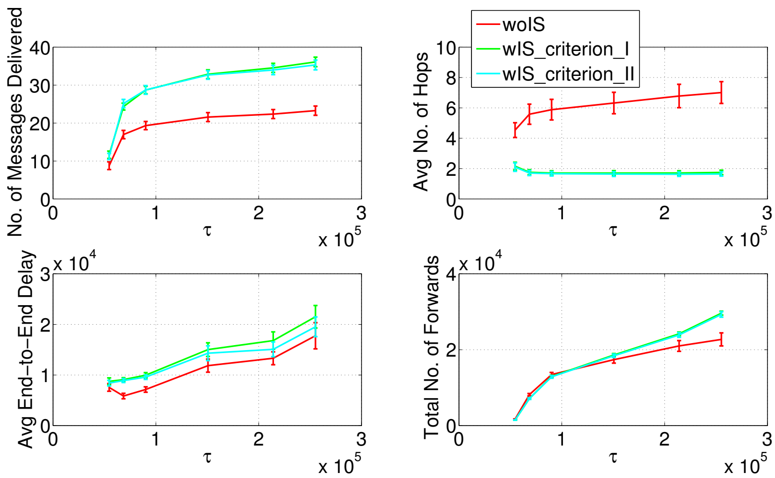

FIGURE 5.9: Simulation results for Infocom05 dataset with 50 concurrent sessions, each with 40 copies.

and possible detection of butterfly topology pattern in Infocom05 dataset contributes to better performance obtained with our decentralized ISNC criteria.

5.3 Choosing the coding criterion with an analytical model

Chapter 3 has presented a ISNC policy and an analytical model where the number of packets per session can be any, corresponding to the case where a file is split into several packets, and the metric (whether it be delay or delivery probability) is on the whole file. Also, 2 sessions are considered. In order to tackle the optimization problem of ISNC policy by reducing the parameter space and identifying meaningful/sound heuristics, we first analyzed in Chapter 4 the impact of the various parameters on the ISNC performance, and in the last sections, we have resorted to do so to a simplification of this first setting, by considering single-packet messages/file, and further added the load of the network as an impacting parameter.

In this section, we get back to the analytical model to show how it can be useful in the design of the decentralized coding criterion (i.e., ISNC policy) in the settings considered in this chapter. The model we present below assumes that two messages (each made of a single packet) are sent from source S_1 to D_1, and S_2 to D_2. The SimBet policy is used, we take as input the topology of the DTMSN (the corresponding multi-community model). By expressing the evolution of the numbers of the three possible types of packets ($P1$ for session-1, $P2$ for session-2 and $P3$ for mixed) depending on the coding criterion, we can then assess what possible gain this criterion brings with respect to no ISNC (w.r.t. the metrics that can be extracted from the model). The full knowledge of the topology makes consider the utilities involved in SimBet as known and fixed. Also, the coding criterion builds on this utilities so it boils down to a binary decision when implemented into the dissemination model.

Let $X_c(t)$, $Y_c(t)$ and $Z_c(t)$ be the number of messages of $P1$, $P2$ and $P3$ in community c, respectively, where $c = 1, \ldots, C$. We hereafter drop the time index for lighter notation. Let $m_{X,c}$ (resp. $m_{Y,c}$ and $m_{Z,c}$) be the mean counter of $P1$ (resp. $P2$ and $P3$) in community c (at time t). Then, still using a fluid model as detailed in Chapters 2 and 3,the dissemination process writes as:

$$\frac{dX_c}{dt} = \sum_{i=1}^{C} \beta_{ic} \Delta_X X_i (N_c - T_c)$$

where $T_c = X_c + Y_c + Z_c$ and

$$\Delta_X = \begin{cases} 1, \text{ if } \lfloor \frac{U_{c \to D_1}}{U_{i \to D_1} + U_{c \to D_1}} m_{X,i} \rfloor > 1 , \\ 0, \text{ otherwise} \end{cases}$$

and we approximate the counters as the total number of copies yet to spread by the number of already-there copies in community c:

$$m_{X,c} = \frac{M - \sum_{i=1}^{C} X_i - \frac{1}{2} \sum_{i=1}^{C} Z_i}{X_c} ,$$

$$m_{Y,c} = \frac{M - \sum_{i=1}^{C} Y_i - \frac{1}{2} \sum_{i=1}^{C} Z_i}{Y_c} ,$$

$$m_{Z,c} = \frac{M - \sum_{i=1}^{C} Z_i - \frac{1}{2} \sum_{i=1}^{C} (X_i + Y_i)}{Z_c} .$$

In the same way, we have:

$$\frac{dY_c}{dt} = \sum_{i=1}^{C} \beta_{ic} \Delta_Y Y_i (N_c - T_c)$$

with a similar expression for Δ_Y as for Δ_X. Then for the mixed messages, we have:

$$\frac{dZ_c}{dt} = \sum_{i=1}^{C} \beta_{ic} \delta_{ic} \left(\Delta_X X_i Y_c + \Delta_Y Y_i X_c \right)$$

$$+\sum_{i=1}^{C} \beta_{ic} \Delta_Z Z_i (N_c - T_c) \ ,$$

where δ_{ic} is the coding criterion being tested, expressed with quantities such as those in the above sections.

Solving numerically this fluid model enforcing the chosen decentralized coding criterion, and its non-ISNC counterpart, allows to easily check whether, on a given topology assuming perfect knowledge of the utilities given the topology, ISNC can be beneficial.

We emphasize that this model is a simplified version of that presented in Section 2.1, allowing only for a buffer size of 1, unlike the latter. Another limitation of these two models is that the only way to account for the load of the network is to modulate the available buffer size for the sessions of interest accordingly to the average number of concurrent sessions.

Let us now get back to the alternative Markov chain-based model named DTN-Meteo [69] and discussed in Chapter 2. One could think of using DTN-Meteo instead of the fluid model defined above, to help design the decentralized coding criterion. We would need to track two sessions (how to do it would need to be determined), the buffer having 3 possible states $P1$, $P2$ and $P3$. Then, implementing the utilities deterministically, as described above and in [69], as well as the coding criterion, we would get back the results of the fluid model, except that the full node heterogeneity could be considered (at the expense of the state explosion). The main challenge would still be to determine a good coding criterion, that is, from what has be presented in the last sections, a good locally-computed approximation to $Delay(S_1 \to D_2)$. Whether the Markov model of DTN-Meteo could be used to derive a decentralized optimization procedure of the coding criterion, in the same way as what has been done in [41], is a direction for future work. Another challenge is to properly take into account the network load leading to effective available buffer size to be better estimated as it is a key component of the coding criterion.

5.4 Conclusion

In this chapter, we have designed decentralized coding criteria allowing to trigger ISNC deciding when and which sessions to mix together in the presence of multiple concurrent sessions in the network, with our goal of designing online ISNC policy that can be beneficial for DTMSN. We also presented a simplified analytical model to tackle the optimization problem of ISNC policy explaining certain limitations of our model.

Conclusions and Perspectives

The goal of this thesis has been to devise efficient transfer policies combining inter-session network coding and socio-aware routing algorithms in Delay Tolerant Mobile Social Networks. To tackle the problem, we have taken both analytical and emprirical approaches. The analytical approach consisted in devising a parameterized control ISNC policy (with legacy Spray-and-Wait routing), and deriving its performance model thanks to a fluid approximation. The empirical approach investigated, on various toy topologies, when ISNC can be beneficial. We have found out that, although the contacts are considered bidirectionnal (and asynchronous), which usually prevents NC to be interesting, ISNC can be beneficial when coupled with a socio-aware routing algorithm (SimBet in our study) owing to the subset of the possible routes selected by the utility-based algorithm. We have then focused on designing an online ISNC policy for DTMSN, by devising a decentralized coding criterion able to trigger ISNC when it can bring some gain. We have also made explicit how the analytical model we developed can serve the design of such criterion.

Conclusions

In **Chapter 1**, we reviewed the works in literature that are closely related to the topic of this thesis. We studied routing protocols for homogeneous and heterogeneous DTN. We discussed about various models to predict performance of routing policies, and how buffer management policies impact on routing. Then, we reviewed works related to intra-session and inter-session NC schemes.

We have first addressed the problem of optimizing routing policies in mobile social DTN. Thanks to a mean-field approximation of the spreading process, we have formulated the problem of finding the optimal time-dependent policies under a given constraint of energy in **Chapter 2**. We have proven theoretically that the optimal policies are per-community threshold policies, thereby generalizing the existing works for homogeneous mobility DTN. We have provided analysis of this result on a numerical example, and discussed the distance to optimal of online utility-based polices of the literature.

Then in the second part of the thesis, we have devised a parameterized pairwise ISNC control policy and expressed the control optimization problem thanks to a performance model in **Chapter 3**. The scheme is at the same time a routing and a coding policy that allows to optimize for a utility function defined over the two sessions, under some energy constraint. Our policy decides which nodes can mix the sessions based on their communities rather than on their individual properties, making the devised policy scalable with the number of nodes if the number of communities keeps limited. We have derived its performance modeling thanks to a mean-field approximation leading to a fluid model of the dissemination process, and validated the model with numerical experiments. By showing numerical gains, we have illustrated the relevance of our approach that consists in designing ISNC control policies not reasoning on specific nodes but instead on the coarse-grained underlying community structure of the social network.

In order to tackle the optimization problem of ISNC policy, in **Chapter 4** we presented an experimental study of pairwise ISNC to investigate when and where it can be beneficial or detrimental. We tested ISNC policy on toy topologies, namely chain and butterfly topologies and examined the impact of a number of parameters, such as the constraint on the maximum number of copies per packet, the load of the network (i.e., the number of concurrent sessions), the buffer size of the relay nodes and the buffer management policies. We built on SimBet routing protocol an ISNC policy and showed that on the set of routes selected by the SimBet routing algorithm, ISNC can be beneficial to social DTN.

After proving the benefit of a centralized social-aware (community-based) pairwise ISNC policy to social DTN, we stepped towards the goal of designing online ISNC policies for DTMSN. In **Chapter 5**, we have designed decentralized coding criteria allowing to trigger ISNC if it may be beneficial based on local information available in the network. We tested these coding criteria on both toy topologies and real-world traces.

Perspectives

This work allowed progresses towards understanding what are the gains ISNC can bring in operating DTMSN, and how they can be achieved. A number of steps remain to be taken yet, in order to obtain a complete online ISNC policy, working along some socio-aware routing algorithms.

Immediate follow-up works are:

- First, it would be important to thoroughly assess the gain of ISNC in terms of fairness amongst the nodes. Indeed, how the nodes that are key hubs in the networks are loaded with others' traffic is an important concern for the sustainability of this kind of networks, and the very SimBet policy [4] or FOG [109] have specifically focused on this problem of trading off fairness and efficiency.

- Second, the design of a good coding criterion, that decides whether to mix sessions taking into account all the important parameters, amongst which the network load, is yet to be achieved. In particular, we have identified the key quantity to be estimated more accurately.

- Third, thorough assessments on real-world traces mus be carried out with such a coding criterion.

Longer-term extensions are:

- An important direction towards devising more efficient ISNC policies in DTMSN is the online optimization procedure of the coding criterion. In this thesis, we have derived analytical models, then endeavored to extract heuristics from experiments so as to design coding criteria which can then be tested a posteriori using the model. Another way to go would be to express the optimization problem in the simpler case of one packet per session (so a simpler formulation than that of Chapter 3), and see whether a greedy approach such as that of [41] could be used.

- This study have relied on our choice to opportunistically use, as remedy packets, those spread by the utility-based routing algorithm. However, we could identify that the timely opportunistic reception of these remedy packets gets more unlikely when the energy constraint or the network load get more stringent. Since the delivery of these remedy packets is crucial in ISNC performance, one can envision to purposely transfer remedy packets when ISNC is expected to increase performance. However, this would come at the expense of more complexity in operating the network, and would require a certain confidence in the gain that ISNC can bring.

Appendix A

Dissemination modeling of concurrent sessions in DTMSN

In [103], the information dissemination model is designed for several concurrent unicast sessions in homogeneous DTN, when ISNC and SaW routing are employed. Now, we extend this work to heterogeneous DTN to predict the performance of contending unicast sessions.

A.1 Modeling Dissemination

We predict the evolution over time of the different numbers of nodes describing the dissemination process. To do so, we resort to a mean-field approximation that allows to predict the mean behavior of a system, modeled as a Markov chain, made of a growing number of interacting objects. We do not provide a formal proof that such a mean-field approximation hold in the present case, but rather only a sketch of the proof which allows to consider the limit behavior of the system when N tends to infinity so as to get fluid model of the performance. By Theorem 3.1 of [99], the quantities X_{ic}, Y_{ic}, $\tilde{X}_{Ic}$, $\tilde{Y}_{Ic}$ defined in Table 3.1, that are random processes depending on the random mobility process, can be approximated by deterministic processes that are the solutions of certain coupled Ordinary Differential Equations (ODEs). These ODEs stem from the limit of the system dynamics (the "drift") for large N and are called the fluid model. Let us now present the ODEs of the fluid model.

Below we present how the number of nodes carrying the packets of different sessions evolve over the time and communities, and describe the main components of the model.

A.1.1 Evolution of the buffer occupancy distribution

The ODE equations for $X_{ic}(t)$ and $Y_{ic}(t)$ write as

$$dX_{ic}(t) = \beta_{cs}N_c \sum_{j=0}^{i-1}\sum_{l=0}^{B-i} P_{gs}(c, i-j, j+l)P_j(c, j, l) + \beta_{cd}N_c \sum_{j=i+1}^{B} P_{ls}(j-i, c, j, m)\dots$$

$$+N_c \sum_{e=1}^{C}\sum_{j=0}^{i-1}\sum_{k=i-j}^{B} \beta_{ce}N_eP_{grs}(i-j, e, c, k, j) - \beta_{cs}N_c \sum_{j=0}^{B-i-1} P_j(c, i, j)P_{gsv}(c, i, j)\dots$$

$$-\beta_{cd}N_c \sum_{j=0}^{B-i} P_{lsv}(c, i, j) - N_c \sum_{e=1}^{C}\sum_{k=1}^{B} \beta_{ce}N_eP_{grsv}(e, c, k, i)\ .$$

$$dY_{ic}(t) = \beta_{ca}N_c \sum_{j=0}^{i-1}\sum_{l=0}^{B-i} P_{ga}(c, i-j, j+l)P_j(c, l, j)\big((j > 0 + (j == 0)P_{cgi0,c}\dots$$

$$+(j == 0)P_{ucus,c}\big) + \beta_{cd}N_c \sum_{j=i+1}^{B}\sum_{m=0}^{B-j} P_{la}(j-i, c, j, m)\dots$$

$$+N_c \sum_{e=1}^{C}\sum_{j=0}^{i-1}\sum_{k=i-j}^{B} \beta_{ce}N_eP_{gra}(i-j, e, c, k, j) - \beta_{ca}N_c \sum_{j=0}^{B-i-1} P_j(c, j, i)P_{gav}(c, j, i)\dots$$

$$-\beta_{cd}N_c \sum_{j=0}^{B-i} P_{lav}(c, j, i) - N_c \sum_{e=1}^{C}\sum_{k=1}^{B} \beta_{ce}N_eP_{grav}(e, c, k, i)\ .$$

The components of the above equations are defined as follows:

- $P_j(c, i, j)$: joint probability that a node which belongs to community c has in its buffer i indices from S and j indices from A. We call such node as (c, i, j)-node. The elements of $P_j(c, i, j)$ must fulfill following constraints: (i) $P_j = 0$ for $j > B - i$, (ii) for all $i\epsilon\{0, \dots, B\}$, $\sum_{j=0}^{B-i} P_j(c, i, j) = \frac{X_{ic}}{N_c}$, $\sum_{i=0}^{B-j} P_j(c, i, j) = \frac{Y_{jc}}{N_c}$, $X_{ic} \leq \sum_{j=0}^{B-i} Y_{jc}$, $Y_{ic} \leq \sum_{j=0}^{B-i} X_{jc}$ and $\sum_{i=0}^{B}\sum_{j=0}^{B-i} P_j(c, i, j) = 1$.
- $P_{nic,c,e}\big(k, bsf, bsr, K(t), K_U(t), v(c, t)\big)$: probability that there are k indices of S(or A) at node N_1 which are not in common with those at node N_2 and whose corresponding spray counters are still below M(or Q), when S(or A) has already spread out $K(t)$ indices. N_1 belongs to community c and has bsf indices of S(or A) and N_2 belongs to community e and has bsr indices of S(or A) in its buffer. $K_U(t)$ denotes the number of indices spread out until time t which still have spray-counter lower than M(or Q) and $v(c, t)$ (resp. $w(c, t)$) stores occurrence probability of every indices of S (resp. A) at time t in community c such that $v(c, t) = (\frac{\tilde{X}_{1c}}{\tilde{X}_c}, \dots, \frac{\tilde{X}_{K(t)c}}{\tilde{X}_c})$ with $\tilde{X}_c = \sum_I \tilde{X}_{Ic}$. Let $T_c = I : \tilde{X}_{Ic} > 0$ and E be the set of indices of S that can still spread: $E = \{I : 0 < \sum_{c=0}^{C} \tilde{X}_{Ic} < M\}$. Let S_c be a set of pairwise different elements from $T_c = \{1, \dots, K(t)\}$ whose cardinal is $|S_c|$. Then $S_{z,c}\big(K(t), z, v\big) = \sum_{S_c\epsilon T_c:|S_c|=z} \prod_{i\epsilon S_c} v_i \prod_{i\epsilon T_c - S_c}(1 - v_i)$. We define $P_{cn,c,e}$ as the probability that the number of indices not in common between nodes of community c and e is greater than k :

$$P_{cn,c,e}\big(k, bsf, bsr, K(t), K_U(t), v(c,t)\big) =$$

$$\begin{cases} 1 & \text{, if } k < bsf - bsr \\ 0 & \text{, elseif } k > bsf \\ \dfrac{\begin{pmatrix} bsf \\ k \end{pmatrix}(bsr+k)! S_{z,c}\big(K(t), bsr+k, v(c,t)\big)}{\sum_{k=0}^{bsf}\begin{pmatrix} bsf \\ k \end{pmatrix}(bsr+k)! S_{z,c}\big(K(t), bsr+k, v(c,t)\big)} & \text{, elseif } K_U(t) \geq K(t) \\ \dfrac{\sum_{S_1 \epsilon E:|S_1|=k}\prod_{i\epsilon S_1} v_i(c,t) S_{z,c}\big(K(t)-k, bsr, v_T - S_1(t)\big)}{\sum_{k=0}^{bsf}\begin{pmatrix} bsf \\ k \end{pmatrix}\sum_{S_1 \epsilon E:|S_1|=k}\prod_{i\epsilon S_1} v_i(c,t) S_{z,c}\big(K(t)-k, bsr, v_T - S_1(t)\big)} & \text{, otherwise} \end{cases}$$

then, $P_{nic,c,e}\big(k, bsf, bsr, K(t), K_U(t), v(c,t)\big) = P_{cn,c,e}\big(k, bsf, bsr, K(t), K_U(t), v(c,t)\big) - P_{cn,c,e}\big(k+1, bsf, bsr, K(t), K_U(t), v(c,t)\big)$.

- f_c: fraction of nodes in community c contaminated by packets of A.

$$\frac{df_c}{dt} = \big(\sum_{e=1}^{C}\beta_{ce}N_e f_e + \beta_{ca}\big)\dots$$

$$\Big((1 - \frac{N_{s,c}}{N_c})\big(1 - \frac{\sum_{c=1}^{C}\sum_{i=1}^{B} iX_{ic}(t) + \sum_{c=1}^{C}\sum_{j=1}^{B} jY_{jc}(t)}{K'_S M + K'_A Q}\big) - f_c\Big)\frac{s_{3,c}}{(1-f_c)} ,$$

where $N_{s,c}$ denotes secure nodes in community c that do not accept to relay the RLCs combining the packets of any contending session other than the session of interest, and $s_{3,c} = \sum_{k=0}^{B-1}\sum_{m=0}^{B-1-k} P_j(c,m,k)$ is the probability that node in community c can receive atleast one packet; $s_{3,c} = \sum_{k=0}^{B-1}\sum_{m=0}^{B-1-k} P_j(c,m,k)$. Let us denote $P_{cgi0,c}$ as probability that a node in community c is contaminated given that no extra-session index is in its buffer and $P_{ucus,c}$ as the probability that a node in community c is uncontaminated and unsecure given that no extra-session index is in its buffer:

$$P_{cgi0,c} = \begin{cases} \frac{f_c - \sum_{i=0}^{B}\sum_{j=1}^{B} P_j(c,i,j)}{\sum_{i=0}^{B} P_j(c,i,0)}, & \text{, if } \sum_{i=0}^{B} P_j(c,i,0) > 0 \\ 1 & \text{, otherwise} \end{cases}$$

$$P_{ucus,c} = \begin{cases} \frac{1 - f_c - \frac{N_{s,c}}{N_c}}{\sum_{i=0}^{B} P_j(c,i,0)}, & \text{, if } \sum_{i=0}^{B} P_j(c,i,0) > 0 \\ 0 & \text{, otherwise} \end{cases}$$

- $R(t)$ (resp. $S(t)$): number of indices of S(resp. A) that D has got until time t. Let $P_c ds$ (resp. $P_c da$) be the average number of indices of S(resp. A) that D receives per unit time around t from community c when S(resp. A) has released $K_S(t)$(resp. $K_A(t)$) indices.

$$\frac{dR(t)}{dt} = \sum_{c=1}^{C} \beta_{dc} N_c P_c ds,$$

$$P_c ds = \sum_{i=1}^{B} \sum_{j=0}^{B-i} P_j(c,i,j) \sum_{l=1}^{i} \sum_{n_s=l}^{i} \sum_{n_A=0}^{j} \sum_{ll=l}^{j+i} l.distrlS(c,d,i,B,n_s,n_A,ll,l) \ldots$$

$$\big((n_s + n_A \geq ll) Pr[\xi = ll] + Pr[\xi > ll](n_s + n_A == ll)\big) \ldots$$

$$P_{nic,c,d}\big(n_s, i, L_S(t), K_S(t), v(c,t)\big) P_{nic,c,d}\big(n_A, i, L_A(t), K_A(t), w(c,t)\big) \ ,$$

The expression for $S(t)$ and $P_c da$ can be deduced similarly.

- $P_{s,c}$(resp. $P_{a,c}$): average number of indices that S(resp. A) gives around time t to community c.

$$P_{s,c} = \sum_{l=1}^{B} \sum_{i=0}^{B-l} \sum_{j=0}^{B-l-i} l.P_{gs}(c,l,i+j) P_j(c,i,j) \ ,$$

$$P_{a,c} = \sum_{l=1}^{B} \sum_{i=0}^{B-l} \sum_{j=0}^{B-l-i} l.P_{ga}(c,l,i+j) P_j(c,j,i) \big((i > 0) + (i == 0) P_{cgi0,c} + (i == 0) P_{ucus,c}\big) \ ,$$

$$\frac{dK_S(t)}{dt} = \sum_{c=1}^{C} \beta_{sc} N_c P_{s,c} \ .$$

- $P_{gs}(c,m,n)$(resp. $P_{ga}(c,m,n)$): probability that S(resp. A) gives m indices of S(resp. A) to a node of community c that already has n indices of S and A.

$$P_{gs}(c,m,n) = Pr\Big(min\big(bw, B-n, K^{'} - K_S(t)\big) = m\Big) \ ,$$

$$P_{ga}(c,m,n) = Pr\big(min(bw, B-n) = m\big) \ .$$

- $P_{ls}(k,c,j,m)$ (resp. $P_{la}(k,c,j,m)$): probability that (c,j,m)-node looses k indices of S(resp. A) by meeting D. Considering two communicating nodes N_1 and N_2, we denote r as the number of indices in N_1 already obtained by N_2, n_S (resp n_A) as the number of indices of S(resp. A) present at N_1 but not yet at N_2, and ll as the number of indices transferred from N_1 to N_2.

$$P_{ls}(k,c,j,m) = P_j(c,j,m) f(k,c,j,m),$$

$$f(k,c,j,m) = \sum_{r=0}^{k} \sum_{n_s=0}^{j} \sum_{n_A=0}^{m} \sum_{ll=k-r}^{n_s+n_A} distrlS(c,d,j,B,j-r,n_A,ll,k-r) \ldots$$

$$\big((j - r + n_A \geq ll) Pr[\xi = ll] + Pr[\xi > ll](j - r + n_A == ll)\big) \ldots$$

$$P_{nic,c,d}\big(j-r, j, L_S(t), K_S(t), v(c,t)\big) P_{nic,c,d}\big(n_A, m, L_A(t), K_A(t), w(c,t)\big) \ .$$

The expression of $P_{la}(k,c,j,m)$ is symmetric.

- $P_{grs}(p,e,c,k,j)$: probability that node of community c with j indices of S gains p indices of S from a node of community e having k indices of S. Let us denote $g(p,e,c,k,m,j,l)$ as the probability that p packets exactly be transferred from (e,k,m)-node to a (c,j,l)-node.

$$P_{grs}(p,e,c,k,j) = \sum_{l=0}^{B-j}\sum_{m=0}^{B-k} P_j(e,k,m)P_j(c,j,l)g(p,e,c,k,m,j,l)\ldots$$

$$\Big((m=0)(1-P_{cgi0,e}) + \big((m==0)P_{cgi0,e} + (m>0)\big)\ldots$$

$$\big((l>0) + (l==0)P_{cgi0,c} + (l==0)P_{ucus,c}\big)\Big),$$

$$g(p,e,c,k,m,j,l) = \sum_{n_s=p}^{k}\sum_{n_A=0}^{m}\sum_{ll=p}^{n_s+n_A} distrlS(e,c,k,j,n_s,n_A,ll,i-j)\ldots$$

$$\big((n_s+n_A \geq ll)Pr[\xi = ll] + (n_s+n_A == ll)Pr[\xi > ll]\big)\ldots$$

$$P_{nic,e,c}\big(n_s, k, j, K_S(t), K_{US}(t), v(c,t)\big) P_{nic,e,c}\big(n_A, m, l, K_A(t), K_{UA}(t), w(c,t)\big) \ ,$$

where, $K_{US}(t)$ (resp. $K_{UA}(t)$) is the number of indices spread by S(resp. A) until time t which have still spray-counter lower than M(resp. Q). $P_{gra}(p,e,c,k,j)$ can be deduced similarly as $P_{grs}(p,e,c,k,j)$.

- $P_{gsv}(c,i,j)$(resp. $P_{gav}(c,i,j)$): probability that S(resp. A) gives atleast one index to (c,i,j)-node.

$$P_{gsv}(c,i,j) = \sum_{k=1}^{B-(i+j)} P_{gs}(c,k,i+j) \ ,$$

$$P_{gav}(c,j,i) = P_j(c,j,i)\big((i>0)+(i==0)P_{cgi0,c}+(i==0)P_{ucus,c}\big) \sum_{k=1}^{B-(i+j)} P_{ga}(b,k,j+i) \ .$$

- $P_{lsv}(c,i,j)$ (resp. $P_{lav}(c,i,j)$): probability that (c,i,j)-node looses atleast one index of S(resp. A) after meeting with D.

$$P_{lsv}(c,i,j) = P_j(c,i,j)\sum_{l=1}^{i} f(l,c,i,j) \ .$$

The expression for $P_{lav}(c,i,j)$ can be deduced similarly as $P_{lsv}(c,i,j)$.

- $P_{grsv}(e,c,k,i)$: probability that a node of community c with indices of S gains atleast one index of S by meeting with a node of community e with k indices of S.

$$P_{grsv}(e,c,k,i) = \sum_{j=0}^{B-i-1}\sum_{m=0}^{B-k} P_j(e,k,m)\ldots$$

$$P_j(c,i,j)\Big((m==0)(1-P_{cgi0,e}) + \big((m==0)P_{cgi0,e} + (m>0)\big)\ldots$$

$$\big((j>0) + (j==0)P_{cgi0,c} + (j==0)P_{ucus,c}\big)\Big)\sum_{l=0}^{k} g(l,e,c,k,m,i,j)\;.$$

A.1.2 Evolution of the index dissemination distribution

The ODE for $\tilde{X}_{Ic}$ write as:

$$\frac{d\tilde{X}_{Ic}}{dt} = \sum_{e=1}^{C}\beta_{ce}N_eN_cA_{R,ce}(I) + \beta_{cs}N_bc_{2c}A_{S,c} - \beta_{cd}\tilde{X}_{Ic}A_{D,c}\;.$$

The components of the above equations are defined as follows:

- $A_{R,ce}(I)$: fraction of nodes in community c without index I that gets it from another relay in community e.

$$A_{R,ce}(I) = \begin{cases} 0 \quad \text{, if } \sum_{c=1}^{C}\tilde{X}_{Ic} \geq M \\ \sum_{k=1}^{B}\sum_{m=0}^{B-k} P_j(e,k,m)\big(1-ptnh(e,I,k)\big)\ldots \\ \sum_{i=0}^{B-1}\sum_{j=0}^{B-1-i} P_j(c,i,j)ptnh(c,I,i)h(e,c,k,m,i,j) \quad \text{, otherwise} \end{cases}$$

where, $ptnh(c,I,i,K(t))$ is the probability that node of community c with indices i of S does not have index I of S in its buffer: $ptnh(c,I,i,K(t)) = \frac{S_{z,c}\left(K(t)-1,i,v_{T_c-\{I\}}(c,t)\right)}{S_{z,c}\left(K(t),i,v(c,t)\right)}$ and $h(e,c,k,m,i,j)$ is the probability that index I of S gets forwarded from (e,k,m)-node to a (c,i,j)-node:

$$h(e,c,k,m,i,j) = \sum_{l=1}^{k}\sum_{n_s=l}^{k}\sum_{n_A=0}^{m}\sum_{ll=l}^{n_s+n_A}\frac{P_c(I)}{sumhnh(e,c,k,i)}distrlS(e,c,k,i,n_s,n_A,ll,l)\ldots$$

$$\big(Pr[\xi=ll](n_s+n_A\geq ll) + Pr[\xi>ll](n_s+n_A==ll)\big)\ldots$$

$$P_{nic,e,c}\big(n_s,k,i,K_S(t),K_{US}(t),v(e,t)\big)P_{nic,e,c}\big(n_A,m,j,K_A(t),K_{UA}(t),w(e,t)\big)\ldots$$

$$\Big((m==0)(1-P_{cgi0,e}) + \big((m==0)P_{cgi0,e} + (m>0)\big)\ldots$$

$$\big((j>0) + (j==0)P_{cgi0,c} + (j==0)P_{ucus,c}\big)\Big)\;,$$

where, $distrlS(e,c,k,i,n_s,n_A,ll,l) = \binom{ll}{l}(sPc)^l(1-sPc)^{ll-l}$, $sPc = \sum_I \frac{P_c(I)}{sumhnh(e,c,k,i)}$ $ptnh(c,I,i)\big(1-ptnh(e,I,k)\big)$ and $sumhnh(e,c,k,i) = \sum_I ptnh(c,I,i)(1-ptnh(e,I,k))$. We denote $P_c(I)$ as the probability that I get replicated or forwarded to community c given that

I pertains to a set of n_s uncommon indices given that $\sum_{c=1}^{C} \tilde{X}_{Ic} < M$. Let c_l be the l-th order statistic of a set of $n = n_s - 1$ elements following a discrete distribution $Pr[X = x]$ such that for $a = 0, \dots, M-1$, $Pr[X = x] = \frac{|I:round(\tilde{X}_{Ic})=a|}{|I:round(\sum_{c=1}^{C}\tilde{X}_{Ic})<M|}$.Then cdf of c_l is $Pr[c_l < x] = \sum_{j=0}^{n-l} \begin{pmatrix} n \\ j \end{pmatrix} (Pr[X \geq x])^j (Pr[X < x])^{n-j}$. When several indices have same number of copies which turn to be l-th minimum, we get:

$$P_c(I) = Pr[c_l \geq \sum_{c=1}^{C} \tilde{X}_{Ic}(t)] \dots$$

$$min\Big(1, \frac{l - (n_s-1)Pr(X < \sum_{c=1}^{C}\tilde{X}_{Ic})}{1 + (n_s-1)Pr[c_l = \sum_{c=1}^{C}\tilde{X}_{Ic}(t)]/Pr[c_l \geq \sum_{c=1}^{C}\tilde{X}_{Ic}(t)]}\Big) .$$

- $A_{S,c}$: activation of index I in community c.

$$A_{S,c} = (K_S(t) \leq I)(K_S(t) + B > I)(K_S(t) \leq K^{'}),$$

$$c_{2c} = \sum_{i=0}^{B-1}\sum_{j=0}^{B-1-i} P_j(c,i,j)ptnh(c,I,i)Pr[\xi > I - K_S(t)]\big(B - (i+j) > I - K_S(t)\big) .$$

- $A_{D,c}$: fraction of nodes holding I in community c that loose it by meeting with D.

$$A_{D,c} = \sum_{i=1}^{B}\sum_{j=0}^{B-i} P_j(c,i,j) \sum_{l=1}^{i}\sum_{r=0}^{l-1}\sum_{n_A=0}^{j}\sum_{n_s=i-r}^{i-r}\sum_{ll=l-r}^{n_s+n_A} \Big(\frac{l}{sumhnh(c,d,i,L_S(t))} \dots$$

$$\big(1 - P_I(t)\big)\big(1 - ptnh(c,I,i)\big) + \frac{r}{sumhnh(c,d,i,L_S(t))} P_I(t)\big(1 - ptnh(c,I,i)\big)\Big) \dots$$

$$distrlS(c,d,i,B,n_s,n_A,ll,l-r)\big((n_s + n_A \geq ll)Pr[\xi = ll] \dots$$

$$+Pr[\xi > ll](n_s + n_A == ll)\big)P_{nic,c,d}\big(n_s, i, L_S(t), K_S(t), K_S(t), v(c,t)\big) \dots$$

$$P_{nic,c,d}\big(n_A, j, L_A(t), K_A(t), K_A(t), w(c,t)\big) ,$$

where, $sumhnh(c,d,i,L_S(t)) = \sum_I (1-P_I(t))(1-ptnh(c,I,i))$ and $sumhh(c,d,i,L_S(t)) = \sum_I P_I(t)(1 - ptnh(c,I,i))$.

- $P_I(t)$(resp. $Q_I(t)$) : probability that destination has received index I of S(resp. A) by time t.

$$\frac{dP_I(t)}{dt} = \sum_{c=1}^{C} \beta_{dc} A^{'}_{D,c}\big(1 - P_I(t)\big)N_c .$$

The expression of $Q_I(t)$ is symmetric to $P_I(t)$.

- $A^{'}_{D,c}$: fraction of nodes holding I in community c give it to D provided that D has not yet received it.

$$A'_{D,c} = \sum_{i=1}^{B}\sum_{j=0}^{B-i} P_j(c,i,j)(1 - ptnh(c,I,i)) \sum_{l=1}^{i}\sum_{n_s=l}^{i}\sum_{n_A=0}^{j}\sum_{ll=l}^{n_s+n_A} \dots$$
$$\frac{P_d(I)}{sumhnh(c,d,i,L_S(t))} n_s distrlS(c,d,i,B,n_s,n_A,ll,l) \dots$$
$$\big((n_s + n_A \geq ll)Pr[\xi = ll] + Pr[\xi > ll](n_s + n_A == ll)\big) \dots$$
$$P_{nic,c,d}\big(n_s, i, L_S(t), K_S(t), K_S(t), v(c,t)\big) \dots$$
$$P_{nic,c,d}\big(n_A, i, L_A(t), K_A(t), K_A(t), w(c,t)\big) \; .$$

A.1.3 Numerical Results

In this section, we assess the accuracy of the analysis based on fluid models. We generate a synthetic contact trace with $N = 1000$, $C = 1$ and $\beta = 5.10^{-4}$ using Matlab. We consider all the relay nodes have fixed buffer size $B = 3$ packets, bandwidth is Poisson distributed with mean $Bw = 3$ packets, $K_S = K_A = 5$, $K'_S = K'_A = 5$, and the number of copies that can be spread is limited to $M = Q = 40$. We obtain the simulation results that are averaged over 30 runs and the 5% confidence interval.

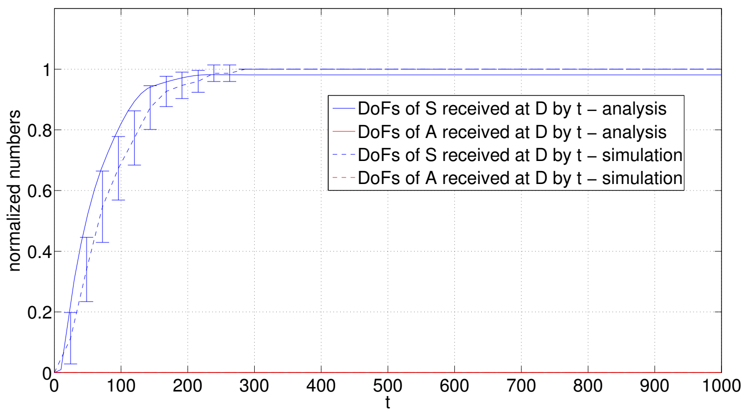

FIGURE A.1: $N_{s,1} = 1000$.

Figure A.1 plot the evolution of the normalized (i.e. divided by K_S or K_A) number of DoFs of S and A received by D over time. These numbers of DoFs represent the mean numbers of indexes received from S and A by D. Since we take $N_{s,1} = N$, the relay nodes are not allowed to carry the packets of adversary session. Therefore, only the packets of session of interest get disseminated in the network.

We now consider the case where $N_{s,1} \neq N$. From Figure A.2 - A.3, we observe that as we decrease the value of $N_{s,1}$, the number of nodes that can relay the packets of adversary session increases. When $N_{s,1} = 0$, all the nodes can equally spread packets of both the sessions, and

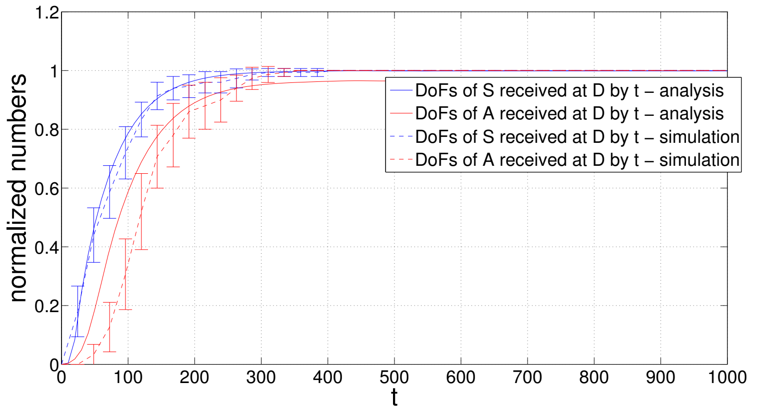

FIGURE A.2: $N_{s,1} = 800$.

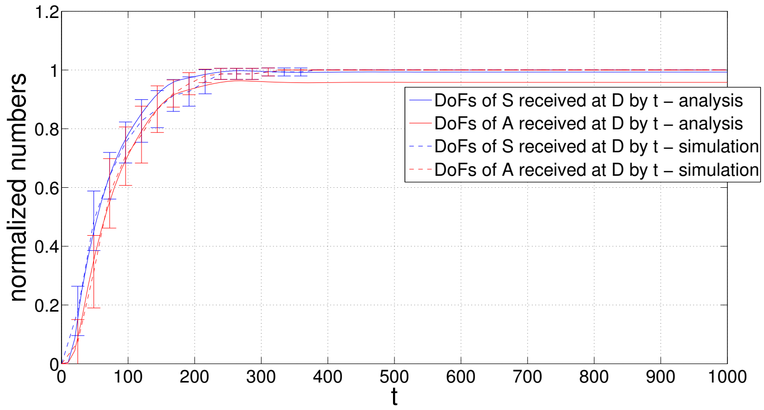

FIGURE A.3: $N_{s,1} = 500$.

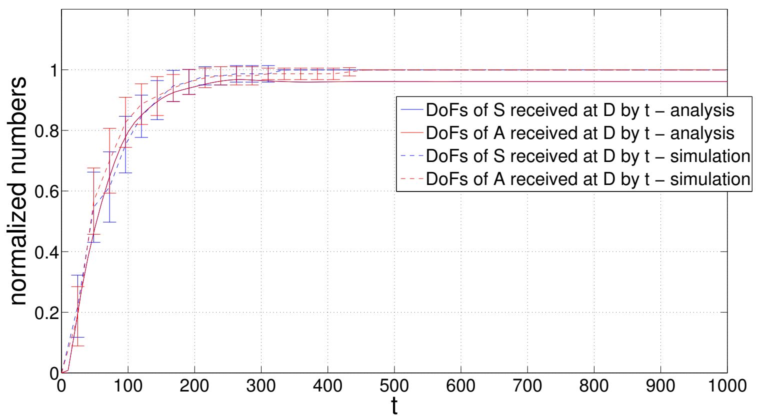

FIGURE A.4: $N_{s,1} = 0$.

thereby the destination node D receives the packets of S as much as that of A. We observe from these figures, the good fit between the simulation results and the analytical prediction.

Appendix B

ISNC transmission algorithm for multiple unicast sessions

We present an algorithm for multi-session ISNC to describe how two nodes transmit packets in their buffer when they have contact opportunity. We consider that a node A may send packets to a node B that is it meeting and all the parameters considered are presented in Table B.1.

Symbol	*Meaning*
Communication settings	
N_S	total message generated during the sending phase
S_i for $i \epsilon \{1, \dots, N_S\}$	source node of session i
D_i	destination node of session i
K_i	number of information packets of session i
K_i'	maximum number of packets that can be released by S_i
M_i	maximum number of copies of an index released by S_i
$\mathcal{S}_i$	set of indices associated to pure payloads sent out by source S_i
$\mathcal{S}_{N_S+(i-1)N_S+j}$	set of indices emitted by S_i associated to a mixed payload, that a combination from pure payloads of S_i and S_j
X_i	number of nodes that carry i indices in $\mathcal{S}_i$
$X_{N_S+(i-1)N_S+j}$ for $j \epsilon \{1, \dots, N_S\}$ and $j \neq i$	number of nodes that carry i indices in $\mathcal{S}_{N_S+(i-1)N_S+j}$)
l	$l = \sum_i l_i + \sum_j l_{N_S+(i-1)N_S+j}$ for a $(c, 1)$-node

TABLE B.1: Main notation used for Multi-session ISNC model.

3: Protocol with Multi-session ISNC - Part 1

Data: a $(a, \mathbf{l}^A)$-node A (i.e., in community a with $\mathbf{l}^A$), and a node $(b, \mathbf{l}^B)$-node B, the number of packet transmission opportunities w, $S = \{S_1, \ldots, S_{N_S}\}$, $D = \{D_1, \ldots, D_{N_S}\}$, $S\_unique = unique(S)$, $D\_unique = unique(D)$, $u_{ab} = \{u_{ab}^i, \ldots, u_{ab}^{N_S+N_S*N_S}\}$.

Result: How many and what packets generated by A to be stored at B

Let $Hd(\mathbf{u})$ be the distribution of a discrete random variable Y with $N_S + N_S * N_S$ values, value i being taken with probability u_{ab}^i.

```
i=1;
while i ≤ length(S_unique) do
    if A == S_i and B ≠ D_i then
        Draw q_i from a binomial distribution (w, u_ab^i).
        if Num_of_RLCs_sent_by_S_i < K'_i then
            Send x = min(q_i, K'_i − Num_of_sent_RLCs_by_A, B − l^B) RLCs with indices
            from Num_of_sent_RLCs_by_S_i + 1 up to Num_of_sent_RLCs_by_S_i + x
    i=i+1;
end
i=1;
while i ≤ length(D_unique) do
    if B == D_i then
        Drop the packets with indices of S_i already present at D_i. Update l^A accordingly.
        x=0;
        while x ≤ w do
            Draw y from Hd(u).
            j=1;
            while j ≤ N_S do
                if y == j then
                    Send 1 packet whose index is in S_j to D_i and drop it from A if
                    j == i.
                    x=x+1;
                j2=1;
                while j2 ≤ N_S do
                    if y = N_S + (j − 1) + j2 then
                        if j2 ≠ j then
                            Send 1 packet whose index is in S_{N_S+(j−1)+j2} to D_i.
                            x=x+1;
                    j2=j2+1;
                end
                j=j+1;
            end
        end
    i=i+1;
end
```

4: Protocol with Multi-session ISNC - Part 2

```
x=0;
while x ≤ w do
    Draw y from Hd(u).
    i=1;
    while i ≤ N_S do
        if y==i then
            if A ≠ S_i and B ≠ D_i then
                (1) Let n^i be the list of indices in S_i at A that are neither in S_i nor in
                S_{N_S+(i-1)+j2} for j2 = {1, ..., N_S} and j2 ≠ i at B and whose counter
                is strictly greater than 1.
                Let p be the set of packets at A corresponding to these indices.
                if n^i not empty and B - l^B ≥ 1 then
                    Send a packet q to B with:
                        q.index = p_1.index = n^i_1
                        q.counter = p_1.counter(SimBet) from A + q.counter at B
                        q.payload = p_1.payload
                    Update
                        l^B_i = l^B_i + 1
                        p_1.counter = p_1.counter(SimBet)
                        Remove n^i_1 from n^i
                    x=x+1;
        j=1;
        while j ≤ N_S do
            if y = N_S + (i - 1) + j then
                if j ≠ i then
                    if A ≠ S_i and A ≠ S_j and B ≠ D_i and B ≠ D_j then
                        (2) Let n^{N_S+(i-1)+j} = [v, n^i], where n^i stems from step (1) and v
                        is the list of indices in S_{N_S+(i-1)+j} at A that are neither in S_i nor
                        in S_{N_S+(i-1)+j} at B and whose counter is strictly greater than 1.
                        Let p be the set of packets at A corresponding to these indices.
                        if n^{N_S+(i-1)+j} not empty and B - l^B ≥ 1 then
                            Send a packet q to B with:
                                q.index = p_1.index = n^{N_S+(i-1)+j}_1
                                q.counter =
                            p_1.counter(SimBet) from A + q.counter at B
                                q.payload = RLC(packet i and j at A)
                            Update
                                l^B_{N_S+(i-1)+j} = l^B_{N_S+(i-1)+j} + 1
                                p_1.counter = p_1.counter(SimBet)
                            x=x+1;
            j=j+1;
        end
        i=i+1;
    end
end
Go to the beginning of Algo. 4, exchange A and B and perform again all the steps.
```

Bibliography

[1] S. Burleigh, “Ring road: User application on a high-latency network,” January 2014. [Online]. Available: http://ipnsig.org/ipn-events/2014-space-technology-innovations/

[2] D. Traskov, N. Ratnakar, D. Lun, R. Koetter, and M. Medard, “Network coding for multiple unicasts: An approach based on linear optimization,” in *IEEE International Symposium on Information Theory*, Seattle, WA, July 2006, pp. 1758–1762.

[3] T. C. Ho, Y. H. Chang, and K. J. Han, “On constructive network coding for multiple unicasts,” in *44th Annual Allerton Conference on Communication, Control and Computing*, Monticello, Illinois, September 2006.

[4] E. M. Daly and M. Haahr, “Social network analysis for information flow in disconnected delay-tolerant MANET,” *IEEE Transactions on Mobile Computation*, vol. 8, no. 5, pp. 606 – 621, May 2009.

[5] “A community resource for archiving wireless data at dartmouth.” [Online]. Available: http://crawdad.cs.dartmouth.edu/index.php

[6] A. Chaintreau, A. Mtibaa, L. Massoulié, and C. Diot, “Distributed caching over heterogeneous mobile networks,” in *Proceedings of ACM International Conference on emerging Networking Experiments and Technologies*, New York, USA, December 2007.

[7] S. Ali, J. Qadir, and A. Baig, “Routing protocols in delay tolerant networks - a survey,” in *International Conference on Emerging Technical*, 2010, pp. 153–158.

[8] “IRTF: Internet research task force.” [Online]. Available: http://www.irtf.org/

[9] “DTNRG: Delay-tolerant networking research group.” [Online]. Available: http://www.dtnrg.org

[10] 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,” in *ACM Architectural Support for Programming Languages and Operating Systems*, San Jose, CA, October 2002.

[11] "Bytewalla DTN on Android Phones." [Online]. Available: http://sourceforge.net/projects/bytewalla/

[12] J. Ott, "Liberouter: Towards autonomous neighborhood networking," in *SophiaTech networks seminar*, France, June 2014.

[13] A. Doria, "Providing connectivity to the saami nomadic community," in *2nd International Conference on Open Collaborative Design for Sustainable Innovation*, Bangalore, India, December 2002.

[14] A. E. Al-Fagih and H. S. Hassanein, "Routing schemes for delay-tolerant networks - an applications perspective," in *PhD dissertations*, Ontario, Canada, 2012.

[15] E. P. C. Jones and P. A. S. Ward, "Routing strategies for delay-tolerant networks," in *ACM Computer Communication Review*, 2006.

[16] J. Broch, D. A. Maltza, D. Johnson, Y. C. Hu, and J. Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols," in *ACM/ IEEE International Conference on Mobile Computing and Networking*, Texas, USA, October 1998, pp. 85–97.

[17] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Efficient routing in intermittently connected mobile networks: the single-copy case," *IEEE/ACM Transactions on Networking*, vol. 16, no. 1, pp. 63 – 76, February 2008.

[18] T. Spyropoulos, T. Turletti, and K. Obraczka, "Routing in delay-tolerant networks comprising heterogeneous node populations," *IEEE Transactions on Mobile Computation*, vol. 8, no. 8, pp. 1132 – 1147, August 2009.

[19] A. Lindgren, A. Doria, and O.Schelén, "Probabilistic routing in intermittently connected networks," *ACM Special Interest Group for Mobile applications, Mobile Computation Communication Review*, vol. 7, no. 3, pp. 19–20, July 2003.

[20] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and wait: An efficient routing scheme for intermittently connected mobile networks," in *ACM Special Interest Group on Data Communication, Workshop on Delay-Tolerant Networking*, August 2005, pp. 252–259.

[21] T. Hossmann, T. Spyropoulos, and F. Legendre, "A complex network analysis of human mobility," in *IEEE International Workshop on Network Science for Communication Networks*, Shanghai, China, April 2011, pp. 876 – 881.

[22] E. Bulut, W. Zijian, and B. K. Szymanski, "Impact of social networks in delay tolerant routing," in *IEEE Global Communications Conference*, Honolulu, HI, December 2009.

[23] P. Hui, J. Crowcroft, and E. Yoneki, "Bubble rap: Social-based forwarding in delay tolerant networks," *IEEE Transactions on Mobile Computation*, vol. 10, no. 11, pp. 1576 – 1589, November 2011.

[24] R. Ahlswede, N. Cai, S. R. Li, and R. W. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204 – 1216, July 2000.

[25] R. Koetter and M. Medard, "An algebraic approach to network coding," *IEEE/ACM Transactions on Networking*, vol. 11, no. 5, pp. 782–795, 2003.

[26] T. Ho, R. Koetter, M. Médard, D. R. Karger, and M. Effros, "The benefits of coding over routing in a randomized setting," in *IEEE International Symposium on Information Theory*, Yokohama, Japan, July 2003.

[27] J. Widmer and J.-Y. Le Boudec, "Network coding for efficient communication in extreme networks," in *ACM Special Interest Group on Data Communication, Workshop on Delay-Tolerant Networking*, Philadelphia, USA, 2005, pp. 284–291.

[28] Y. Lin, B. Li, and B. Liang, "Stochastic analysis of network coding in epidemic routing," *IEEE Journal on Selected Areas in Communication*, vol. 26, no. 5, June 2008.

[29] A. Eryilmaz and D. S. Lun, "Control for inter-session network coding," in *Information Theory and Applications workshop*, San Diego, CA, January 2007.

[30] Z. Li and B. Li, "Network coding in undirected networks," in *Annual Conference Information Sciences and Systems*, Princeton, NJ, March 2004.

[31] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft, "XORs in the air: Practical wireless network coding," *IEEE/ACM Transactions on Networking*, vol. 6, no. 3, pp. 497 – 510, 2008.

[32] M. Frodigh, P. Johansson, and P. Larsson, "Wireless ad hoc networking: the art of networking without a network," in *Ericsson Review*, 2000, pp. 248–263.

[33] P. Ghosekar, G. Katkar, and P. Ghorpade, "Mobile ad hoc networking: Imperatives and challenges," in *International Journal of Computer Applications*, 2010, pp. 153–158.

[34] Z. Zhang, "Routing in intermittently connected mobile ad hoc networks and delay tolerant networks: Overview and challenges," in *IEEE Communications Surveys and Tutorials*, 2006.

[35] V. Cerf, "Solar system internet," January 2014. [Online]. Available: http://ipnsig.org/wp-content/uploads/2014/02/CERF-IPN-and-DTN.pdf

[36] K. Fall, "A delay-tolerant network architecture for challenged internets," in *ACM Special Interest Group on Data Communication*, Karlsruhe, Germany, August 2003.

[37] S. Kapadia1, B. Krishnamachari, and L. Zhang, "Data delivery in delay tolerant networks: A survey." [Online]. Available: http://www.intechopen.com

[38] T. Small and Z. J. Haas, "Resource and performance tradeoffs in delay-tolerant wireless networks," in *ACM Special Interest Group on Data Communication, Workshop on Delay-Tolerant Networking*, 2005.

[39] S. Jain, K. Fall, and R. Patra, "Routing in a delay-tolerant network," in *ACM Special Interest Group on Data Communication*, Protland, USA, August 2004.

[40] J. Shen, S. Moh, and I. Chung, "Routing protocols in delay tolerant networks: A comparative survey," in *International Technical Conference on Circuits/Systems,Computation and Communication*, 2008.

[41] A. Krifa, C. Barakat, and T. Spyropoulos, "Message drop and scheduling in DTNs: Theory and practice," *IEEE Transactions on Mobile Computing*, vol. 11, no. 9, pp. 1470 – 1483, September 2012.

[42] Z. J. Haas and T. Small, "A new networking model for biological applications of ad hoc sensor networks," *IEEE/ACM Transactions on Networking*, vol. 14, no. 1, February 2006.

[43] T. Small and Z. J. 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 Mobile Ad Hoc Networking and Computing*, 2003, p. 233 244.

[44] Y. Zhu, B. Xu, X. Shi, and Y. Wang, "A survey of social-based routing in delay tolerant networks: Positive and negative social effects," *IEEE Communication Surveys and Tutorials*, vol. 15, no. 1, pp. 387 – 401, 2013.

[45] J. Fan, Y. Du, W. Gao, J. Chen, and Y. Sun, "Geography-aware active data dissemination in mobile social networks," in *IEEE International Conference on Mobile Ad hoc and Sensor Systems*, San Francisco, CA, November 2010, pp. 109–118.

[46] E. Brewer, M. Demmer, B. Du, M. Ho, M. Kam, S. Nedevschi, J. Pal, R. Patra, S. Surana, and K. Fall, "The case for technology in developing regions," *IEEE Journal on Computer*, vol. 38, no. 6, pp. 25 – 38, 2005.

[47] P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, and C. Diot, "Pocket switched networks and human mobility in conference environments," in *ACM Special Interest Group on Data Communication, Workshop on Delay-Tolerant Networking*, August 2005, pp. 244–251.

[48] I. Psaras, L. Wood, and R. Tafazolli, "Delay-/disruption-tolerant networking: State of the art and future challenges," in *Center for Communication Systems Research*, Surrey, UK, November 2009.

[49] B. Hull, V. Bychkovsky, Y. Zhang, K. Chen, M. Goraczko, A. Miu, E. Shih, H. Balakrishnan, and S. Madden, "Cartel: A distributed mobile sensor computing system," *ACM Conference on Embedded Networked Sensor Systems*, pp. 125–138, November 2006.

[50] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, "Maxprop: Routing for vehicle-based disruption-tolerant networks," in *IEEE International Conference on Computer Communications*, vol. 10, Barcelona, Spain, April 2006, pp. 1–11.

[51] H. Ntareme, M. Zennaro, and B. Pehrson, "Delay tolerant network on smartphones: Applications for communication challenged areas," in *Proceedings of the 3rd Extreme Conference on Communication: The Amazon Expedition*, Manaus, Brazil, September 2011, pp. 26 – 30.

[52] "DTN Bone." [Online]. Available: https://sites.google.com/site/dtnresgroup/home/dtn-bone

[53] M. Garetto, P. Giaccone, and E. Leonardi, "Capacity scaling of sparse mobile ad hoc networks," in *IEEE International Conference on Computer Communications*, Arizona, USA, April 2008, pp. 206–210.

[54] A. Mtibaa and K. A. Harras, "Social forwarding in large scale networks: Insights based on real trace analysis," in *Proceedings of 20th International Conference on Computer Communication Networks*, Maui, Hawaii, USA, August 2011, pp. 1–8.

[55] A. Vahdat and D. Becker, "Epidemic routing for partially-connected ad hoc networks," in *Technical Report CS-200006*, Duke University, 2000.

[56] A. Picu, T. Spyropoulos, and T. Hossmann, "An analysis of the information spreading delay in heterogeneous mobility DTNs," in *IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*, San Francisco, CA, June 2012.

[57] A. Chaintreau, J.-Y. Le Boudec, and N. Ristanovic, "The age of gossip: spatial mean field regime," *ACM Special Interest Group on Measurement and Evaluation*, vol. 37, no. 1, pp. 109–120, June 2009.

[58] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Efficient routing in intermittently connected mobile networks: the multi-copy case," *IEEE/ACM Transactions on Networking*, vol. 16, no. 1, pp. 77–90, February 2008.

[59] F. D. Pellegrini, E. Altman, and T. Bascar, "Optimal monotone forwarding policies in delay tolerant mobile ad hoc networks with multiple classes of nodes," in *Proceedings of the 8th IEEE Symposium on Modeling and Optimization in Mobile Ad Hoc and Wireless Networks*, Avignon, France, May 31 2010, pp. 497 – 504.

[60] L. C. Freeman, “A set of measures of centrality based on betweenness,” *Sociometry*, vol. 40, no. 1, pp. 35–41, 1977.

[61] ——, “Centrality in social networks conceptual clarification,” *Social Networks*, vol. 1, no. 3, pp. 215–239, 1979.

[62] P. V. Marsden, “Egocentric and sociocentric measures of network centrality,” *Social Networks*, vol. 24, no. 4, pp. 407–422, October 2002.

[63] M. Everett and S. P. Borgatti, “Ego network betweenness,” *Social Networks*, vol. 27, no. 1, pp. 31–38, 2005.

[64] A. Mtibaa, M. May, C. Diot, and M. Ammar, “Peoplerank: Social opportunistic forwarding,” in *IEEE International Conference on Computer Communications*, San Diego, CA, March 2010, pp. 1–5.

[65] T. Karagiannis, J.-Y. Le Boudec, and M. Vojnović, “Power law and exponential decay of inter contact times between mobile devices,” in *ACM Mobile Computing and Networking*, Montreal, Canada, September 2007.

[66] A. Picu and T. Spyropoulos, “Minimum expected *-cast time in DTNs,” in *Proceedings of nternational Conference on Bio-Inspired Models of Network, Information, and Computing Systems*, Avignon, France, December 2009, pp. 103–116.

[67] X. Zhang, G. Neglia, J. Kurose, and D. Towsley, “Performance modeling of epidemic routing,” *Computer and Telecommunications Networking*, vol. 51, pp. 2867–2891, July 2007.

[68] R. Diana and E. Lochin, “Modeling the delay distribution of binary spray and wait routing protocol,” in *IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, Autonomic and Opportunistic Communication*, San Francisco, CA, June 2012.

[69] A. Picu and T. Spyropoulos, “Forecasting dtn performance under heterogeneous mobility: The case of limited replication,” in *IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, Seoul, June 2012, pp. 569 – 577.

[70] N. Eagle and A. Pentland, “Reality mining: sensing complex social systems,” *Personal and Ubiquitous Computing*, vol. 10, no. 4, pp. 255 – 268, March 2006.

[71] V. Lenders, J. Wagner, and M. May, “Measurements from an 802.11b mobile ad hoc network,” in *International Symposium on a World of Wireless, Mobile and Multimedia Networks*, NY, USA, June 2006.

[72] S. Ioannidis, L. Massoulié, and A. Chaintreau, "Distributed caching over heterogeneous mobile networks," *ACM Special Interest Group on Measurement and Evaluation*, vol. 38, no. 1, pp. 311–322, June 2010.

[73] H. S. B. Shally and M. Garg, "Performance evaluation of rapid, and sppray-and-wait dtn routing protocols under backhole attack," *International Journal of Research in Engineering and Technology*, vol. 3, no. 1, pp. 35–40, January 2014.

[74] A. Balasubramanian, B. Levine, and A. Venkataramani, "DTN routing as a resource allocation problem," *ACM Special Interest Group on Data Communication, Computer Communication Review*, vol. 37, no. 4, pp. 373–384, October 2007.

[75] M. S. Rahim, P. Casari, F. Guerra, and M. Zorzi, "On the performance of delay-tolerant routing protocols in underwater networks," in *IEEE OCEANS*, Spain, June 2011, pp. 1–7.

[76] A. Lindgren and K. S. Phanse, "Evaluation of queueing policies and forwarding strategies for routing in intermittently connected networks," in *Communication System Software and Middleware*, New Delhi, India, 2006, pp. 1 – 10.

[77] A. Krifa, C. Barakat, and T. Spyropoulos, "Optimal buffer management policies for delay tolerant networks," in *IEEE Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, 2008, pp. 260–268.

[78] A. Krifa, "Towards better content dissemination applications for disruption tolerant networks," in *PhD dissertations*, Sophia Antipolis, France, April 2012.

[79] F. N. dos Santos, B. Ertl, C. Barakat, T. Spyropoulos, and T. Turletti, "CEDO: Content-centric dissemination algorithm for delay-tolerant networks," *ACM/IEEE International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, pp. 377–386, November 2013.

[80] C. Fragouli, J.-Y. Le Boudec, and J. Widmer, "Network coding: An instant primer," *ACM Special Interest Group on Data Communication, Computer Communication Review*, vol. 36, no. 1, pp. 63–68, January 2006.

[81] J. Widmer, C. Fragouli, and J.-Y. Le Boudec, "Energy-efficient broadcasting in wireless ad-hoc networks," in *IEEE International Symposium on Network Coding*, 2005.

[82] Y. Lin, B. Li, and B. Liang, "Efficient network coded data transmissions in disruption tolerant networks," in *IEEE International Conference on Computer Communications*, Phoenix, USA, April 2008, pp. 1508–1516.

[83] M. Kim, M. Medard, U. M. O'Reilly, and D. Traskov, "An evolutionary approach to inter-session network coding," in *IEEE International Conference on Computer Communications*, 2009, pp. 450 – 458.

[84] S. Jain, M. Demmer, R. Patra, and K. Fall, "Using redundancy to cope with failures in a delay tolerant network," *Special Interest Group on Data Communication, Mobile Computation Communication Review*, vol. 35, no. 4, pp. 109–120, 2005.

[85] A. G. Dimakis, V. Prabhakaran, and K. Ramchandran, "Decentralized erasure codes for distributed networked storage," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2809–2816, June 2006.

[86] Y. Wang, S. Jain, M. Martonosi, and K. Fall, "Erasure-coding based routing for opportunistic networks," in *ACM Special Interest Group on Data Communication, Workshop on Delay-Tolerant Networking*, Philadelphia, USA, August 2005.

[87] T. T. Thai, J. Lacan, and E. Lochin, "Joint on-the-fly network coding/video quality adaptation for real-time delivery," *Signal Processing: Image Communication*, vol. 29, no. 4, p. 449461, 2014.

[88] M. Yang and Y. Yang, "A linear inter-session network coding scheme for multicast," in *IEEE International Symposium on Networking Computation and Applications*, 2008, pp. 177 – 184.

[89] M. Heindlmaier, D. Lun, D. Traskov, and M. Medard, "Wireless inter-session network coding - an approach using virtual multicasts," in *IEEE International Conference on Communication*, 2011, pp. 1–5.

[90] D. S. Lun, M. Medard, T. Ho, and R. Koetter, "Network coding with a cost criterion," in *International Symposium on Information Theory and its Applications*, 2004, pp. 1232–1237.

[91] A. Khreishah, W. Chih-Chun, and N. B. Shroff, "Rate control with pairwise intersession network coding," *IEEE/ACM Transactions on Networking*, vol. 18, no. 3, pp. 816–829, June 2010.

[92] H. Seferoglu, A. Markopoulou, and K. K. Ramakrishnan, "I2NC: Intra- and inter-session network coding for unicast flows in wireless networks," in *IEEE International Conference on Computer Communications*, September 2011, pp. 1035 – 1043.

[93] X. Zhang, G. Neglia, J. Kurose, and D. Towsley, "Benefits of network coding for unicast applications in disruption-tolerant networks," *IEEE/ACM Transactions on Networking*, vol. 21, no. 5, pp. 1407–1420, October 2013.

[94] S. Ahmed and S. S. Kanhere, "Hubcode: hub-based forwarding using network coding in delay tolerant networks," *Wireless Communication and Mobile Computation*, vol. 13, no. 9, p. 828846, May 2011.

[95] S. Maheshwar, Z. Li, and B. Li, "Bounding the coding advantage of combination network coding in undirected networks," *IEEE Transactions Information Theory*, vol. 58, no. 2, pp. 570– 584, 2012.

[96] A. Picu and T. Spyropoulos, "Distributed stochastic optimization in opportunistic networks: The case of optimal relay selection," in *ACM Workshop on Challenged Networks*, Chicago, IL, USA, September 2010.

[97] R. Groenevelt and P. Nain, "Message delay in MANETs," in *ACM Special Interest Group on Measurement and Evaluation*, Banff, Canada, June 2005, pp. 412–413.

[98] ——, "Message delay in MANETs," in *ACM Special Interest Group on Measurement and Evaluation*, Banff, Canada, June 2005, pp. 412–413.

[99] 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, April 1970.

[100] M. Benaïm and J.-Y. Le Boudec, "A class of mean field interaction models for computer and communication systems," *Performance Evaluation Methodologies and Tools*, vol. 65, no. 11-12, pp. 823–838, April 2008.

[101] N. Shrestha and L. Sassatelli, "On optimality of routing policies in delay-tolerant mobile social networks," in *IEEE Vehicular Technology Conference*, Dresden, Germany, June 2013, pp. 1–5.

[102] K. Price and R. Storn, "Differential Evolution (DE) for continuous function optimization." [Online]. Available: http://www1.icsi.berkeley.edu/~storn/code.html

[103] L. Sassatelli and M. Medard, "Inter-session network coding in delay-tolerant networks under spray-and-wait routing," in *IEEE International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks*, Paderborn, Germany, May 2012, pp. 103–110.

[104] N. Gast, B. Gaujal, and J.-Y. Le Boudec, "Mean field for markov decision processes: From discrete to continuous optimization," *IEEE Transactions on Automatic Control*, vol. 57, no. 9, pp. 2266–2280, September 2012.

[105] I. M. Mitchell, "A toolbox of level set methods," 2012. [Online]. Available: http://www.cs.ubc.ca/~mitchell/ToolboxLS/toolboxLS-1.1.pdf

[106] V. D. Blonde, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal Statistical Mechanics: Theory and Experiment*, July 2008.

[107] M. E. J. Newman, “Analysis of weighted networks,” *Physical Review E, 70: 056131*, July 2004.

[108] G. Palla, I. Derényi, I. Farkas, and T. Vicsek, “Uncovering the overlapping community structure of complex networks in nature and society,” *Nature 435, 814-818*, 2005.

[109] A. Mtibaa and K. A. Harras, “Fog: Fairness in mobile opportunistic networking,” in *IEEE Communication Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, Salt Lake City, UT, 2011, pp. 260–268.

Traductions Françaises

B.1 Introduction

Contexte

En 2002, Kevin Fall a introduit le terme "Réseaux Tolérant le Délai" alors qu'il adoptait des idées du design de l'Internet Inter-Planétaire (IPN) au déploiement de réseau terrestre [7]. Depuis, une quantité importante de recherches ont été dédiées au design des Réseaux Tolérant le Délai (ou les Déconnexions), abrégés par DTN. Les DTN sont des réseaux sans-fil sans connectivité continue en raison de leur mode de fonctionnement sans infrastructure. Les équipements sans-fil utilisent un mode ad hoc de communication pour partager/transférer l'information quand il se retrouvent à proximité l'un de l'autre uniquement. C'est pour cela que ces communications sont aussi qualifiés d'"opportunistes". Elles reposent sur la mobilité des équipements participants, mobilité qui peut être aléatoire comme dans les environnements terrestres extrêmes, ou déterministe comme dans les réseaux spciaux planifiés. Le nombre croissant de technologies radio courte portée (dont Bluetooth et WiFi 802.11) dans les appareils mobiles permet d'envisager le déploiement de tels réseaux. Quand les DTN sont constitués d'appareils mobiles portés par des humains, ces équipements peuvent se scinder en différents groupes, ou communautés, en raison de la mobilité hétérogène. Nous appelons de tels DTN consitiués de noeuds mobiles regroupés en communautés Réseaux Sociaux Mobiiles Tolérant le Délai (DTMSN).

L'usage répandu d'appareils électroniques portables, les bénéfices financiers dus au bas coût des équipements rendent les DTN applicable dans divers environnements. Tandis que le groupe de recherche DTN de l'IRTF (*Internet Research Task Force*) [8, 9] se concentre principalement sur les communications dans l'espace lointain, il y a beaucoup d'autres applications des DTN etlles que: applications civiles (par exemple *Pocket Switched Networks* - PSN, réseaux véhiculaires), applications militaires, réseaux de capteurs (par exemple ZebraNet [10]), et les réseaux sociaux. Les trois buts principaux dans les applications civiles sont: (i) fournir un accès Internet à des communautés/peuplades lointaines, (ii) fournir un accès meilleur marché au contenu grâce à l'échange de fichiers en mode ad hoc, (iii) exploiter les interactions des utilisateurs pour convoyer l'information, aidant ainsi à décharger les réseaux de télécommunications pour satisfaire la demande croissante des utilisateurs. Nous pouvons citer de récents prototypes de

DTN tels que Bytewalla [11], Liberouter [12] et Saami Network Connectivity [13].

Les DTN se caractérisent par des partitions fréquentes et longues de leurs noeuds en raison de leur faible densité. Ces réseaux sont faits pour opérer dans différents scenarii incluant une hétérogénéité des standards, une connectivité intermittente entre des noeuds adjacents, et une absence de route de bout en bout au moment entre la source et la destination de l'information. Dans de tels environnements, les ressources des noeuds sont très limitées: puissance de calcul, mémoire et capacité de transmission [14]. Il est donc important de prendre en compte les problèmes de gestion de buffer et de files. Un des enjeux fondamentaux pour la prise en compte des déconnexions est le routage [15]. Le routage dans les DTN peut être vu comme un problème d'optimisation où les branches du graphe sont indisponibles par périodes et les capacités de stockage aux noeuds sont limitées [14]. Les protocoles de routage classiques de l'Internet et ceux des réseaux mobiles ad hoc (MANET) supposent que le réseaux est connecté, c'est-à-dire qu'à tout moment il existe une route entre toute paire de noeuds. ces protocoles de sont pas applicables aux DTN DTN [16, 17]. Plusieurs protocoles de routage pour DTN ont été proposés avec des stratégies allant de l'inondation) la réplication limitée.

Motivation

Le routage est l'un des défis pour faire fonctionner les DTN parce-qu'il n'y a donc pas de garantie d'existence d'un chemin entre deux noeuds à tout instant. Pour arriver à communiquer, les noeuds doivent donc avoir recours au paradigme Stocker-Porter-Transférer qui implique un délai inhérent pour la communication de bout en bout. Pour diminuer ce délai, plusierus copies du même messages peuvent être disséminées. Comment les disséminer a fait l'objet de nombreuses études menant à ce qu'on appelle des protocoles de routage opportunistes pour atteindre un compromis entre performance et consommation de ressources [18, 19, 20]. Ces protocoles considèrent les noeuds et leur mobilité homogène.

Quand le DTN est constitué d'humains, leur mobilités déterminent les occasions de tranferts. La mobilité humaine exhibe des motifs hétérogènes où les noeuds se regroupent en communautés en raison des relations sociales (amitiés, échanges professionnels, etc.) [21]. ces caractéristiques sociales ont été prises en compte dans plusieurs algorithmes de routage [4, 18, 22, 23]. Ceci nous motivent à étudier les politiques de routage optimal dans les DTMSN. De plus, les noeuds doivent utiliser précautionneusement leur ressources en mémoire et énergue.

Pour augmenter le bénéfice de disséminer des paquets redondants, des paquets codés peuvent être générés par les relais plutôt que de simples copies, ce qui correspond à effectuer du codage réseau (NC). NC est un paradigme réseau qui généralise le concept de routage [24, 25]. En particulier, le NC aléatoire [26] a attiré un intérêt croissant pour les DTN [27, 28]. Les bénéfices sont une augmentation de débit, ainsi qu'une adaptabilité aux changements de topologie. Il y a deux types de NC: en NC intra-session, seulement les paquets de la même session sont combinés, alors qu'en NC inter-session (ISNC), les paquets appartenant à des sessions différentes peuvent être combinés. ISNC est nécessaire pour atteindre un débit optimal en général [29] mais représente un problème d'optimisation difficile. Dans les graphes non-orientés, il a été montré théoriquement que les bénéfices de NC sont faibles [30]. Cependant, même si les contacts bidirectionnels en DTN rendent le graphe équivalent des contacts non-orientés, les ressources limitées (buffer et bande passante) peuvent rendre ISNC attractif, comme en réseaux sans-fil maillés [31]. C'est pourquoi nous traitons dans cette thèse le problème ouvert d'identifier si, et dans quelles conditions, ISNC peut petre bénéfique aux DTMSN, et de concevoir des politiques ISNC pour servir plusieurs utilisateurs efficacement, où l'efficacité réfère à des métriques telles que délai moyen, probabilité de réception, ou équité dans l'occupation mémoire.

Plan du manuscript

Cette thèse comporte quatre chapitres principaux, qui visent tous à traiter graduellement le problème de la conception de politiques ISNC pour servir plusieurs utilisateurs efficacement, où l'efficacité réfère à des métriques telles que délai moyen, probabilité de réception, ou équité. Avant de plonger dans le problème du ISNC, le premier chapitre 2 traite de l'optimisation des politiques de routage en DTMSN. Une polique de tranfert optimale est conçue pour un modèle de réseau mobile multi-communautés, basée sur des approximations de champ moyen menant à un modèle fluide du processus de dissemination. L'optimalité de la politique est vérifiée théoriquement et comparée aux politiques en ligne basée utilité, pour évaluer la distance entre ces politiques pratique et l'optimale supposant une connaissance complète du réseau. Dans le chapitre 3, une politique ISNC paramétrée est conçue pour inclure à la fois le routage et le codage avec une contrainte en énergie. Nous discutons aussi du problème d'optimisation d'ISNC. Dans le chapitre 4, nous examinons un certain nombre de topologies pour identifier quand ISNC peut être bénéfique. Nous construisons sur l'algorithme de routage SimBet une politique ISNC qui procure des bénéfices dans les DTMSN. Dans le chapitre 5, nous concevons des critères de codage décentralisés permettant aux noeuds de déclencher le codage ISNC en se basant seulement sur des informations qu'il ont collectées localement, et nous faisons des tests sur des traces synthétiques et de réseaux réels.

Contributions

Dans cette thèse, nous apportons les contributions suivantes:

- Sur l'optimalité des politiqus de routage en DTMSN.
    - Nous formulons le problème de trouver la politique de transfert optimal dans le temps entre toute paire de communautés, pour maximiser la probabilité de réception

avant une certaine date limite, avec une contrainte d'énergie, en utilisant des approximations de champ moyen menant à un modèles fluides du processus de dissémination pour un réseau multi-communautés.

  ◦ Nous prouvons que les politiques optimales ont une structure de seuil par communauté, généralisant ainsi les travaux existants pour les DTN homogènes.
  ◦ Nous fournissons une illustration numérique en utilisant un algorithme d'optimisation heuristique, et nous discutons la comparaison entre de telles politiques optimal et les politique de routage en ligne, c'est-à-dire ne supposant pas une connaissance centralisée, et donc implémentables.

- Modélisation de la dissémination de sessions concurrentes dans les DTMSN.
  ◦ Nous concevons un politique paramétrée d'ISNC par paire, qui inclut à la fois le routage et le codage avec une contrainte en énergie. Nous présentons le protocole de dissémination résultant.
  ◦ Nous dérivons le modéle de performance grâce à une approximation de champ moyen menant à un modèle fluide, et validons ce modèle par des expérimentations numériques.
  ◦ Nous discutons l'optimisation du contrôle de ISNC dans les DTMSN, et montrons que le modèle fluide peut être utilisé pour concevoir une politique optimal exploitant conjointement les relations sociales et ISNC. En montrant des gains numériques, nous illustrons la pertinence de notre politique de contrôle qui est basé sur le structure grain épais des communautés.
- Etude empirique de ISNC par paire dans les DTMSN.
  ◦ Nous étudions ISNC sur différentes topologies jouet, à savoir les topologies chaîne et papillon, pour identifier quand ISNC peut être bénéfique ou néfaste.
  ◦ Nous examinons l'impact sur les performances ISNC de plusieurs paramètres, tels que la contrainte sur le nombre maximum de copies par paquet, la charge réseau, la taille buffer des noeuds relais et les politiques de gestion de buffer.
  ◦ Nous utilisons ISNC au dessus d'un algorithme de routage social, SimBet [4], et montrons par simulation que sur l'ensemble des routes sélectionnées par SimBet, ISNC peut être bénéfique.
- Conception de politiques ISNC en ligne pour DTMSN.
  ◦ Nous concevons des critères de codage décentralisés pour déclencher ISNC quand il peut être bééfique basés seulement sur l'information locale disponible aux noeuds sur les différentes sessions actives.

- Nous validons notre approche sur des topologies jouet et les testons sur des traces réelles.
- Nous présentons un modèle analytique simplifié pour utiliser l'optimisation précédente, et expliquons les limites de notre modèle.

B.2 Résumé du Chapitre 2

Avant d'aborder NC, nous considérons d'abord le problme de l'optimisation de la politique de routage dans DTMSN. Ce chapitre vise identifier la structure des politiques de routage optimales, étant donné un modle de réseau multi-communautaire, puis évaluer la distance la politique optimale de certaines politiques de routage basées sur l'utilité.

Nous utilisons le modle de mobilité hétérogne pris en compte dans [18, 22]: le réseau est fait de N nuds mobiles divisées en M communautés (peuvent łtre centrées autour d'un point central chacune) comme indiqué dans la figure B.1 multi-communautaire. Notre accent est mis sur le réseau qui est composé de personnes portant des appareils portables tels que les réseaux commutation de poche (PSN). Dans ces réseaux, les gens sont liés par différentes relations sociales telles que l'amitié, le commerce, le statut, etc., et ils se divisent en différents groupes ou communautés telles que les gens l'intérieur de la młme communauté se réunissent plus souvent que ceux entre les différentes communautés.

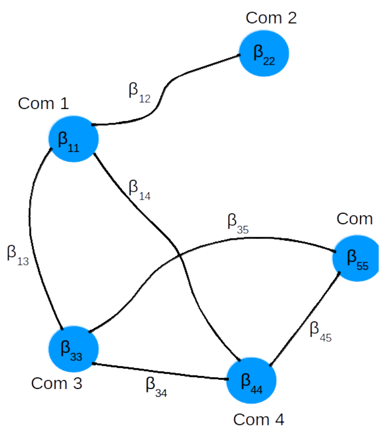

FIGURE B.1: Multi-community network model.

Dans notre modle de réseau nous considérons que le nombre de nuds dans la communauté i est N_i, et nous supposons qu'un noeud appartient une seule communauté. On note le nombre total de nuds par $N = \sum_{i=1}^{M} N_i$, et considérons la partition de noeud vecteur $\mathbf{N} = (N_1, \ldots, N_m)$. Le temps entre les contacts de n'importe quelle paire de noeuds des communautés i et j est répartie de faon exponentielle [65, 97] avec une certaine moyenne et l'exactitude de ce modle a été discuté dans [98] et représenté par un certain nombre de modles de mobilité (Random Walker, direction aléatoire, aléatoire Waypoint). Nous supposons que la valeur moyenne est de $1/\beta ij$, où β_{ij} est l'intensité inter-réunion définie comme le nombre moyen de réunions par unité de temps entre un noeud donné de la communauté i et un noeud donné de la communauté j et β_{ii} est l'intensité intra-réunion entre les nuds de la communauté i, et nous supposons que $\beta_{ii} > \beta_{ij}$, pour $i \neq j$, pour tout $i, j \in \{1, \ldots, M\}$, c'est--dire les nuds se réunissent plus souvent quand ils appartiennent la młme communauté que quand ils appartiennent différentes communautés. Soit β la matrice stockant les $\{\beta_j\}_{i,j=1}^{M}$. Le scénario l'étude est un noeud source S de la communauté C_S qui veut envoyer un message (ou paquets) un noeud de destination de la communauté C_D. Nous considérons le cas multi-paquet plus tard dans notre travail. Nous considérons une session unicast sans trafic de fond. Soit τ le délai de livraison jusqu'auquel le message est pertinent la destination. Pour des raisons de notation plus légères, un nud de la communauté i est noté un i-noeud.

Dans ce chapitre, nous avons abordé le problme de l'optimisation des politiques de routage dans les DTN mobiles sociaux. Grâce une approximation de champ moyen du processus de propagation, nous avons formulé le problme de trouver les probabilités de transfert en fonction du temps entre deux communautés pour maximiser la probabilité de livraison avant une certaine date limite sous contrainte énergétique donnée. Nous avons prouvé théoriquement que les politiques de transfert optimales sont à seuil par communauté, fournissant une analyse de ce résultat sur un exemple numérique et nous avons discuté de comparaison des politiques existantes décentralisées pour évaluer la distance de ces politiques de routage pratiques aux politiques de routage optimales en termes de structure sociale de réseau sous-jacent.

B.3 Résumé du Chapitre 3

Pour améliorer le bénéfice de la diffusion de paquets redondants, des paquets redondants codés peuvent łtre générés par les relais la place de ou en plus de copies répliquées, qui correspond à effectuer du réseau de codage (NC) dans le DTN.

Décrivons simplement comment un noeud ayant deux paquets de deux sessions différentes, effectue ISNC pour forger un nouveau paquet codé pour łtre envoyé. Le processus décrit ciaprs est représenté dans la figure B.2. Tous les nuds peuvent identifier le numéro de session de chaque paquet. Considérez que les paquets P_1 et P_2, appartenant aux sessions S_1 et S_2, sont des combinaisons linéaires aléatoires (RLC) des K_1 et K_2 paquets d'information d'origine, respectivement.

Les coefficients d'entłte P_1 et P_2 sont donc de longueur K_1 et K_2, tandis que les charges utiles sont de longueur L_1 et L_2, où L_1 et L_2 sont la taille maximale de paquets de S_1 et S_2, respectivement. Le paquet résultant d'une RLC de P_1 et P_2 a une entłte de coefficients de taille $K_1 + K_2$, et la charge utile est de taille $\max(L_1, L_2)$.

Les paquets K_1 originaux de la session 1 peuvent łtre récupérés si et seulement si la matrice fait des coefficients de codage peut subir une élimination de Gauss-Jordan entranant seuls éléments de la matrice d'identité de taille K_1 plus les K_1 colonnes assignées une session et pour les rangées correspondantes, toutes les autres colonnes sont égales zéro. Par la suite, le nombre de reus Degré de liberté (DOF) de la session 1 est le nombre d'éléments d'identité sur ces K_1 colonnes.

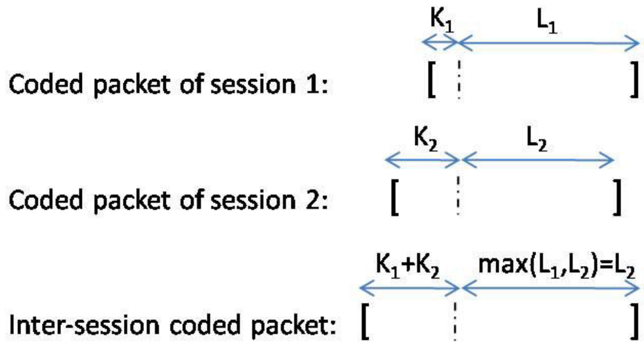

FIGURE B.2: Generation of an inter-session network coded packet

Nous avons élaboré une politique de contrle paramétrée de ISNC par paires, et exprimé les problmes de contrle d'optimisation grce un modle de performance. Le schéma est en młme temps une politique de routage et codage qui permet d'optimiser une fonction d'utilité définie sur les deux sessions, sous une contrainte d'énergie. Notre politique décide quels nuds peuvent mélanger les sessions en fonction de leurs communautés plutt que sur leurs propriétés individuelles, ce qui rend la politique conue évolutive avec le nombre de noeuds si le nombre de communautés reste limitée. Nous avons montré que les gains numériques de ISNC sur NC intra-session peuvent en effet łtre obtenus.

Ce chapitre ne vise pas présenter un protocole de ISNC décentralisé autonome qui peut łtre confronté des politiques de routage existantes dans DTN. Il vise concevoir, modéliser et prouver le bénéfice d'une politiques ISNC par paire et basée sur la connaissance des communautés sociales. Plus précisément, le problme regrouper les sessions par deux n'est pas étudié ici. Détecter et sélectionner ces paires de sessions mélanger fait partie du problme de décentralisation. En outre, une étude plus approfondie est nécessaire, autorisée par le protocole présenté et son modle analytique, pour étudier sur quels graphes sociaux (parmi lesquels non-orienté comme dans la figure 3.4) et dans quelles conditions (par exemple, la taille K_1 et K_2 des sessions et le budget énergie) ISNC peut łtre bénéfique. La prochaine étape aprs ce travail est d'étudier numériquement le problme d'optimisation afin d'en extraire des heuristiques pour élaborer une politique de ISNC décentralisée pour DTN sociaux. En particulier, le modle sera adapté aux politiques de routage décentralisées existantes (telles que SimBet [4]) afin d'élaborer des décisions pertinentes et locales de ISNC.

B.4 Résumé du Chapitre 4

Le chapitre précédent a présenté une politique ISNC et un modle analytique où le nombre de paquets par session peut łtre quelconque, correspondant au cas où un fichier est divisé en plusieurs paquets, et la métrique (qu'il s'agisse du délai ou de la probabilité de réception) est sur l'ensemble du fichier. En outre, deux sessions sont considérées. Afin de s'attaquer au problme de l'optimisation de la politique ISNC en réduisant l'espace des paramtres et en identifiant des heuristiques pertinentes, nous analysons d'abord dans le présent chapitre l'impact de divers paramtres sur la performance ISNC. Ce chapitre vise donc identifier, travers des expériences progressives sur les topologies jouet, ce que peuvent łtre les avantages de ISNC dans DTMSN, spécifiquement dans quelles hypothses sur le routage, la gestion de la mémoire tampon, et la charge du réseau.

Dans [94], NC est considéré certains nuds hubs intermédiaires, mais seulement travers les paquets destinés au młme noeud de destination. Dans [93], Zhang *et al.* considrent la fois NC intra et inter-session dans les DTN homogne. Pour les sessions unicast avec différentes sources et destinations, ISNC incontrlé est montré ne pas faire mieux que intra-session. Dans cette thse, nous nous attaquons au problme plus général du ISNC pour sessions unicast avec des destinations différentes. Lors de l'examen de plusieurs sessions unicast, ISNC peut apporter des gains en débit et équité [2, 91] sur liens avec ou sans pertes. Cependant, le problme d'optimisation de ISNC pour plusieurs sessions unicast a été prouvé NP-complet [25], en particulier en raison des problmes communs de sélection de sous-graphe et des décisions de codage, qui peuvent łtre résolus indépendamment pour une seule session de multidiffusion [90]. Par conséquent, toutes les uvres portant sur le problme du sous-optimal ISNC ciblent, de façon constamment améliorée, des méthodes approchées [2, 29, 31, 91]. Ces approches ne sont pas directement applicables DTN où ils assument topologies fixes et peuvent entraner une lourde signalisation. En particulier, dans [29], Eryilmaz et Lun introduisent une stratégie de routage-ordonnancement-codage utilisant des techniques de contre-pression. En modélisant les flux codés comme *empoisonné* [2], l'échange des tailles des file d'attente est destiné déterminer l'emplacement de l'encodage, le décodage et les noeuds générant les paquets remèdes. En outre, tous ces travaux consièrent des réseaux orientés. Cependant, il n'y a a priori aucune raison de considérer que deux noeuds peuvent échanger des paquets dans une seule direction dans les DTN. Li et Li dans [30] ont montré théoriquement ce que peut łtre l'amélioration de débit maximal avec NC intra-session dans les réseaux non orientés, par rapport ce qui peut łtre obtenu avec le routage entier, demi-entier et fractionné. En particulier, pour le problme de multidiffusion, le rapport d'augmentation de débit est borné par deux entre NC et routage demi-entier, ou młme moins avec le routage fractionnée [30, 95]. Toutefois, les ressources partagées (mémoire tampon, la bande passante de contact) dans DTMSN rendent ISNC attrayant tout comme dans les réseaux maillés sans fil [31], même si ces réseaux ne sont pas orientés. Par conséquent, pour s'attaquer au problme ouvert de la conception de ISNC dans les DTN sociaux, l'une des étapes que nous choisissons est d'étudier par des expériences sur des topologies simples, faites de communautés de nuds homognes reliés entre eux d'une manire hétérogne, quelle amélioration peut łtre apportée par ISNC, et comment.

La premire section traite d'une topologie de chane simple, tandis que le second étudie la topologie papillon typique. Dans ces topologies, les noeuds du graphe ne représentent plus noeuds de réseau, mais les communautés entières de noeuds.

Dans ce chapitre, nous avons analysé la performance de la politique ISNC en réalisant des expériences sur un certain nombre de topologies de jouets tels que la topologie du réseau avec deux communautés, les topologies de la chane et le papillon, pour déterminer quand ISNC peut łtre avantageux ou préjudiciable aux DTMSN. Nous avons étudié les effets de divers paramtres tels que le nombre maximum de copies par paquets, la charge du réseau, la taille du tampon et les politiques de gestion de mémoire tampon. Nous concluons que lorsque ISNC est mis en oeuvre avec des protocoles de routage sociale tels que SimBet, il peut łtre bénéfique. Dans le chapitre suivant, nous concevons des critres décentralisés de ISNC où les paquets codés sont générés sans connaisance globale de la topologie du réseau, mais avec seulement l'information locale disponible aux noeuds.

B.5 Résumé du Chapitre 5

Dans ce chapitre, nous abordons le problme de décider de déclencher ISNC, si oui, où et quand dans le réseau, c'est-à-dire mélanger des sessions en se basant uniquement sur des informations locales recueillies au niveau des nuds. Nous présentons notre approche pour atteindre cet objectif, présentant les critres conus progressivement améliorés.

Nous considérons le problme de décision de mélange séance $Z_1 = (S_1, D_1)$ avec la session $Z_2 = (S_2, D_2)$ au noeud N_c. Comme précédemment, P_1, P_2 et P_3 désignent les paquets de session et XOR des deux, respectivement. Il peut łtre assuré que ISNC ne fait pas pire que le routage simple si et seulement si:

$$\begin{cases} \text{Delay for } D_1 \text{ to get } P_2 < \text{Delay for } n_c \text{ to get } P_1 \text{ and } P_2 + \text{Delay for } P_3 \text{ to reach } D_1 \text{ from } n_c \\ \text{AND} \\ \text{Delay for } D_2 \text{ to get } P_1 < \text{Delay for } n_c \text{ to get } P_1 \text{ and } P_2 + \text{Delay for } P_3 \text{ to reach } D_2 \text{ from } n_c. \end{cases} \tag{B.1}$$

qui correspond au codage seulement si chaque paquet de remde peut être arrivé aux deux destinations au moment où le paquet codé arrive. Une autre condition nécessaire pour ISNC pas pire que le routage simple est d'assurer que le paquet codé n'atteint pas la destination plus lentement qu'un paquet non codé. Cependant, nous n'incluons pas explicitement cette contrainte dans le critre de codage parce que nous considérons qu'elle est assurée par la gestion de la mémoire tampon choisie.

Dans ce chapitre, nous avons conu des critres de codage décentralisés permettant de déclencher ISNC en décidant quand et quelles sessions mélanger ensemble en présence de plusieurs sessions simultanées dans le réseau, avec notre objectif de concevoir une politique de ISNC en ligne qui peut łtre bénéfique pour DTMSN. Nous avons également présenté un modle analytique simplifié pour s'attaquer au problme de l'optimisation de la politique ISNC en expliquant certaines limites de notre modle.

B.6 Conclusions et Perspectives

B.6.1 Conclusions

Le but de cette thèse a été de concevoir des politiques de transfert efficaces combinant codage réseau inter-session et algorithmes de routage social dans les Réseaux Sociaux Mobiles Tolérant le Délai. Pour traiter le problème, nous avons pris des approches à la fois théoriques et pratiques. L'approche analytique a consister à concevoir une politiques de contrôle ISNC paramétrée (avec l'algorithme Spray-and-Wait), et dériver son modèle de performance grâce à une approximation fluide. L'approche empirique a étudié, sur diverses topologies, quand ISNC peut être bénéfique. Nous avons identifié que, malgré la bi-directionalité (et l'asynchronisme) des contacts en DTMSN, qui empêche habituellement NC d'être intéressant, ISNC peut être bénéfique quand un couplé à un algorithme de routage social (SimBet dans notre étude) en raison du sous-ensemble de route sélectionné, introduisant le niveau nécessair de directionnalité. Nous nous sommes ensuite concentrés sur le design de politiques ISNC en ligne pour les DTMSN, en concevant des critères de codage décentrailsés pour déclencher ISNC quand un gain peut être apporté. Nous avons également explicité comment le modèle analytique développé peut servir au design d'un tel critère.

B.6.2 Perspectives

Ce travail a permis des progrès dans la compréhension de quels peuvent être les gains de ISNC dans l'opération des DTMSN, et comment ils peuvent être obtenus. Un certain nombre d'étapes restent à être franchies, dan sle but d'obtenir une politique d'ISNC en ligne complète, fonctionnant conjointement avec des algorithmes de routage sociaux.

Les suites immédiates à ces travaux sont:

- Tout d'abord, il serait important d'évaluer soigneusement le gain de ISNC en termes d'équité parmi les nuds. En effet, comment les nuds qui sont des concentrateurs principaux dans les réseaux sont chargés avec le trafic des autres est une préoccupation importante pour la viabilité de ce type de réseaux, et la politique SimBet [4] ou FOG [109] ont spécifiquement porté sur ce problme de troquer équité et l'efficacité.

- Deuximement, la conception d'un bon critre de codage, qui décide de mélanger les sessions en prenant en compte tous les paramtres importants, parmi lesquels la charge du réseau, reste atteindre. En particulier, nous avons identifié la quantité clé à estimer avec plus précision.

- Troisimement, des évaluations approfondies sur les traces du monde réel doivent łtre effectuées avec un tel codage critre.

Les extensions à plus long terme:

- Une direction importante vers l'élaboration de politiques ISNC plus efficaces dans les DTMSN est la procédure d'optimisation en ligne du critre de codage. Dans cette thse, nous avons dérivé des modles analytiques, puis nous sommes efforcés d'extraire des heuristiques partir d'expériences afin de concevoir des critres qui peuvent ensuite łtre testés a posteriori en utilisant le modle de codage. Une autre voie serait d'exprimer le problme d'optimisation dans le cas simple d'un paquet par session (donc une formulation plus simple que celle du chapitre 3), et de voir si une telle approche gloutonne, comme celle de [41] peut łtre utilisée.

- Cette étude s'est appuyée sur notre choix d'utiliser de manire opportuniste, sous forme de paquets de remède, ceux propagés par l'algorithme de routage fondé sur l'utilité. Toutefois, nous avons pu identifier que la réception dans les temps de ces paquets de remède devient moins probable lorsque le contexte énergétique ou la charge du réseau deviennent plus contraint. Comme la réception de ces paquets de remède est cruciale dans la performance de ISNC, on peut envisager de transférer délibérément les paquets de remède lorsqu'il est prévu que ISNC augmente les performances. Toutefois, cela se ferait au détriment de plus de complexité dans l'exploitation du réseau, et exigerait une certaine confiance dans le gain que peut apporter ISNC.