Energy-efficient Networking in Wireless Ad Hoc Networks

A Dissertation

submitted to the designated
by the General Assembly of Special Composition
of the Department of Computer Science and Engineering
Examination Committee

by

Nikolaos Papanikos

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

University of Ioannina
March 2017

Advisory Committee:

- **Evangelos Papapetrou**, Assistant Professor, Dept. of Computer Science and Engineering, University of Ioannina
- **Apostolos Zarras**, Associate Professor, Dept. of Computer Science and Engineering, University of Ioannina
- **Fotini-Niovi Pavlidou**, Professor, Dept. of Electrical and Computer Engineering, Aristotle University of Thessaloniki

Examining Committee:

- **Evangelos Papapetrou**, Assistant Professor, Dept. of Computer Science and Engineering, University of Ioannina
- **Apostolos Zarras**, Associate Professor, Dept. of Computer Science and Engineering, University of Ioannina
- **Fotini-Niovi Pavlidou**, Professor, Dept. of Electrical and Computer Engineering, Aristotle University of Thessaloniki
- **Lisimachos-Paul Kondi**, Associate Professor, Dept. of Computer Science and Engineering, University of Ioannina
- **Christos Douligeris**, Professor, Dept. of Informatics, University of Piraeus
- **Vassilis Tsaoussidis**, Professor, Dept. of Electrical and Computer Engineering, Democritus University of Thrace
- **Panagiotis Sarigiannidis**, Assistant Professor, Dept. of Informatics and Telecommunications Engineering, University of Western Macedonia

Dedication

To my family and Andromachi.

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my advisor, Evangelos Papapetrou, who offered his continuous advice and encouragement throughout the course of this thesis. Especially, I would like to thank him for the systematic guidance and great effort he put into training me in the scientific field. During our long discussions, I have learned many valuable lessons about research and life as well. A special thank goes to Apostolos Zarras who was always there to help and encourage. I will surely miss our basketball sessions where he was making sure I was tired enough to stay away from office and have a healthy work-life balance. Apostolos, it has been a pleasure sharing the floor with you.

I would also like to thank the members of my examination committee, Fotini-Niovi Pavlidou, Lisimachos-Paul Kondi, Christos Douligeris, Vassilis Tsaoussidis and Panagiotis Sarigiannidis for their insightful and kind comments.

I am deeply grateful to my family for their continuous support, encouragement and kindness. I am particularly indebted to my parents Stergios and Anna, who always given me unconditional support in all of my endeavors. Their faith in me has always been a driving and influential force in my life. Also to my brother Dimitris and my uncle Giannis, who were always there for whatever I needed.

I am very grateful to have been surrounded by extraordinary collegues that made my graduate life pleasant and always challenging. Giorgos Margaritis, Andreas Vasilakis, Argyris Kalogeratos, Petros Manousis, Dimitri Melissovas, Panagiotis Giannakis, Alexandros Papadogiannakis, Thanasis Koufoulis, Vassileios Papadopoulos, Ioannis Papadopoulos, Dimitrios-Georgios Akestoridis and Nikolaos Kallimanis: Thank you all for the fruitful conversations, but especially for making easier the long working hours. I could not forget my close friend Vassilis Bourgos and my skype-mate Christos Theodorakis who helped me stay strong during tough times. For this I am profoundly grateful.

Over the last several years, I have also been supported by many friends outside University. I am especially grateful to Dimitris Karathanos, Alexandra Chatzieleftheriou, Giorgos Kouis, Aphrodite Mpitouni, Euthymis Papatheodorou, Manos and Nikos Bogdos, Fanis Gerodimos, Dimitra Tsintikidou, Abdallah Hasan, little Thomas and “princess” Aphrodite.

Last but not least, this thesis would not have been possible without Andromachi’s endless love, support and understanding. Through it all – the good, the bad, the ups and the downs – thank you for being there.

Table of Contents

List of Figures

List of Tables

List of Algorithms

Abstract

Nikolaos Papanikos, Ph.D., Department of Computer Science and Engineering, University of Ioannina, Greece, March 2017.
Energy-efficient Networking in Wireless Ad Hoc Networks.
Advisor: Evangelos Papapetrou, Assistant Professor.

Wireless multi-hop ad hoc networks are self-organizing networks that can be spontaneously deployed without any need of fixed infrastructure. In order to enable communication, network nodes share their resources to store and forward other nodes' data packets. However, the current hardware technology significantly limits the battery power network nodes run on. As a result, designing energy-efficient networking algorithms is of paramount importance for the viability of this type of networks. In the present thesis, we study networking algorithms that rely on packet redundancy to provide fair communication. This approach can significantly increase the number of transmissions and have a severe impact on the energy efficiency. Our main goal is to devise novel algorithms that efficiently handle packet redundancy in order to reduce the related energy costs without compromising the overall performance. We focus on two well-known fields; broadcasting in mobile ad hoc networks (MANETs) and routing in opportunistic networks (OppNets).

In the first part, we examine energy-efficient broadcasting in MANETs. The latest trend in this field combines traditional broadcast schemes with network coding. Besides enhancing the energy efficiency through the reduction of transmissions, this synergy also increases the resilience to loss and improves security. Initially, we focus on XOR-based broadcasting and reveal cases where the well-established approach suffers performance breakdowns. We attribute this behavior to an essential component of the underlying broadcast algorithm that is inherently incompatible with network coding. To tackle the problem, we introduce a novel coding-friendly broadcast algorithm. Furthermore, for the first time, we use XOR coding as a mechanism not only

for enhancing energy efficiency but also for reducing the end-to-end-delay. Through extensive simulations, we demonstrate the effectiveness of the proposed algorithm on improving the energy efficiency, delivery delay and utilization of network resources. Then, we focus on RLNC-based broadcasting and introduce an analytical model that captures the performance of coding-based broadcast schemes. We observe that the traditional approach to combine RLNC and probabilistic forwarding significantly impacts the performance of RLNC. To this end, we design a novel RLNC-based broadcast algorithm that for the first time applies RLNC over CDS-based broadcasting. The proposed algorithm provides a more systematic pruning of redundant transmissions without compromising RLNC's efficiency. We also investigate generation management that is a key issue in RLNC and introduce a new distributed scheme that is suitable for mobile environments. Finally, through extensive simulations, we show that the proposed algorithm outperforms XOR-based as well as RLNC-based schemes even when global knowledge is used for managing packet generations.

In the second part of the thesis, we investigate energy-efficient routing in OppNets. The prominent routing strategy in coping with intermittent connectivity of this type of networks is packet replication. Although this strategy maximizes the delivery efficiency, it can lead to the creation of an excessive number of replicas thus exhausting the limited energy resources of the network nodes. We introduce a simple yet efficient method which allows nodes to share information about the replication process in order to avoid unnecessary replication. The proposed approach comes at negligible cost and significantly increases the energy efficiency without sacrificing delivery rate. At the same time, our solution is generic in the sense that it can be implemented regardless of the utility metric used for making replication decisions. Additionally, we provide a lightweight extension based on Bloom filters that further improves the energy efficiency. In contrast to state-of-the-art, the proposed extension allows non-carrier nodes to play a more active role in the replication process and deny receiving redundant packet replicas. We validate the performance gains of our solutions through analysis as well as extensive simulations.

Finally, we examine some interesting topics that lie within the context of efficient routing in OppNets. These involve the implementation of an event-driven simulator for OppNets, the development of a paradigm for constructing large scale synthetic trace from real ones and the design of a congestion control algorithm that provides an effective trade-off between fairness and performance.

Εκτεταμενη Περιληψη

Νικόλαος Παπανίκος, Δ.Δ., Τμήμα Μηχανικών Η/Υ και Πληροφορικής, Πανεπιστήμιο Ιωαννίνων, Μάρτιος 2017.
Αλγόριθμοι Δικτύωσης για Εξοικονόμηση Ενέργειας σε Ασύρματα Αδόμητα Δίκτυα.
Επιβλέπων: Ευάγγελος Παπαπέτρου, Επίκουρος Καθηγητής.

Τα τελευταία χρόνια έχει παρατηρηθεί μια ραγδαία αύξηση στο πλήθος των φορητών συσκευών που χρησιμοποιούν την ασύρματη τεχνολογία για επικοινωνία. Ωστόσο, η συνδεσιμότητα μεταξύ των συσκευών αυτών είναι σχετικά περιορισμένη καθώς οι τρέχουσες τεχνικές δικτύωσης απαιτούν την ύπαρξη κατάλληλων δικτυακών υποδομών. Για το λόγο αυτό, η ερευνητική κοινότητα εστίασε στην ανάπτυξη μια νέας κατηγορίας δικτύων, γνωστά ως ασύρματα αδόμητα δίκτυα πολλών αλμάτων. Τα δίκτυα αυτά σχηματίζονται στιγμιαία σε οποιοδήποτε περιβάλλον με κάθε κόμβο να λειτουργεί ως δρομολογητής προωθώντας πακέτα των άλλων κόμβων.

Παρά τις σημαντικές τεχνολογικές εξελίξεις στο υλικό, οι φορητές συσκευές έχουν περιορισμένα αποθέματα ενέργειας καθώς τροφοδοτούνται από μπαταρίες μικρής διάρκειας ζωής. Επομένως, ένα από τα βασικά ζήτημα στα ασύρματα αδόμητα δίκτυα είναι η σχεδίαση αλγορίθμων δικτύωσης που είναι αποδοτικοί ως προς τη διαχείριση ενέργειας των κόμβων του δικτύου. Στην παρούσα διατριβή εξετάζουμε αλγορίθμους δικτύωσης που βασίζονται στη διασπορά πολλαπλών αντιτύπων κατά μήκος του δικτύου. Η χρήση πολλαπλών αντιτύπων έχει άμεσο αντίκτυπο στα αποθέματα ενέργειας των κόμβων του δικτύου καθώς μπορεί να αυξήσει σημαντικά το πλήθος των εκπομπών. Παρόλα αυτά η τεχνική αυτή χρησιμοποιείται κατά κόρον στην ευρεία εκπομπή στα κινητά αδόμητα δίκτυα (mobile ad hoc networks) και στη δρομολόγηση στα οπορτουνιστικά δίκτυα (opportunistic networks). Θέτουμε ως βασικό στόχο την εξοικονόμηση των αποθεμάτων ενέργειας των κόμβων του δικτύου εξασφαλίζοντας την ίδια στιγμή την αποδοτική επικοινωνία μεταξύ τους. Προς αυτή την κατεύθυνση, προτείνουμε καινοτόμους αλγορίθμους δικτύωσης που πετυχαίνουν τον παραπάνω στόχο ελαττώνοντας σημαντικά το πλήθος των εκπομπών.

Στο πρώτο τμήμα της παρούσας διατριβής εξετάζουμε αλγορίθμους ευρείας εκπομπής σε κινητά αδόμητα δίκτυα. Η ευρεία εκπομπή δίνει τη δυνατότητα στους κόμβους να στέλνουν πακέτα σε όλους τους άλλους κόμβους του δικτύου και αποτελεί βασικό μηχανισμό δικτύωσης καθώς χρησιμοποιείται εκτενώς κατά τη δρομολόγηση, την αναζήτηση υπηρεσιών και πόρων κ.α. Ακολουθούμε τη σύγχρονη τάση που συνδυάζει παραδοσιακούς αλγορίθμους εκπομπής με τη μέθοδο της κωδικοποίησης δικτύου για την αύξηση της εξοικονόμησης ενέργειας, της αξιοπιστίας και της ασφάλειας. Ειδικότερα, μελετάμε δύο βασικές κατηγορίες αλγορίθμων εκπομπής με κωδικοποίηση δικτύου που διαφέρουν ως προς την τεχνική κωδικοποίησης που εφαρμόζουν. Η πρώτη χρησιμοποιεί την τεχνική XOR, ενώ η δεύτερη βασίζεται στην τυχαία γραμμική κωδικοποίηση δικτύου (random linear network coding).

Οι αλγόριθμοι εκπομπής που χρησιμοποιούν την τεχνική XOR βασίζονται στην ικανότητα των ενδιάμεσων κόμβων να «κρυφακούν» τα πακέτα που λαμβάνουν οι γειτονικοί τους κόμβοι και να εντοπίζουν ευκαιρίες κωδικοποίησης. Αρχικά, παρουσιάζουμε μια πειραματική αξιολόγηση του σημαντικότερου αλγορίθμου εκπομπής αυτής της κατηγορίας. Τα αποτελέσματα της αξιολόγησης αυτής αποκαλύπτουν περιπτώσεις όπου η επικρατούσα μέθοδος εκπομπής με κωδικοποίηση XOR παρουσιάζει σημαντικά προβλήματα απόδοσης. Εντοπίζουμε τα αίτια του προβλήματος στον τρόπο που η κωδικοποίηση δικτύου συνδυάζεται με παραδοσιακούς αλγορίθμους εκπομπής. Πιο συγκεκριμένα, δείχνουμε ότι υπάρχουν προβλήματα συμβατότητας ανάμεσα στην κωδικοποίηση δικτύου και ένα στοιχειώδη μηχανισμό των παραδοσιακών πρωτοκόλλων εκπομπής. Προκειμένου να επιλύσουμε το πρόβλημα αυτό, επανασχεδιάζουμε τον παραπάνω μηχανισμό ώστε να είναι πλήρως συμβατός με την κωδικοποίηση δικτύου. Στη συνέχεια, προτείνουμε αλλαγές στον τρόπο που εφαρμόζεται η κωδικοποίηση δικτύου βελτιώνοντας σημαντικά την εξοικονόμηση ενέργειας και τη διαχείριση των πόρων του δικτύου. Επιπλέον, εισάγουμε τη τεχνική του «κωδικοποιημένου πλεονασμού», η οποία, για πρώτη φορά, αξιοποιεί την κωδικοποίηση XOR όχι μόνο για τη μείωση των μεταδόσεων αλλά και την ελάττωση της καθυστέρησης στην παράδοση πακέτων.

Σε αντίθεση με την κωδικοποίηση XOR, η τυχαία γραμμική κωδικοποίηση δικτύου κωδικοποιεί τα πακέτα από άκρο σε άκρο υπό την έννοια ότι οι ενδιάμεσοι κόμβοι δεν είναι αναγκαίο να αποκωδικοποιήσουν πλήρως τα πακέτα που λαμβάνουν και να τα επανακωδικοποιήσουν. Στα πλαίσια της παρούσας διατριβής, παρουσιάζουμε ένα αναλυτικό μοντέλο που αξιολογεί την απόδοση των αλγορίθμων εκπομπής που εφαρμόζουν κωδικοποίηση δικτύου για την εξοικονόμηση ενέργειας. Χρησιμοποιώντας το προτεινόμενο μοντέλο, δείχνουμε ότι η τρέχουσα προσέγγιση που συνδυάζει την τυχαία γραμμική κωδικοποίηση δικτύου με πιθανοτικούς αλγορίθμους εκπομπής είναι προβληματική. Πιο συγκεκριμένα, υπάρχουν περιπτώσεις όπου η πιθανοτική διασπορά των κωδικοποιημένων πακέτων μπορεί να μειώσει σημαντικά την αποτελεσματικότητα της γραμμικής κωδικοποίησης. Βάσει του αποτελέσματος αυτού, προτείνουμε ένα καινοτόμο αλγόριθμο ευρείας εκπομπής που μεγιστοποιεί τα οφέλη της τυχαίας γραμμικής κωδικοποίησης δικτύου εφαρμόζοντάς την πάνω από ένα ντετερμινιστικό αλγόριθμο προώθησης των πακέτων ειδικά σχεδιασμένο για τη μείωση των περιττών μεταδόσεων. Επιπλέον, παρουσιάζουμε μια επέκταση του προτεινόμενου αλγορίθμου που αυξάνει την αξιοπιστία της εκπομπής στους κόμβους με χαμηλή συνδεσιμότητα. Τέλος, μελετάμε ένα βασικό ζήτημα για την εύρυθμη λειτουργία της γραμμικής κωδικοποίησης δικτύου. Το ζήτημα αυτό σχετίζεται με το διαχωρισμό των εισερχομένων πακέτων στο δίκτυο σε ομάδες σύμφωνα με τις οποίες δημιουργούνται κωδικοποιημένα πακέτα. Η συγκρότηση και διαχείριση αυτών των ομάδων αποτελεί δύσκολο πρόβλημα σε ένα κατανεμημένο περιβάλλον όπως αυτό των ασύρματων αδόμητων δικτύων. Προς αυτή την κατεύθυνση, σχεδιάζουμε ένα νέο κατανεμημένο αλγόριθμο ομαδοποίησης πακέτων που είναι κατάλληλος για δίκτυα κινούμενων κόμβων.

Στο δεύτερο τμήμα της παρούσας διατριβής μελετάμε αλγορίθμους δρομολόγησης σε οπορτουνιστικά δίκτυα στα οποία η συνδεσιμότητα μεταξύ των κόμβων είναι διακοπτόμενη. Στα δίκτυα αυτά η επικρατούσα στρατηγική δρομολόγησης βασίζεται στη διασπορά πολλαπλών αντιτύπων κατά μήκος του δικτύου (replication-based routing). Σε αντίθεση με τα κινητά αδόμητα δίκτυα, τα μονοπάτια από άκρο σε άκρο στα οπορτουνιστικά δίκτυα σχηματίζονται δυναμικά στο χρόνο. Ως αποτέλεσμα, η δρομολόγηση επιτυγχάνεται επιτρέποντας σε κάθε κόμβο να αποθηκεύει προσωρινά πακέτα και να τα προωθεί στις μελλοντικές του επαφές με άλλους κόμβους. Η χρήση πολλαπλών αντιτύπων κατά τη δρομολόγηση αυξάνει σημαντικά την πιθανότητα εύρεσης ενός μονοπατιού ανάμεσα στον αποστολέα και τον παραλήπτη. Ωστόσο, η προσέγγιση αυτή εισάγει επιπρόσθετα ζητήματα που σχετίζονται με την ευέλικτη διαχείριση του πλήθους των αντιτύπων στο δίκτυο. Η άσκοπη δημιουργία αντιτύπων έχει σημαντικό αντίκτυπο στους περιορισμένους πόρους των κόμβων, όπως η αύξηση της κατανάλωσης ενέργειας και η αλόγιστη χρήση του αποθηκευτικού χώρου. Από την άλλη πλευρά, η υπερβολική μείωση των αντιτύπων μπορεί να ελαττώσει σημαντικά την απόδοση της δρομολόγησης.

Προκειμένου να βελτιώσουμε την εξοικονόμηση ενέργειας κατά τη δρομολόγηση στα οπορτουνιστικά δίκτυα παρουσιάζουμε ένα καινοτόμο αλγόριθμο δρομολόγησης πολλαπλών αντιτύπων. Παρόμοια με τους πιο αποδοτικούς αλγόριθμους της βιβλιογραφίας, ο προτεινόμενος αλγόριθμος βασίζεται σε ειδικές μετρικές (utility metrics) οι οποίες ποσοτικοποιούν τη χρησιμότητα κάθε κόμβου στη δρομολόγηση χρησιμοποιώντας τοπικές πληροφορίες, όπως για παράδειγμα το ιστορικό των επαφών του. Με αυτό τον τρόπο, η δημιουργία ενός νέου αντιτύπου πραγματοποιείται δυναμικά όταν ένας κόμβος που έχει αποθηκευμένο ένα αντίτυπο έρχεται σε επαφή με κάποιον άλλο κόμβο με μεγαλύτερη μετρική. Για τη μείωση των περιττών αντιτύπων περιορίζουμε τη δημιουργία νέου αντιτύπου μόνο στις περιπτώσεις που ο κόμβος σε επαφή έχει καλύτερη μετρική δρομολόγησης από όλους τους κόμβους που διατηρούν ένα αντίτυπο. Προς αυτή την κατεύθυνση, ο προτεινόμενος αλγόριθμος αξιοποιεί τις επαναλαμβανόμενες επαφές μεταξύ των κόμβων που διατηρούν αντίτυπα για να συγχρονίσει την εικόνα τους σχετικά με την κατάσταση της δρομολόγησης. Επιπλέον, ο προτεινόμενος αλγόριθμος δίνει έναν πιο ενεργό ρόλο στους κόμβους που δεν έχουν λάβει κάποιο αντίτυπο. Πιο συγκεκριμένα, επιτρέπει στους κόμβους αυτούς να αναγνωρίζουν πακέτα για τα οποία κρίθηκαν ακατάλληλοι στο παρελθόν και να ακυρώνουν τη δημιουργία νέων αντιτύπων. Για την υλοποίηση αυτών των μηχανισμών προτείνουμε ευέλικτες δομές που χαρακτηρίζονται από μικρό κόστος αποθήκευσης και προσφέρουν γρήγορη αναζήτηση και αποθήκευση.

Τέλος, παρουσιάζουμε λύσεις για επιμέρους ζητήματα που αντιμετωπίσαμε στα πλαίσια της έρευνας στα οπορτουνιστικά δίκτυα. Το πρώτο αφορά την αξιολόγηση της απόδοσης των αλγορίθμων δρομολόγησης μέσω ρεαλιστικών προσομοιώσεων, ενώ το δεύτερο σχετίζεται με την ανάγκη για έλεγχο της συμφόρησης. Περιγράφουμε συνοπτικά τις προτεινόμενες λύσεις που περιλαμβάνουν την ανάπτυξη ενός νέου προσομοιωτή για οπορτουνιστικά δίκτυα, μια μέθοδο κατασκευής ρεαλιστικών συνθετικών οπορτουνιστικών δικτύων μεγάλης κλίμακας και έναν καινοτόμο αλγόριθμο ελέγχου συμφόρησης που μπορεί εύκολα να συνδυαστεί με οποιοδήποτε πρωτόκολλο δρομολόγησης που χρησιμοποιεί ένα αντίτυπο.

Chapter 1

Introduction

1.1 Scope of thesis

Wireless networks have been thoroughly studied over the past four decades. Today more than ever before, we witness the results of the research in this field, since users are connected wirelessly around the clock using a diverse range of devices, e.g., laptops, tablets and smartphones. However, in conventional wireless networks, network connectivity is still limited in the sense that it depends in the existence of a fixed infrastructure, such as an access point or a base station. To provide wireless communication when fixed infrastructure is absent, the research community introduced the concept of wireless ad hoc networks. In this class of networks, user devices form a self-organizing network that requires minimum user intervention and it is deployed easily with minimal cost and planning. Communication is direct when nodes are within each other's range, otherwise nodes rely on other nodes to forward their packets. In the latter case, network nodes act as packet relays enabling multi-hop communication.

Despite their unique features, wireless ad hoc networks have yet to make the transition to the commercial world. Most real-world deployments are ephemeral and are built to provide short-term communication. Some of the application areas include battlefield communications in military operations, search and rescue operations during a disaster, data gathering of environmental conditions in hostile environments and communication for educational reasons in classrooms, campuses and conferences. Recently, the growing popularity of the concept of "Internet of Things" (IoT) [2] is paving the way for commercially viable wireless ad hoc networks. The main idea is to extend Internet connectivity beyond traditional devices like personal computers, smartphones and tablets to a diverse range of devices and everyday things that will communicate and interact with the external environment. Towards this direction, wireless ad hoc networks can play an important role in enabling the communication between things/devices and relaying data traffic to the Internet infrastructure [3, 4]. Another area of deploying wireless ad hoc networks is when censorship disrupts or filters conventional communication networks. In these cases, the distributed nature and adaptability of ad hoc networks render them perfect to promote free speech and allow public communication [5]. Furthermore, wireless ad hoc networks can be combined with the existing infrastructure in order to extend the network coverage, capacity and scalability. Especially, there is an increasing research interest in extending the capacity of cellular infrastructure through wireless ad hoc networks [6]. Such approaches will have a great impact on the current carrier networks which are overloaded by high traffic demands.

Over the past years the research community introduced a diverse range of protocols that enable communication in wireless ad hoc networks. Multiple networking protocols were proposed able to adapt to the network conditions in a distributed fashion without using any kind of central synchronization. However, there still exist open issues that need to be solved. One main challenge is related to the energy costs of the networking protocols. Network nodes run on battery power which, despite the major advances in battery technology, is currently limited for mobile devices. Failing to provide energy-efficient networking could lead to situations where devices disjoin the network because either their battery was fully drained or the users were disappointed from the over-utilization of their devices. In these situations, the number of nodes can be reduced to critical levels, severely affecting the network stability.

The main goal of this thesis is to improve the energy efficiency of networking algorithms in wireless ad hoc networks. We focus on two well-known classes of wireless networks; mobile ad hoc networks (MANETs) [7] and opportunistic networks (OppNets) [8]. Both include mobile nodes that move freely to any direction and communicate arbitrarily among each other. MANETs assume that an end-to-end path always exists between each pair of nodes, while OppNets can be seen as a generalization of MANETs because the aforementioned assumption is relaxed. In both types of networks, we study the implications of generating and handling more than one packet instances to the energy efficiency of networking algorithms. Especially, we focus on broadcasting in MANETs and routing in OppNets where using multiple packet duplicates is the only way to achieve an acceptable performance. We investigate both fields and introduce novel schemes that are energy-efficient in the sense that they significantly reduce the number of transmissions (replications) without compromising the delivery and delay performance. Minimizing transmissions is crucial for the energy consumption at intermediate nodes since data transmission/reception is known to be the most energy-consuming operation in wireless devices.

1.2 Contributions

We initially examine the problem of energy-efficient broadcasting in MANETs. Broadcasting is an essential networking component that allows nodes to send a packet to all other nodes in the network [9–13]. It is extensively used when nodes engage in discovery phases including on-demand routing protocols for constructing a path [14,15], service discovery applications for finding a resource [16, 17] and peer databases for retrieving volatile data [18]. Since broadcasting significantly affects the performance of other networking mechanisms, using energy-efficient approaches is of paramount importance. We focus on a new research direction towards energy-efficient broadcasting that combines conventional approaches with network coding [19]. Coding-based schemes "mix" two or more packets into one minimizing the total number of the required transmissions, increasing the resilience to transmission errors and enhancing security. Based on the coding technique, two approaches can be identified; XOR-based coding [20, 21] and Random Linear Network Coding (RLNC) [22, 23].

XOR-based approaches encode packets on a hop-by-hop basis using bitwise XOR and then forward them using a Connected Dominating Set (CDS) based broadcasting algorithm [24–27]. Although this strategy has been proved successful, we bring to light several occasions where its performance severely degrades and the coding gain becomes negligible. Motivated by this finding, we examine in depth the synergy of network coding and the underlying broadcast algorithm and reveal that the weak link is a component of the baseline algorithm known as "the termination criterion". We introduce a new XOR-based broadcasting algorithm which incorporates a novel termination criterion fully compatible with XOR coding. Moreover, we revisit the coding internals in order to enhance the overall performance in terms of energy efficiency, delivery delay and utilization of network resources.

RLNC-based approaches operate on an end-to-end basis in the sense that intermediate nodes are not required to fully decode and re-encode the encoded packets. Initially, generated packets are grouped in the so called "generations". Encoded packets are produced as random linear combinations of the packets in a generation, based on the theory of finite fields [28, 29], and then probabilistically forwarded. We introduce an analytical model that captures the performance of coding-based broadcast schemes that focus on energy efficiency. The model reveals that pruning transmissions, which is an essential process for energy efficiency, has a significant impact on the effectiveness of RLNC. Therefore, we introduce a new RLNC-based broadcasting algorithm that follows the innovative approach of integrating RLNC into deterministic broadcasting. In contrast to the current approach that is based on probabilistic forwarding, the proposed algorithm provides a more systematic and topology-aware pruning of redundant transmissions without impairing the coding efficiency of RLNC. Furthermore, we address the problem of generation management, i.e., the need of nodes to distributively agree in the grouping of packets into generations, and propose a realistic distributed scheme that does not compromise the coding efficiency of RLNC.

The second part of this thesis focuses on energy-efficient routing in OppNets [8]. In these networks the connectivity between nodes is intermittent, rendering traditional networking protocols inefficient. In order to tackle the absence of end-to-end paths, the "store-carry-and-forward" communication model [1, 30] was introduced. Nodes store data packets until they come in contact with other network nodes. One of the dominant approaches to enhance the routing efficiency is packet replication [1, 31]. Controlling the level of replication allows for a trade-off between delivery rate and energy efficiency. To this end, utility-based replication [32–35] takes advantage of a utility metric that captures the fitness of a node for delivering and/or forwarding the packet and then creates replicas by comparing the utility value of the nodes in contact. A more efficient approach is utilized by Delegation Forwarding [36] which performs replication only when the node in contact has a better utility value than the one perceived as the highest in the network.

Despite the previous research efforts, in several cases replication-based routing still creates an excessive number of replicas exhausting the node energy resources. We make the observation that packet redundancy could be reduced if nodes were allowed to share information about the state of the replication process in order to use it as a whole when making replication decisions. We design an energy-efficient routing algorithm that incarnates the aforementioned functionality with minimal cost, and achieves a significant reduction of replications minimizing the energy consumption at intermediate nodes. Moving a step further, we enhance the routing process by utilizing non-carriers, i.e., nodes which do not carry a packet replica. In contrast to the current approach where non-carriers are always willing to receive any replica, our proposed algorithm allows these nodes to deny receiving replicas for which they were previously rejected.

The main contributions of this thesis can be summarized as follows.

In the field of broadcasting in MANETs:

- We unveil the shortcomings in the synergy between XOR coding and CDS-based broadcasting. After analyzing the reasons of this finding, we propose a coding-friendly termination criterion for the CDS-based algorithm and illustrate its efficiency.

- We delineate a novel method for detecting coding opportunities in XOR-based broadcasting. Our method is lightweight in the sense that, in contrast to current approaches, requires each node to maintain minimum state while it mostly utilizes information that is already available through the underlying broadcast mechanism.

- We enhance the pruning efficiency of the underlying CDS-based algorithm by exploiting information available from the coding mechanism, i.e., we establish a bidirectional synergy between XOR-based network coding and CDS-based forwarding.

- We address the problem of increased end-to-end delay in XOR-based broadcasting. We explore the causes and propose a solution that uses XOR coding for achieving a cost-free increase of packet redundancy across the network in order to reduce end-to-end delay.

- We develop an analytical model that sheds light on the differences between RLNC and XOR based broadcast schemes oriented towards energy efficiency.

- We unveil the potential pitfalls of combining RLNC and probabilistic forwarding. Then, we turn to deterministic broadcasting which has never been used for pruning transmissions in the context of an RLNC enabled scheme.

- We address the problem of generation management that is critical in RLNC when packets from different sources are "mixed" into a generation (inter-source coding). We review the pending problems and propose a realistic distributed solution that is lightweight and does not compromise the coding efficiency of RLNC.

- We show that an increased packet loss rate significantly impairs the performance of RLNC in nodes experiencing poor connectivity. This holds true even if deterministic broadcasting is used. To tackle the problem, we extend the proposed algorithm in order to enhance the topology-awareness of the pruning process.

In the field of routing in OppNets:

- We devise a new replication strategy for energy-efficient routing in opportunistic networks. The proposed strategy allows network nodes to coordinate their views regarding the replication process and exploit this information to drastically prune replications.

- We demonstrate that nodes which do not carry a replica can play an active role in the replication process. To this end, we extend our proposed algorithm by allowing these nodes to exploit their past rejections in order to deny receiving future replicas.

Furthermore, the author of this thesis participated in the examination of additional research challenges in OppNets closely related to the context of this thesis. This research resulted in the following contributions:

- An event-driven simulator for OppNets that operates in a contact basis and is capable of processing real-world traces as well as synthetic ones. Our simulator includes a plethora of routing protocols, while it also provides several congestion control algorithms, packet deletion mechanisms and buffer management policies.
- A new paradigm for constructing synthetic traces from real ones that is able to produce contact traces that exhibit a configurable degree of separation and a series of characteristics observed in real-world human traces.
- A novel congestion control algorithm that provides an effective trade-off between fairness and routing performance based on the social ties of the network nodes. The proposed algorithm can be easily incorporated into virtually any utility-based routing protocol.

1.3 Outline

The remainder of this thesis is organized as follows.

In Chapter 2, we provide the background required to understand the communication challenges that arise in MANETs and OppNets. We review the problem of broadcasting in MANETs focusing on energy-efficient algorithms that combine traditional broadcast approaches with network. Then, we present the store-carry-and-forward paradigm introduced to tackle the problem of intermittent connectivity in OppNets and review state-of-the-art routing approaches.

In Chapter 3, we experimentally demonstrate that the common approach to incorporate XOR-based coding into broadcasting suffers from performance breakdowns. After unveiling the problem source, we introduce a novel broadcast algorithm that combines XOR-based coding with deterministic broadcasting in the most beneficial way.

In Chapter 4, we develop an analytical model which reveals the shortcomings of building RLNC on top of probabilistic broadcasting. Motivated by this finding, we introduce a novel broadcast algorithm that combines RLNC with deterministic broadcasting successfully improving both energy efficiency and error resilience in the same time.

In Chapter 5, we examine multi-copy routing in OppNets. We demonstrate that the number of redundant replications, and, thus, energy consumption, could be significantly reduced if (i) packet carriers cooperatively share information about the replication state, and (ii) non-carriers are allowed to refrain from receiving replicas for which they were previously rejected.

In Chapter 6, we discuss some interesting issues that appear in OppNets. We provide our solutions towards a more realistic performance evaluation of routing algorithms through simulation and we present a congestion control algorithm that can be easily applied to utility-based routing.

In Chapter 7, we present some promising directions for future work and summarize the basic conclusions of this thesis.

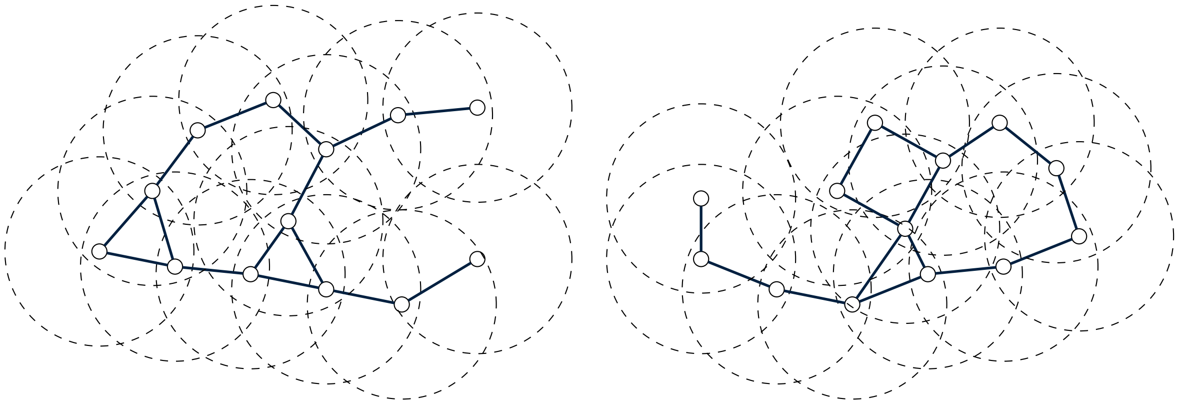
Chapter 2

Background and related work

The key contributions of this thesis span in two classes of wireless ad hoc networks, namely mobile ad hoc networks (MANETs) and opportunistic networks (OppNets). One of the main challenges in both classes is to improve the energy efficiency of the networking algorithms that provide communication. The need for improving energy efficiency is more evident in the cases of broadcasting in MANETs and routing OppNets. This is because the prominent algorithms in both fields use the same energy-consuming approach. Multiple packet instances are spread throughout the network producing an excessive number of transmissions. In this chapter, we first provide the fundamental background in MANETs and OppNets. Then, we present state-of-the-art of energy-efficient broadcasting in MANETs and routing in OppNets, respectively.

2.1 Mobile ad hoc networks

MANETs can be easily deployed at any place and time, without the need of infrastructure or specialized devices. In order to enable communication between the



(a)

(b)

Figure 2.1: Snapshot of a mobile ad hoc network at time instances: (a) t, and (b) $t' > t$

participating nodes, this type of ad hoc network adapts the “give to get” model. According to this model, each network node is willing to act as a router and relay packets of other nodes, while, at the same time, it exploits its connection with other nodes to send data to distant destinations. In other words, each network node trades its resources, i.e., energy, memory and bandwidth, for the ability to communicate. The only limitation in this type of network is that an end-to-end path should always exist between the two communicating nodes. In case that the networking path between a pair of nodes breaks temporarily, ongoing traffic gets lost never reaching the destination. Fig. 2.1a illustrates a snapshot of a MANET in a specific time instance. Each node has at least one or more direct links with another nodes that lie within its transmission range. In this way, at least one end-to-end path is present between all node pairs. As nodes move without restrictions, network topology and end-to-end paths change dynamically. Therefore, as time passes by, the network of Fig. 2.1a transforms to the one depicted in Fig. 2.1b.

Enabling communication in such environments is very challenging. Networking protocols should dynamically adapt to the network conditions in a distributed fashion without using any kind of central synchronization. On top of that, routing decisions should take into account the limited network resources. Communication paths should be discovered efficiently in terms of energy consumption. This is because network nodes run on battery power which is currently limited for mobile devices. Networking protocols must be energy-efficient since weakly handling the energy resources could be fatal for the network’s stability.

An essential networking component of MANETs is network-layer broadcasting, also know as network-wide broadcasting. This mechanism allows a network node to send a data packet to all other nodes that exist across the network [9–13]. Besides application data, broadcast protocols are also utilized to distribute control information to every network node. The latter is very useful in situations where network nodes engage in discovery phases, e.g., in on-demand routing protocols for constructing a path [14, 15], in service discovery applications for finding a resource [16, 17] and in peer databases for retrieving volatile data [18]. Since broadcasting indirectly affects the performance of other networking components using efficient broadcast techniques is of paramount importance. In the following, we present an overview of the long-established broadcast schemes focusing on the energy-efficient ones. Initially, we describe the most well-known traditional broadcast schemes proposed in the literature. Then, we present a new research direction that combines conventional broadcasting with the network coding technique [19].

2.1.1 Traditional broadcast approaches

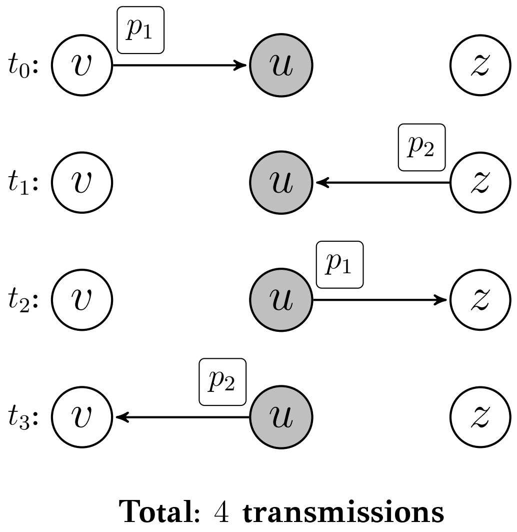
All broadcast schemes take advantage of the broadcast nature of the wireless medium, i.e., a single transmission at any node simultaneously reaches multiple receivers that lie in the sending node's transmission range. This operation, known as physical-layer or single-hop broadcast, minimizes the number of transmissions improving the utilization of the network resources and subsequently increases the network's throughput. In order to differentiate between the types of broadcasting, we will refer to this operation by single-hop broadcast for the reminder of this thesis. On the contrary, the terms broadcasting and broadcast will refer to network-wide broadcasting.

The simplest broadcast approach that operates in a distributed fashion is flooding. In flooding, each network node forwards every broadcast packet exactly once. However, flooding produces a large number of redundant transmissions that cause the *broadcast storm problem* [37]. In particular, the excessive number of transmissions inflates network congestion to critical levels and increases the number of packet collisions. To mitigate the effects of the broadcast storm problem the research community focused on probabilistic and deterministic broadcast approaches. Both select a subset of the network nodes, called *forwarders*, to relay broadcast packets. According to the probabilistic approach [13], each node relays a broadcast packet based on a probability p. The value of p can be either predefined or dynamically assigned based on network condition, e.g., the neighborhood size. The main drawback of this category is that choosing the proper p at each node is hard. Large values lead to high packet redundancy, while small ones degrade the delivery efficiency. On the other hand, the deterministic approach, also known as CDS-based broadcasting, constructs a connected dominating set (CDS) [24] of the network in a distributed fashion. The nodes constituting the CDS are the forwarders, while all other nodes act as passive receivers. Finding a minimal connected dominating set (MCDS) is proven to be NP-hard. As a result, energy-efficient deterministic broadcast approaches target at minimizing the number of forwarders that disseminate the broadcast packets across the network.

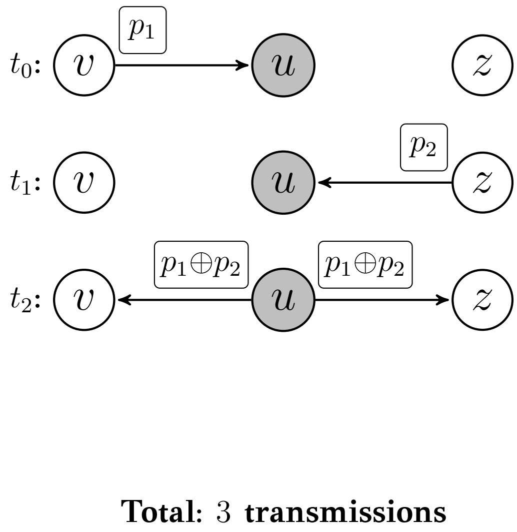
Deterministic broadcast approaches can be classified into three broad categories. In the first, a CDS of the network is locally built using local topology information, i.e., 1-hop and 2-hop neighborhood. The computed CDS is used to forward broadcast packets throughout the network with packets stemming from different sources using the same CDS. Most algorithms in this category differentiate on the heuristics used for constructing the CDS [38–44]. Algorithms in the second category again locally build a CDS that is common to every network node but use additional dynamic rules based on broadcast state information to reduce the initial CDS. More specifically, dynamic rules usually exploit reception of packet duplicates to compute nodes that already received the packet and enhance the pruning process [25, 45–50]. Other algorithms in this category focus on reliability either by introducing packet acknowledgements [49, 51] or by modifying the construction of the CDS [52]. Finally, the third category follows a different approach. Instead of building an initial CDS and then pruning it, the algorithms in this category construct a source-specific CDS on a hop-by-hop basis as packets are spread throughout the network [26, 27, 53]. The CDS is constructed using both local topology information, i.e, 1-hop and 2-hop neighbors, and broadcasting state information obtained through packet duplicates. Algorithms that focus on reliability also exist in this category [54].

2.1.2 Broadcasting using XOR network coding

In the field of coding-based broadcast the proposed algorithms can be classified into: i) energy efficient [21, 23, 55–62], and ii) delivery guarantee [63–74] approaches. The first category aims at striking the best possible trade-off between energy expendi-



(a)



(b)

Figure 2.2: Energy cost in terms of number of transmissions when using: (a) the traditional approach, and (b) the XOR-based approach.

ture (usually expressed by the number of transmissions) and delivery efficiency. On the other hand, the second category targets at 100% packet delivery and treats the minimization of the related costs as a second priority task.

Broadcasting approaches that utilize XOR network coding either for energy efficiency [21, 55–59] or reliability [63–66] build on the same primitives. They follow the concept of coding opportunity [20] to combine multiple packets into an encoded one using bitwise XOR operations. More specifically, the forwarding nodes exploit coding opportunities to produce encoded packets composed of original ones. Then, every neighboring node that receives an encoded packet decodes it in order to retrieve the original packets. The coding process is performed on a hop-by-hop basis, i.e., decoding takes place at every hop. However, encoding is not always possible since a coding opportunity arises only when all receiving nodes are able to decode. The common approach to increase the number of coding opportunities is to use a specialized buffering scheme, called RAD mechanism, that introduces a short delay before the actual transmission of each packet. Fig. 2.2 presents a well known example that demonstrates the benefits of XOR network coding. In this example, the forwarding node u transmits the encoded packet $p' = p_1 \oplus p_2$ (Fig. 2.2b) instead of transmitting p_1 and p_2 separately (Fig. 2.2a). On the receivers' side, node v decodes p_2 by computing $p' \oplus p_1$ while node z retrieves p_1 by computing $p' \oplus p_2$. Overall, the total number of transmissions reduces by one directly improving the energy efficiency. In the same time, the reliability increases indirectly due to the reduction in the number of transmissions that results in fewer packet collisions.

The primary focus of this thesis is on energy efficient broadcasting. The prominent algorithm of this category, CodeB [21], combines CDS-based broadcasting with XOR network coding. It also provides information exchange mechanisms that make possible the implementation in mobile environments. CodeB builds on top of the non coding Partial Dominant Pruning (PDP) [27] scheme, which is actually a CDS-based broadcast algorithm. However, it can be directly applied to other CDS-based approaches. Wang et al. explore the benefits of employing XOR network coding on various underlying CDS-based broadcast schemes [55,56]. Moreover, the use of XOR coding over PDP [27] and MPR [25], two typical CDS based algorithms, in tactical networks has been studied in [57]. In [58] the authors study the problem of broadcasting with deadlines in static ad hoc networks and propose alternative buffering schemes for the RAD mechanism. Finally, Yang and Wu [59] explore the benefits of XOR coding in energy efficient broadcasting when combined with directional antennas. In the second major category of XOR -based broadcast algorithms the focus is on guaranteeing 100% reliability. The algorithms of this category adopt a rateless approach [63–66]. More specifically, they keep producing encoded packets until all receivers are capable of decoding the initial packets. As a result, these schemes require feedback information. However, implementing a feedback mechanism is not straightforward in mobile networks. Therefore, most algorithms of this category are limited to static networks.

In chapter 3, we revisit the XOR-based broadcast approach that targets at enhancing the energy efficiency in mobile ad hoc networks. In particular, we demonstrate that the common approach, which is to benefit from the synergy of XOR network coding with a CDS-based broadcast algorithm, suffers performance breakdowns. Then, we propose a novel XOR-based algorithm that works cooperatively with the underlying broadcast scheme and revises the coding internals in order to enhance the broadcasting performance in terms of delivery delay, energy efficiency and network resources utilization.

2.1.3 Broadcasting using random linear network coding

Random linear network coding (RLNC) is based on the observation that a linear code, i.e. to linearly combine packets based on the theory of finite fields, is adequate for providing the benefits of network coding [28]. Similar to XOR-based schemes, RLNC-based algorithms can be classified into: i) energy efficient [23, 60, 61], and ii) delivery guarantee [67–73] approaches.

All RLNC-based algorithms [23, 60, 61, 67–74] build on the concepts of practical RLNC [22]. First, all non-encoded packets, known as native, are organized in groups, the so called *generations* [22]. Then, an encoded packet is produced as a linear combination of the native packets in a generation, using $\mathbb{F}_{2^s}$ arithmetic. In contrast to XOR-based approaches, RLNC-based algorithms operate on an end-to-end basis, i.e., decoding of encoded packets is required only at the communication end points. Decoding packets of generation i at node v is performed by means of a decoding matrix $\mathbb{G}_{v,i}$. The matrix is populated by *innovative packets*, i.e., the encoded packets that increase the rank of $\mathbb{G}_{v,i}$. The final decoding is accomplished by performing the Gaussian elimination when $\mathbb{G}_{v,i}$ has a full rank. It is also possible to decode a subset of packets when a full rank submatrix of $\mathbb{G}_{v,i}$ exists (*partial decoding*). Furthermore, encoding at an intermediate node is possible without the need of decoding the native packets since a new encoded packet may be produced by linearly combining other encoded packets that reside inside $\mathbb{G}_{v,i}$.

All RLNC-based algorithms that focus on energy efficiency [23, 60, 61] take a probabilistic approach to forward encoded packets. The prominent algorithm of this category described in [23] extends the probabilistic algorithm proposed in [60]. More specifically, it introduces two topology-aware heuristics to determine the number of encoded packets, that each node should forward, in order for the receivers to decode the original packets. This algorithm also allows the encoding of packets from different sources by incorporating rules for the distributed management of packet generations. Other techniques extend this algorithm by modifying the forwarding heuristics and the generation management mechanism [61]. The second category of RLNC-based algorithms [67–73] focuses on reliability and integrates some kind of feedback mechanism. The feedback information is used to determine the optimal rate, i.e., the number of packets to be forwarded by intermediate nodes, so that delivery of packets is guaranteed. Clearly, this strategy is not oriented towards minimizing the energy consumption of broadcasting. Furthermore, a feedback mechanism increases the cost while its implementation is not straightforward in mobile networks. Therefore, those algorithms have only been proposed for static networks. Finally, in [74], the authors study the problem of timeliness in broadcasting. They use RLNC over broadcasting trees in a static network and under the assumptions of lossless links and knowledge of global information.

In chapter 4, we introduce a novel RLNC-based algorithm that uses the synergy of RLNC and deterministic broadcasting to improve both resilience to failures and energy efficiency. The deterministic algorithm not only forwards packets but also dynamically determines the number of transmissions through a pruning process.

2.2 Opportunistic networks

Another type of wireless ad hoc networks that fully supports node mobility is the class of Opportunistic networks (OppNets). This class shares a lot of common characteristics with MANETs, e.g., nodes move freely to any direction and communicate among each other without any need of infrastructure or specialized hardware. Structurally, they differentiate only in a single aspect, i.e., the connectivity among network nodes is intermittent. This differentiation induces radical changes in the way communication is performed. Network is divided in multiple connected sub-networks, known as connectivity islands, which consist of one or more nodes. In most cases, network nodes are isolated communicating sparsely with other nodes in pairs when moving in the communication range of each other due to mobility. It is clear that in this type of network traditional networking protocols, designed for MANETs, fail to provide communication.

2.2.1 The store-carry-and-forward paradigm

To enable communication between network nodes, routing protocols in OppNets rely on node mobility building on top the store-carry-and-forward paradigm [1, 30]. According to this technique, each network node stores data packets and keeps them until a forwarding opportunity arises. A forwarding opportunity occurs when two nodes come in contact, meaning that they lie in the transmission range of each other, and are able to exchange information. In this way, a path towards the destination is

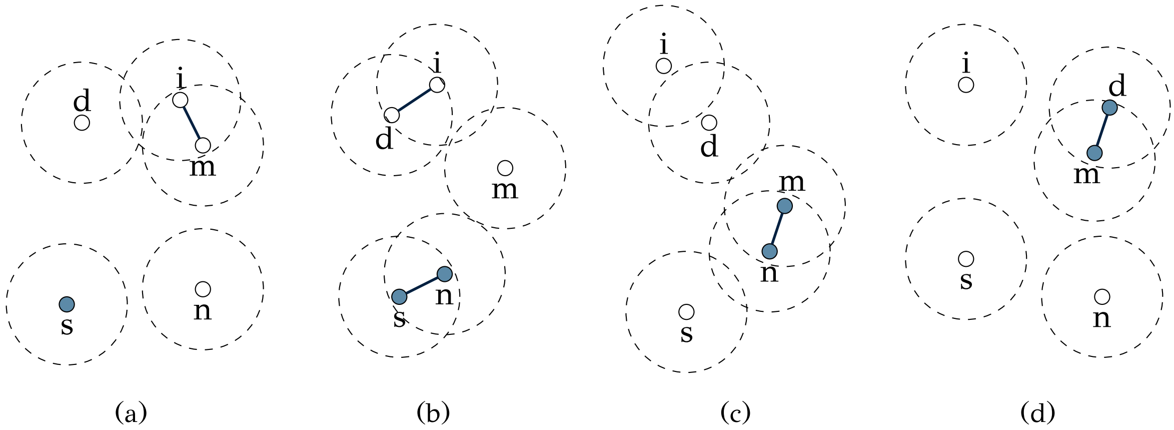


Figure 2.3: Snapshots of a small opportunistic network, consisting of five nodes, during routing a packet from node s to d at time instances: (a) t_1, (b) t_2, (c) t_3, and (d) t_4, where $t_1 < t_2 < t_3 < t_4$

formed over time. Fig. 2.3 depicts an example of how packet routing is performed using the store-carry-and-forward paradigm. We monitor a single packet generated at node s and destined to node d. The routing path is $s \rightarrow n \rightarrow m \rightarrow d$ and is formed gradually at time instances t_1, t_2, t_3 and t_4 during which the intermediate nodes come in contact. At each of these meetings, the packet is exchanged among the participating nodes (filled in blue circles, Fig. 2.3a-2.3d). Finally, at time t_4, the intermediate node m contacts the destination node d and successfully delivers the packet. Measurements on real-world network deployments [75–77] have shown that, due to node mobility, node contacts can be successfully exploited for routing packets. However, routing paths require a significant amount of time to form leading to increased delay for delivering data packets. For this reason opportunistic networks are also known as Delay Tolerant Networks (DTNs).

The main challenge in the design of routing algorithms for OppNets is finding the best path for each packet in a distributed fashion across time. The best path is the one that delivers the packet to its destination with the best possible delay using the minimal number of transmissions (lowest energy consumption). However, in most cases, a path with all the aforementioned characteristics does not exist. The fastest path is not the one with the minimal number of transmissions and vice versa. In this thesis, we focus on energy-efficient routing algorithms that target at striking the best possible balance between delay and energy cost (as expressed by the number of transmissions) without sacrificing the delivery efficiency. In the rest of this chapter, we present the current state of the art considering routing in opportunistic networks as the product of two main processes. The first process provides the means for identifying good relay candidates among network nodes (utility estimation). The second is responsible for exploiting the information provided by the aforementioned process in order to forward packets or disseminate packet replicas across the network in the most beneficial way (routing strategy).

2.2.2 Utility estimation

According to the store-carry-and-forward paradigm, each node u that holds a packet p destined to a node d in the network has to take the decision whether to forward p when it encounters another node v. Choosing an inappropriate relay node could lead to situations where the packet is never delivered or it is delivered with high delay and/or excessive energy cost (in terms of transmissions). As a result, in order to identify good relay candidates most routing algorithms use a utility estimation process. The key component of this process is the concept of the utility metric which is defined as:

Definition 2.1 (Utility metric). A locally estimated metric that captures the fitness (or quality) of a node for delivering and/or forwarding a packet.

Each network node computes its own utility metric locally using the history of previous contacts such as the frequency and duration of past encounters. Then, during a contact, the two participating nodes exchange their utility values before any packet forwarding takes place. In this way, each node is able to better evaluate the benefits of the forwarding opportunity and act accordingly. There exists a diverse range of metrics [32–34, 36, 78–85] that are constructed from a node's feature such as the frequency or the regularity of its contacts, its importance in a social context, etc. Each of them can be classified among one of the following categories.

Definition 2.2 (Destination dependent utility metric). A utility metric that captures the ability of a node to reach a specific destination.

Destination dependent metrics require each node to store multiple utility values, one for each network node that the current node is aware of. This category of metrics is very useful for nodes close, either socially or topologically, to the destination. However, they are not so effective when routing is performed at distant nodes which have never interacted with the destination. One commonly used metric of this category is the so called Last Time Seen (LTS) [32, 80] which uses the elapsed time since the last contact with the destination. The basic idea behind LTS is that the best relay candidates are the nodes which have seen the destination most recently. Another well known destination dependent metric is the one utilized by the PRoPHET algorithm [81, 82]. It is based on the delivery predictability that measures the probability of encountering a node. Moreover, it is enhanced with an aging mechanism and utilizes the transitive property. The social pressure metric (SPM) [83, 84] is another destination dependent metric that takes into account the social relationship among nodes. Essentially, it captures their friendship using the frequency, the longevity and the regularity of past contacts.

Definition 2.3 (Destination independent utility metric). A utility metric that captures the ability of a node to interact with other network nodes regardless of the actual destination.

When a destination independent metric is utilized, each node estimates a single utility value that captures its importance for the network. Most routing schemes exploit this category of utility metrics to gather packets at nodes in central points of the network where they remain stored waiting for an interaction with the respective destinations. However, this approach has two main disadvantages; it leads to resource over-utilization in the central nodes and fails to deliver packets to destinations that do not interact with central points of the network. A well known destination independent utility metric is ENC [34, 78] that uses the total number of past encounters (with all other network nodes) of a node. A more sophisticated metric is Betweenness Centrality [86] that measures to what extend the node lies on the shortest paths from all nodes to all other. The distributed version of this utility, i.e. Ego Betweenness [33, 79, 85], is calculated locally at each node using its local contact graph (ego network) formed by node's contact history.

Definition 2.4 (Hybrid utility metric). A utility metric constructed using a mix of both destination dependent and independent metrics.

Typical examples of routing schemes that utilize a hybrid metric are the SimBet [79] and SimBetTS [33] algorithms. SimBet uses a mix of Ego Betweenness (destination independent) and the Similarity metric (destination dependent). The latter measures the number of common neighbors between two nodes using the local contact graph (ego network). SimBetTS is an extension of SimBet introducing additional metrics that work as tie strength indicators [87], i.e., measure how strong or weak is the relationship among network nodes. All of these indicators are destination dependent. More specifically, the frequency metric is based on the number of encounters with the other network nodes, the intimacy/closeness metric uses the duration of encounters between network nodes and the recency metric is based on the amount of time passed since the last contact between two nodes.

2.2.3 Routing strategy

The second major component of the routing process is the routing strategy which is comprised of a set of forwarding rules that defines which packets should be exchanged when network nodes come in contact. The initial approaches [1, 31, 88] use simple forwarding rules, e.g., based only on information about which packets each node carries. On the other hand, the most efficient approaches [32–36, 78–84] take into consideration the forwarding capability of the meeting nodes as expressed by a utility metric (section 2.2.2). Fig. 2.4 illustrates a classification of the well established routing strategies proposed in the literature.

Routing strategies can be grouped into two main categories; single-copy [78–84,89] and multi-copy [31–36]. The first category follows the common routing approach in wired and wireless networks, i.e., forwarding of a single packet instance across the network. The latter creates and forwards multiple copies of a packet, known as packet replicas. The basic idea behind the second category is that spreading more replicas increases the probability that a node carrying a replica will meet, i.e., move into the communication range of, the destination. Both single-copy and multi-copy schemes come with advantages and disadvantages. On one hand stands single-copy approaches that are energy-efficient but perform poorly in terms of delivery rate and delay. On the other hand multi-copy approaches achieve high performance in terms of delivery rate and delay at the cost of more transmissions and increased storage requirements. For example, consider the two extremes of each category; Direct [88] and Epidemic [1]. The first never performs packet forwarding with every source node waiting to directly encounter the destination nodes in order to deliver its packets. Epidemic produces replicas in a greedy manner that exploits all contacts among

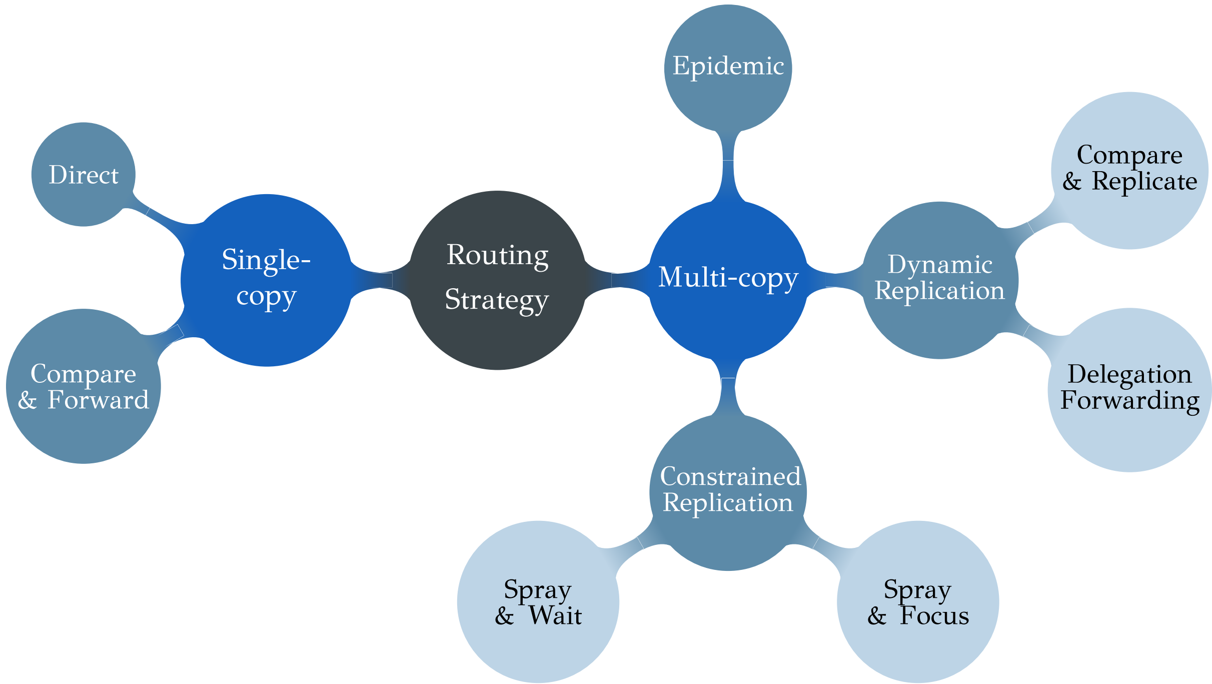


Figure 2.4: Classification of the well established routing strategies of the literature

nodes, essentially flooding the network. Clearly, the first strategy suffers poor delivery rate since the message is delivered only if the source node meets the destination. On the contrary, the second increases the delivery probability but is not suitable for a context of limited resources because it results in energy depletion and memory starvation at nodes.

Most routing strategies exploit a utility metric (section 2.2.2) to enhance their performance. In these cases, when nodes u and v come in contact they first exchange their corresponding utility values that are U_u and U_v, respectively. Then, for each packet p stored at node u the following criterion is applied.

Definition 2.5 (Relative criterion [32]). Packet p is forwarded (replicated) from u to v if and only if $U_v > U_u$.

After examining all packets that reside in u's buffer, the same process is performed for all packets stored at node v.

Compare & Forward is a single-copy strategy that takes forwarding decisions based on the relative criterion and is extensively utilized by a wide range of routing protocols [78–84,89]. This approach drastically increases the delivery rate and reduces the end-to-end delay compared to the Direct strategy. However, the performance gains are still limited. The main reason is because this approach heavily depends on the quality of the utility metric used. Even the most efficient utility metrics do not provide 100% accuracy leading to incorrect forwarding decisions that degrade the routing performance. Also, most of the metrics lack consistency in the sense that a node's utility value fluctuates over time. This inconsistency produces routing circles where packets are exchanged among the same set of nodes inducing increased transmissions and poor delay. Furthermore, using only one packet instance acts as a single point of failure substantially reducing the delivery efficiency. For example, intermediates nodes may require to drop packets due to storage limitations increasing the delivery failures.

To overcome the limitations of single-copy schemes the research community focused on multi-copy strategies that allow multiple packet replicas being forwarded simultaneously. In these schemes, each replica uses a different path increasing the delivery probability since only one of them is required to reach the destination node. However, failing to control the level of replication can lead to excessive number of transmissions and resource over-utilization. A well known approach is to use constrained replication which sets beforehand an upper bound for the number of packet replicas that can exist throughout the network [31–35]. This is accomplished by assigning a predefined value L to each generated packet that indicates how many times it can be replicated. During the replication phase, also known as spraying phase, the L value is split between the new packet replica and the old. More specifically L can be equally distributed among the two replicas [31, 32, 35] or it can be unevenly divided according to the utility values of the corresponding nodes in contact [33, 34]. When the spraying phase terminates, i.e., the L value decreases to one, a single-copy strategy is engaged. Each replica can be either buffered to the current node waiting to be directly delivered to its destination [31, 32] (Spray & Wait) or further forwarded, but not replicated, using the relative criterion [33–35] (Spray & Focus). Overall, the performance of all routing schemes that utilize constrained replication is severely affected by the replication's upper bound (L value) that provides a trade-off between cost and efficiency. Predefining this upper bound is hard since it strongly depends on dynamically changing factors, such as network state and network characteristics.

An alternative approach to control replication operates in a more dynamic fashion by exploiting the utility values of the network nodes. An initial scheme that uses dynamic replication is Compare & Replicate which utilizes the relative criterion in the same way as Compare & Forward. However, during contact opportunities, Compare & Replicate transmits packet replicas instead of original packets. Although Compare & Replicate significantly reduces the routing cost in comparison to Epidemic, it still produces a large number of transmissions. A more efficient strategy is Delegation Forwarding [36] which takes into account the history of a node's observations in order to regulate packet replication. According to this scheme, each node is required to keep track of the highest utility value seen among its previous contacts. Then, during a contact between nodes u and v the following criterion is applied.

Definition 2.6 (Delegation criterion [36]). Packet p is replicated from u to v if and only if $U_v > \max_{k \in N_u}\{U_k\}$, where N_u is the set of all nodes that u has contacted since the reception of p.

The rationale is that there is a little benefit in replicating a message to a node with a utility lower than the highest recorded utility. In this way, Delegation Forwarding succeeds a significant reduction in the energy cost, without impacting the delivery efficiency.

In chapter 5, we build on the premises of Delegation forwarding aiming to further minimize the energy cost through the reduction of the number of replications. To this end, we take advantage of the cooperation between nodes to coordinate their views about the highest utility value seen in the network used in the replication process. Furthermore, we allow nodes which do not carry a packet replica to play an active role in the replication process and deny receiving replicas for which they were rejected in previous contacts.

2.3 Summary

Spreading multiple packet instances throughout the network is a common approach followed by a diverse range of networking algorithms in wireless ad hoc networks. Although this approach is energy-consuming, it is the only way for achieving an acceptable performance when broadcasting in mobile ad hoc networks (MANETs) and routing in opportunistic networks (OppNets). In both fields, we presented an overview of the proposed schemes in the literature focusing on the energy-efficient ones in the sense that they target to minimize the energy consumption through the reduction of the number of transmissions without affecting the overall performance.

Chapter 3

XOR-based Broadcasting with Coded Redundancy

Network coding is commonly used to improve the energy efficiency of network-wide broadcasting in wireless multi-hop networks. In this chapter, we focus on XOR-based broadcasting in mobile ad hoc networks with multiple sources. Initially, we demonstrate through extensive experimentation that the state-of-the-art XOR-based approach suffers performance breakdowns. Motivated by this observation, we examine in depth the synergy of network coding and the underlying broadcast algorithm exposing the causes that lead to performance degradation. Then, we introduce a novel XOR-based broadcast algorithm that efficiently applies XOR coding over CDS-based broadcasting increasing the broadcasting performance in terms of energy cost, delivery delay and utilization of network resources.

3.1 Preliminaries

We first provide a review of the basic principles of CDS-based broadcasting as well as XOR-based coding.

3.1.1 CDS-based broadcast principles

Energy efficient broadcast algorithms aim to minimize the number of transmissions required for delivering a packet to all network nodes [9, 10]. The most effective algorithms follow the CDS-based broadcasting approach. According to this, the algorithm constructs a connected dominating set of the network [24–26]. The nodes constituting the CDS are the *forwarders*, i.e., those elected to forward the broadcast packets, while all other nodes just act as passive receivers. Since computing the forwarders should be performed in a distributed fashion, the common approach is to approximate them locally at each node v using its 1-hop neighbor set ($\mathcal{N}(v)$), i.e., the set that consists of v's one hop neighbors, and the 2-hop neighbor set ($\mathcal{N}(\mathcal{N}(v))$), i.e., the set consisting of all nodes that lie at maximum two hops away from v.

Even though transmitting packets only through forwarders successfully reduces packet duplicates, a significant number of them still exists across the network. This is because the selection of forwarders is made in a distributed manner and with limited information. As a result, special attention should be given to these duplicates as they could lead to additional transmissions and degrade energy efficiency. Therefore, the reception of a packet duplicate in a forwarder node leads to a dilemma whether to forward it or not. Forwarding the duplicate could increase redundant transmissions while dropping it could potentially impact the delivery efficiency. The mechanism that is responsible to handle such situations is the *termination criterion*. Multiple criteria have been proposed in the literature [27, 50, 90, 91]. In the rest of this chapter we will use the terminology proposed in [27] and [90] to refer to these criteria:

Termination Criterion 1 [Marked/unmarked (M/U)]: *Each node keeps track of the packets received by each of its 1-hop neighbors. Then, in the case of a duplicate reception, a forwarder transmits the received duplicate if at least one of the neighbors is not marked to have received the packet.*

Termination Criterion 2 [Relayed/unrelayed (R/U)]: *A forwarder transmits a duplicate only if no other duplicate of the same packet has been relayed by the same forwarder in the* *past.*

Termination Criterion 3 [Covered/uncovered (C/U)]: *A node acting as a forwarder relays packets seen for the first time while it drops already seen packets including the ones not relayed in the past because the node was not elected as a forwarder at that time.*

The M/U is the most well-known approach. However, having all nodes to store the reception status for all of their 1-hop neighbors and for all packets could be a daunting challenge in terms of both memory usage and processing overhead [27]. In contrast to M/U, the latter two approaches are more realistic due to the limited storage and processing requirements.

The algorithms proposed in the literature follow two major strategies for building the CDS, i.e., calculating the forwarders. The first is to build a CDS that is common to every network node using local information [12, 25, 38–40, 45–52] while the second is to build a source-specific CDS [26, 27, 53, 54]. In the first category the nodes of the CDS are used for any packet regardless of its source and updated whenever topology changes are detected. Most efficient studies in this line of research also use information related to the broadcast process, e.g., packet reception status, in order to further prune transmissions and/or enhance reliability [25,45–52]. On the other hand, in the second category, a node that relays a packet calculates the list of forwarders by considering the previous hop of the packet and piggybacks the corresponding list on it. In this way, a source-based CDS is formed for each packet. More specifically, when a node v receives a packet from u checks whether it is selected as a forwarder. If so, a common approach is to elect forwarders so as to deliver the packet to (or "cover") the set $\mathcal{U}(v)$ of nodes that lie exactly 2-hops away from v, i.e., $\mathcal{U}(v)=\mathcal{N}(\mathcal{N}(v))-\mathcal{N}(v)$. The set of candidate forwarders $\mathcal{C}(v)$ is in general a subset of v's neighbors, i.e., $\mathcal{C}(v)\subseteq\mathcal{N}(v)$. Note that $\mathcal{U}(v)\subseteq \bigcup_{\forall u\in\mathcal{C}(v)} \mathcal{N}(u)$ and that $\mathcal{C}(v)$ can be seen as a set of sets if each node $u \in \mathcal{C}(v)$ is replaced by $\mathcal{N}(u)$, thus the election of forwarders is modeled as a set cover problem. The solution is usually given by the well-known greedy set cover (GSC) algorithm [92], however other more efficient approximation algorithms exist [25,93–95]. Furthermore, node v takes advantage of u's neighborhood to reduce both the set of candidate forwarders, i.e., $\mathcal{C}(v)=\mathcal{N}(v)-\mathcal{N}(u)$, and the set of nodes $\mathcal{U}(v)$ that should receive the packet. Algorithms in the sourced-based CDS category vary in the approach taken to minimize the set $\mathcal{U}(v)$ and therefore the number of forwarders. TDP and PDP [27] exploit u's two-hop neighborhood and further minimize the $\mathcal{U}(v)$

set. For example, node v in PDP elects forwarders in order to cover the nodes in $\mathcal{U}(v) = \mathcal{N}(\mathcal{N}(v)) – \mathcal{N}(v) – \mathcal{N}(u) – \mathcal{N}(\mathcal{N}(u) \cap \mathcal{N}(v))$. For presentation purposes we selected Partial Dominant Pruning (PDP) [27] as the reference algorithm. However, our findings can be easily generalized to all CDS-based broadcast protocols. The reason is that, as far as the CDS-based operation is concerned, we focus on the termination criterion which is a generic mechanism that does not depend on the specifics of each algorithm.

3.1.2 XOR coding specifics

XOR-based coding works on a hop-by-hop basis, i.e., packets encoded by a node are decoded by its neighbors. The idea is that each node v can combine packets using bitwise XOR operations in order to produce an encoded packet. For the neighboring nodes to be able to decode the encoded packet, the choice of native, i.e., non coded, packets is important. More specifically, for a successful coding of k packets , each neighbor should know $k-1$ of those packets beforehand. This requirement guarantees that each neighbor should be able to decode the encoded packet. The existence of $k > 1$ packets that can be encoded is known as a *coding opportunity* [20]. It is clear that, finding a coding opportunity depends on v's knowledge about the packets that each of its neighbors has already received. To acquire such information, v employs opportunistic listening [20,21] and snoops all communication in the wireless medium. The acquired information is stored in what is called the *neighbor reception table*. Moreover, node v should store in what is called the *packet pool* all recently received native packets in order to be able to perform decoding of encoded packets. To describe the method more formally, let $\mathcal{P}_v$ denote v's packet pool, i.e., the set of native packets recently received by v and $\mathcal{R}_v^u$ denote u's view of the same buffer. Note that $\mathcal{R}_v^u$ is part of u's neighbor reception table. Node v may choose a set of native packets $\mathcal{B}' \subseteq \mathcal{P}_v$ and produce an encoded packet, by using bitwise XOR, in the presence of a coding opportunity. This means that a set $\mathcal{B}' \neq \emptyset, |\mathcal{B}'| > 1$ can be found such that, according to v's neighbor reception table, each node $u \in \mathcal{N}(v)$ has received at least $|\mathcal{B}'|–1$ of the native packets in $\mathcal{B}'$, i.e., $|\mathcal{R}_u^v \cap \mathcal{B}'| \geq |\mathcal{B}'| - 1, \forall u \in \mathcal{N}(v)$. Successful decoding depends on the consistency of $\mathcal{R}_u^v$, i.e., whether $\mathcal{R}_u^v \subseteq \mathcal{P}_u$. Decoding failures at a node u occur when $|\mathcal{P}_u \cap \mathcal{B}'| < |\mathcal{B}'| - 1$ and result in the loss of all packets included in the encoded one.

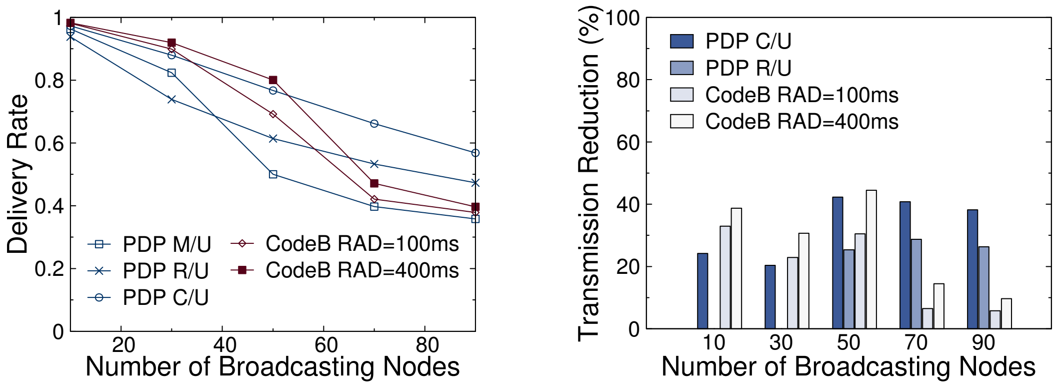
The efficiency of XOR-based coding clearly depends on the existence of coding opportunities. This is because for each encoded packet that contains k native ones only one transmission is required instead of k, thus saving energy and reducing packet collisions. To maximize the number of coding opportunities XOR-based coding approaches introduce a *Random Assessment Delay (RAD)* before relaying a packet. Higher values of RAD result in more candidate packets for encoding, however this comes at the cost of increased end-to-end delay.

3.2 The synergy between network coding and the termination criterion

XOR network coding as well as the termination criterion of a CDS-based algorithm are essential mechanisms for energy efficient broadcasting as both aim to minimize packet transmissions. Ensuring a smooth synergy is critical for building an efficient algorithm. Most proposed coding-based broadcast algorithms combine XOR-coding with CDS-based approaches that utilize the M/U criterion [21, 55–57] while others do not provide insight on the termination criterion used [57, 58]. In the following we show that M/U faces performance issues that hamper the coding operation. At the same time, using other proposed termination criteria, such as R/U and C/U, in parallel with XOR coding raises significant design issues.

3.2.1 The M/U criterion limits coding gains

The choice to combine M/U with XOR coding is reasonable. First, M/U is compatible with the RAD technique that is essential for network coding. In fact, RAD improves the pruning efficiency of M/U. This is because the imposed delay allows the reception of more packet duplicates which could potentially change the initial decision to relay the packet. This is the reason for which RAD has also been proposed in the context of broadcasting without network coding [9, 10]. However, although in non-coded approaches there are alternatives that provide performance improvements similar to that of RAD but without the associated delay, in coding-based approaches using RAD is essential. The second reason for which state-of-the-art XOR-based approaches adopt the M/U criterion is because in this case XOR coding can be implemented with limited cost. Recall that the latter requires information about the reception status of the



(a) (b)

Figure 3.1: Performance under varying traffic load: (a) Delivery rate (b) Transmission reduction compared to PDP M/U

neighboring nodes, i.e., the neighbor reception table. This is exactly the information on which M/U decisions are based on, therefore this information is already available through the implementation of M/U. Besides the advantages of using M/U there is also a significant downside. M/U is known to be less efficient than other proposed criteria [90, 91]. This motivated us to further examine the performance of XOR-based broadcasting implementing M/U against non coding schemes utilizing the other termination criteria, i.e., R/U and C/U.

For our investigation, we conducted a series of experiments using the ns2 simulator [96]. We chose to experiment with the well-established CodeB algorithm [21] that utilizes network coding and builds on top of PDP using the M/U termination criterion. In our experimental setup, 100 nodes move with maximum speed of 1 m/sec in a square area according to the Random Waypoint (RW) model [97]. Each node has a neighborhood with an average size of 15 nodes, while 50 nodes generate broadcast traffic with a rate of 1 packet/sec. More information about the simulation set-up can be found in Section 3.5.

First, we evaluated the performance of CodeB under various levels of offered traffic by varying the number of source nodes. Fig. 3.1a depicts its delivery efficiency, while Fig. 3.1b displays its ability to prune transmissions compared to the non-coding PDP scheme that uses the M/U criterion. As expected, CodeB successfully reduces the number of transmissions and provides enhanced packet delivery in cases of low to medium traffic load. However, as the network traffic increases its efficiency deteriorates. Based on this observation, we attempted to boost CodeB's performance.

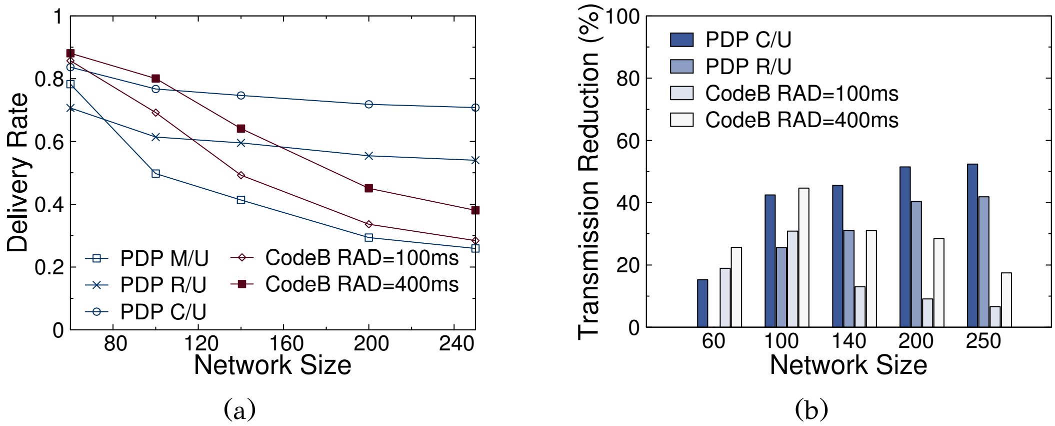


Figure 3.2: Performance for different network sizes: (a) Delivery rate (b) Transmission reduction compared to PDP M/U

More specifically, we included a new version of CodeB that maximizes the benefits of network coding by increasing the RAD value from 200 ms to 400 ms. However, despite the fact that the new CodeB version clearly discovers more coding opportunities and thus further reduces transmissions (Fig. 3.1b), its improvement in terms of delivery efficiency is limited (Fig. 3.1a). To further investigate the reasons behind CodeB's poor behavior, we compared it against two versions of the non-coding PDP scheme, one implementing the R/U criterion and the other the C/U. If there exists at least one simple PDP scheme that performs better than CodeB then the origins of the witnessed poor behavior reside in the termination criterion rather than the coding mechanism itself. Interestingly, this is confirmed by our results in Fig. 3.1. After the breaking point of 50 broadcast sources (half of the network nodes), the non-coding PDP schemes outperform CodeB regardless of the RAD value used. The only exception is PDP M/U that, similar to CodeB, suffers from a performance breakdown.

Similar findings are witnessed in our second experiment where we assess the scalability of all algorithms by increasing the number of network participants (Fig. 3.2). CodeB outperforms all schemes for networks comprised of fewer than 100 participants. Despite the fact that the offered load remains constant as the network size increases, CodeB and PDP M/U cannot avoid performance breakdown (Fig. 3.2a). Both generate a large number of transmissions (Fig. 3.2b) that induce failures due to packet collisions. Increasing the RAD value offers CodeB a performance improvement, however the gain is still limited and the problem is not solved. On the other hand, the non-coding schemes, PDP R/U and C/U, present a relatively stable behavior regardless of the network size. Clearly, the best performing algorithm is PDP C/U that reduces transmissions by up to $\sim$60% while keeping the delivery efficiency above $\sim$75%.

Overall, the results revealed that the combination of network coding with the M/U termination criterion is not always the best choice. In particular, non-coding schemes perform far better than CodeB when the traffic in the network increases; either because more traffic is offered from more sources (first experiment) or because in a bigger network (second experiment) more forwarders exist and produce more packet duplicates. As explained, this behavior is not the result of the coding operation itself but is inherited from the underlying broadcast scheme and more specifically the termination criterion. There are two reasons for this. The first and predominant one is the limited ability of M/U to prune redundant transmissions. As a result, congestion quickly builds up and results in more collisions, therefore reducing delivery efficiency. The second reason is related to the neighbor reception table, i.e., the structure containing information about the packets received by each neighbor, which is necessary for both M/U and XOR coding. In the typical implementation of this structure, information for each packet is maintained for a limited time period. As traffic in the network increases, the delay jitter between the first and the last duplicate of a packet also increases. As a result, there is an increased probability that a packet duplicate arrives at a node v after the information for that packet has expired and been removed from the neighbor reception table. This results in node v transmitting more duplicates and thus aggravating congestion. Increasing the expiration period for information in the neighbor reception table improves performance up to a limit. After that, no further improvement is possible and performance breakdown is still evident due to the limited pruning ability of M/U. We also implemented the neighbor reception table as a fixed size structure without imposing an expiration period. We tested different sizes and found that performance degradation appears to be more severe in this case.

3.2.2 The pitfalls of using other termination criteria

Our observations highlight the need for replacing M/U with alternative criteria, such as R/U and C/U. However, doing so is not straightforward. The main reason is what we call "the packet reordering problem". Before analyzing the packet reordering problem,

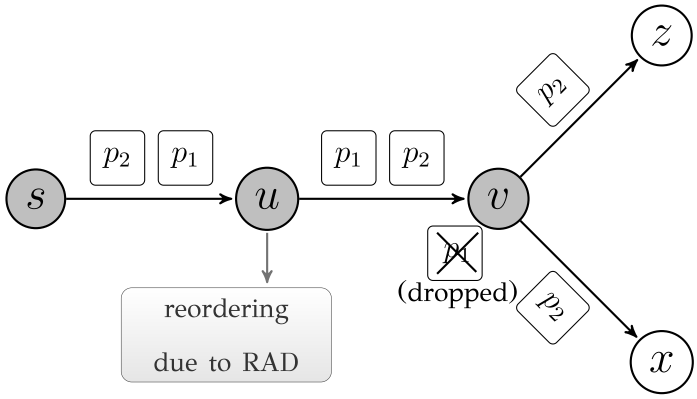
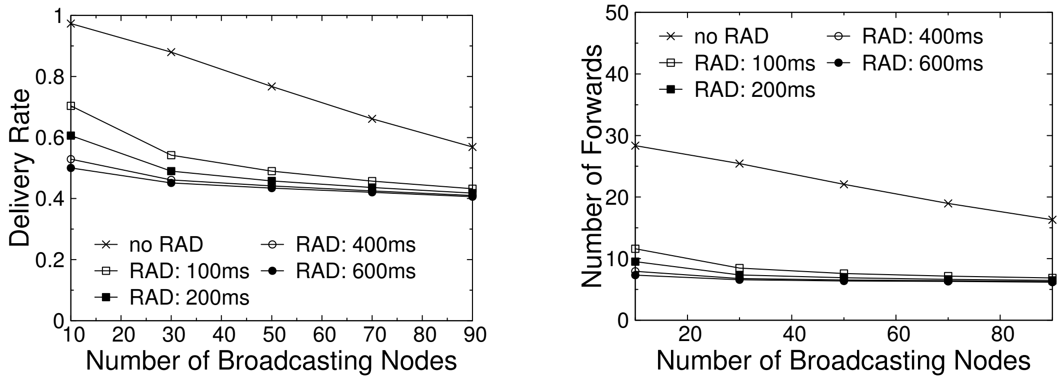


Figure 3.3: Example where the propagation of packet p_1 terminates due to the packet reordering problem

let us first describe the implementation aspects of the two alternative termination criteria. Recall that both R/U and C/U delineate a policy for handling packet duplicates. More specifically, in R/U a forwarder relays a duplicate only if no other duplicate of the same packet was relayed in the past. On the other hand, in C/U a forwarder v relays only the packets seen for the first time and ignores packets seen in the past even if v did not forward those packets, i.e., v was not selected as a forwarder at that time. In order for both R/U and C/U to function properly, there are two prerequisites. The first is that packets should be uniquely identified through a number added by the source node at creation time. The second prerequisite is that each node implementing the termination criterion should store a full reception history on a packet basis (i.e., the id's of received packets). This is neither practical nor realistic due to the high storage and processing requirements. For this reason, the traditional implementation of both R/U and C/U takes a much simpler approach. The numbers used to identify packets are assigned by the source in a sequential manner (thus called sequence numbers) so as packets with higher numbers correspond to the ones created more recently. This allows each node v that implements either R/U or C/U to only store a single sequence number ($\mathsf{SN_s}$) for every source node s. In the R/U criterion (C/U criterion), this is the largest number seen in a packet from s and forwarded (received) by v. Then, for an incoming packet p_1 carrying the sequence number $\mathsf{SN_{p_1}}$ it is sufficient to check that $\mathsf{SN_{p_1}} > \mathsf{SN_s}$ so as to decide that it is not a duplicate.

Unfortunately, the aforementioned implementation is fully functional only under the assumption that all nodes in the network receive packets in the same order in which they were created. When this order is altered the problem that we call *packet*



(a) (b)

Figure 3.4: Performance of PDP C/U vs traffic load using different RAD intervals: (a) Delivery efficiency (b) Average number of forwards per packet

reordering emerges, impairing the ability of both R/U and C/U to detect duplicates and therefore having a severe impact on their performance. The utilization of XOR coding unfortunately results in packet reordering and thus its incompatibility with current implementations of both R/U and C/U. More specifically, packet reordering appears due to the random assessment delay (RAD) that network coding uses at each node in order to maximize the probability of finding a coding opportunity. To make it more clear, let us examine the problem through an example in which both R/U and C/U fail to work properly. Fig. 3.3 illustrates the propagation of packets p_1 and p_2 across an example network. Both packets originate from the same source s and p_1 is created before p_2. Therefore, the sequence number of p_1 is smaller than that of p_2, i.e., $\mathsf{SN}_{p_1} < \mathsf{SN}_{p_2}$. When RAD is not utilized, u will forward both packets in the same order as received. Then, v will first receive p_1, update SN_s, i.e., $\mathsf{SN}_s \leftarrow \mathsf{SN}_{p_1}$, and finally forward p_1. Upon reception of p_2, v will confirm that $\mathsf{SN}_{p_2} > \mathsf{SN}_s$ and will forward p_2. On the other hand, if RAD is utilized, u introduces a random delay before forwarding p_1 and p_2. Due to randomness, the delay for p_2 may be significantly smaller than the corresponding delay for p_1, thus resulting in u forwarding the two packets in the reverse order, i.e., p_2 first and then p_1. After receiving p_2, v updates SN_s to the value SN_{p_2} and forwards p_2. Later on, when v receives p_1 makes the observation that $\mathsf{SN}_{p_1} < \mathsf{SN}_s$, therefore rejects p_1 although it is not a duplicate. This decision has a major impact on the delivery efficiency since p_1 never reaches nodes z and x.

To validate the impact of the packet reordering problem we conducted a series of experiments on the PDP algorithm using the C/U termination criterion. We examined

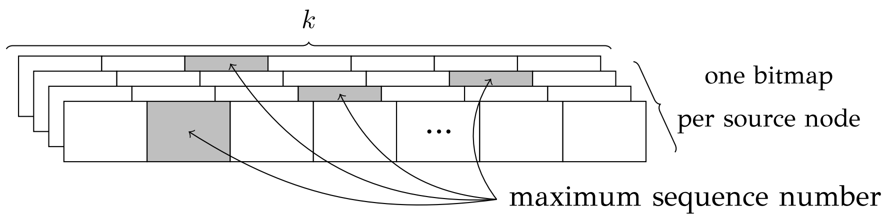


Figure 3.5: Structure used for the modified covered/uncovered (MC/U) termination criterion

the effect of different values of the random assessment delay. For the experimental setup we used the same settings described in Section 3.2.1. Fig. 3.4 illustrates our main results. Clearly, RAD has a considerable impact on the overall broadcasting performance regardless of the offered load. More specifically, the packet reordering problem may result in a reduction of the number of relaying decisions up to $\sim$3 times (Fig. 3.4b). However, this pruning is erroneous in the sense that it prematurely terminates the broadcasting process, thus, reducing the delivery efficiency up to $\sim$50% (Fig. 3.4a). As expected, higher RAD values have a more severe impact on the performance as in these cases the probability of receiving packets out of order increases. We observed similar findings in the R/U case.

3.3 Building a coding friendly termination criterion

Establishing the compatibility of R/U and C/U with XOR coding requires solving the packet reordering problem. Allowing each node that implements either of these criteria to store a full packet reception history can provide a solution. However, as mentioned earlier, this approach is neither practical nor realistic due to the high storage and processing requirements.

Towards a more efficient solution, we propose the *modified covered/uncovered (MC/U)* termination criterion that extends C/U. We choose to build on top of C/U because both the related literature [90, 91] and our experimental results (Fig. 3.1 and Fig. 3.2) confirm that it achieves the best performance against all other proposed termination criteria. The main idea behind our approach is to implement the same forwarding criteria as in C/U but to allow each node to store information (just one bit as we will discuss in the following) for each of the k, instead of just one, most recently

Algorithm 3.1 Pseudocode of the MC/U forwarding procedure.

```
RelayOrNot(packet p, bitmap BM_s, int SN_s^MAX, int mindex)
if (p.SN > SN_s^MAX) then
    Update(p,BM_s,SN_s^MAX,mindex)
    relay p if forwarder
else
    SN_s^MIN ← SN_s^MAX − k
    if (p.SN <= SN_s^MIN) then
        drop p
    else
        index ← p.SN − SN_s^MAX + mindex
        if (index < 0) then
            index ← index + k
        end if
        val ← BM_s.get(index)
        if (val) then
            drop p
        else
            BM_s.set(index)
            relay p if forwarder
        end if
    end if
end if
```

seen packets from each source node. This allows the node to detect any duplicate of these k packets without problems caused by packet reordering. Duplicate detection is not possible for a packet that is older than the k recorded ones because no relevant information is available. However, this is important only if a copy of a packet p from source s is received by a node after the k-th packet that s generated after p. By increasing k it is possible to minimize the probability of such an occasion. Even if such an occasion arises we choose to drop the packet, i.e., adopt the C/U policy, rather than forwarding it (which corresponds to the M/U policy) in order to avoid increasing the network congestion levels.

Selecting a proper value for k is clearly a challenging task. Large values increase

Algorithm 3.2 Pseudocode for updating the bitmap.

Update(packet p, **bitmap** $\mathtt{BM_s}$, **int** $\mathtt{SN_s^{MAX}}$, **int** $\mathtt{mindex}$**)**

$\mathtt{mindex}' \leftarrow (\mathtt{mindex} + p.\mathtt{SN} - \mathtt{SN_s^{MAX}}) \% k$

$\mathtt{rollover} \leftarrow \left\lfloor \frac{\mathtt{mindex} + p.\mathtt{SN} - \mathtt{SN_s^{MAX}}}{k} \right\rfloor$

if $(\mathtt{rollover} == 1)$ **then**

  $\mathtt{BM_s}.zero(\mathtt{mindex} + 1, k - 1)$

  $\mathtt{BM_s}.zero(0, \mathtt{mindex}' - 1)$

else if $(\mathtt{rollover} > 1)$ **then**

  $\mathtt{BM_s}.zero(0, k - 1)$

else

  $\mathtt{BM_s}.zero(\mathtt{mindex} + 1, \mathtt{mindex}' - 1)$

end if

$\mathtt{mindex} \leftarrow \mathtt{mindex}'$

$\mathtt{SN_s^{MAX}} \leftarrow p.\mathtt{SN}$

$\mathtt{BM_s}.set(\mathtt{mindex})$

the storage and processing requirements at each node while small values increase the probability of receiving a packet without being able to decide whether it is a duplicate or not. After experimentation, we concluded that the MC/U criterion has a competitive performance even when a small value of k is required due to space limitations. Nonetheless, the storage and processing requirements may raise a concern. To address such concerns, we implement MC/U using bitmaps. Note that bitmaps have been used for similar purposes in the context of multicasting in ad hoc networks [98]. More specifically, a node v that implements MC/U, instead of storing the k last seen sequence numbers from a source s, it uses only one bit for each of them, i.e., a total of k bits in the form of a bitmap $\mathtt{BM_s}$ (Fig. 3.5). Then sequence numbers are mapped to the bits of $\mathtt{BM_s}$ and each bit is used to indicate whether the corresponding sequence number is known (bit set to 1), i.e., v has already received a packet carrying this sequence number, or not (bit set to 0). At the same time, by using bitmaps the node takes advantage of the low cost read/write operations. Furthermore, node v stores the maximum known sequence number from s ($\mathtt{SN_s^{MAX}}$) as well as the index ($\mathtt{mindex}$) of the bit in $\mathtt{BM_s}$ that corresponds to $\mathtt{SN_s^{MAX}}$.

The functionality of MC/U is illustrated in algorithm 3.1. When a node v receives a packet p from s it first checks whether its sequence number $p.\mathtt{SN}$ is greater than

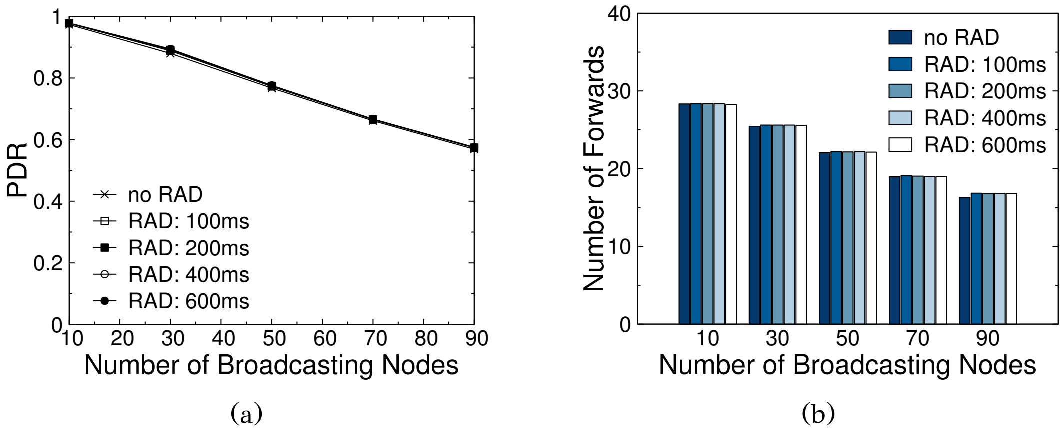


Figure 3.6: Performance of PDP MC/U under varying traffic load using different RAD intervals: (a) Delivery efficiency (b) Average end-to-end delay

$\mathsf{SN}_s^{\mathsf{MAX}}$. If this is the case v relays the packet (if it is an elected forwarder) and updates its state (algorithm 3.2). This update involves the following steps. First v calculates the index ($\mathtt{mindex}'$) of the bit that corresponds to the new sequence number (line 1, algorithm 3.2). Observe that this calculation may involve a rollover, i.e., reusing the bits of the bitmap. Then v resets all the bits from position $\mathtt{mindex}+1$ to $\mathtt{mindex}'-1$ (lines 2-10, algorithm 3.2). This is done because those bits correspond to the sequence numbers between $\mathsf{SN}_s^{\mathsf{MAX}}$ and $p.\mathsf{SN}$ and no packet carrying one of these numbers has been received so far. Note that if $p.\mathsf{SN} - \mathsf{SN}_s^{\mathsf{MAX}} > k$, i.e., a multiple rollover occurs, then all bits of the bitmap must be reset (lines 6-7, algorithm 3.2). Finally, v updates $\mathsf{SN}_s^{\mathsf{MAX}}$ and $\mathtt{mindex}$ (lines 11-12, algorithm 3.2) and sets the corresponding bit to indicate that a packet carrying $\mathsf{SN}_s^{\mathsf{MAX}}$ has already been received (line 13, algorithm 3.2). Going back to the basic algorithm, if $p.\mathsf{SN} \leq \mathsf{SN}_s^{\mathsf{MAX}}$ then v should decide whether $p.\mathsf{SN}$ is one of the k most recent sequence numbers. If not (lines 5-7, algorithm 3.1) the packet is dropped because it is not possible to decide whether it is a duplicate or not. Otherwise, v calculates the index of the bit that corresponds to $p.\mathsf{SN}$ (lines 9-12, algorithm 3.1). If that bit is set to 1 then p is dropped because it is a duplicate otherwise the bit is set to 1 and p is relayed if v is an elected forwarder (lines 13-18, algorithm 3.1).

To validate the efficacy of MC/U we developed a version of the PDP scheme that utilizes it instead of C/U. Then, we repeated the same experiments described in Section 3.2.2, testing different values of RAD under varying offered load. Fig. 3.6 illustrates our main results. In contrast to PDP C/U (Fig. 3.4a), RAD has a negligible impact on the delivery performance of PDP MC/U (Fig. 3.6a). At the same time, the ability of MC/U to prune transmissions is not damaged. Both MC/U and C/U achieve roughly the same number of transmissions (compare Fig. 3.6b and 3.4b when the performance of C/U does not collapse, i.e., when no RAD is used). Overall, the results prove that MC/U successfully tackles the packet reordering problem.

3.4 Network coding broadcast with coded redundancy

In this section, we introduce the *Network cOding Broadcast with Coded Redundancy* (NOB-CR) algorithm. NOB-CR, similar to other schemes, takes the approach to implement XOR coding on top of a CDS-based broadcast algorithm. However, in order to maximize the performance of network coding, NOB-CR employs the MC/U termination criterion introduced in section 3.3. As the default CDS algorithm we choose PDP although any algorithm of this category could be used. Regarding the coding process, similar to XOR coding approaches, NOB-CR utilizes network coding on a hop-by-hop basis. Each intermediate node uses bitwise XOR operations to combine native packets into encoded ones under the requirement that all neighboring nodes can decode them. Nonetheless, NOB-CR uses only one of the two specialized data structures required for coding (see Section 3.1.2), i.e., the packet pool. This is because its lightweight coding detection mechanism renders obsolete the use of the other one, i.e., the neighbor reception table. We discuss this issue in detail in Section 3.4.2. Besides the aforementioned differences, NOB-CR deviates from other approaches by introducing a series of mechanisms that significantly improve the broadcasting performance and alleviate the related costs. In particular, these mechanisms are:

- A lightweight coding detection method that operates without the need of maintaining a neighbor reception table.
- A novel method for the computation of forwarders that uses information provided by the RAD mechanism to increase the overall pruning efficiency.
- A cost-free method to inject packet redundancy in the network in order to reduce the end-to-end delay.

In the following, we delineate NOB-CR's basic operation as well as the aforementioned mechanisms.

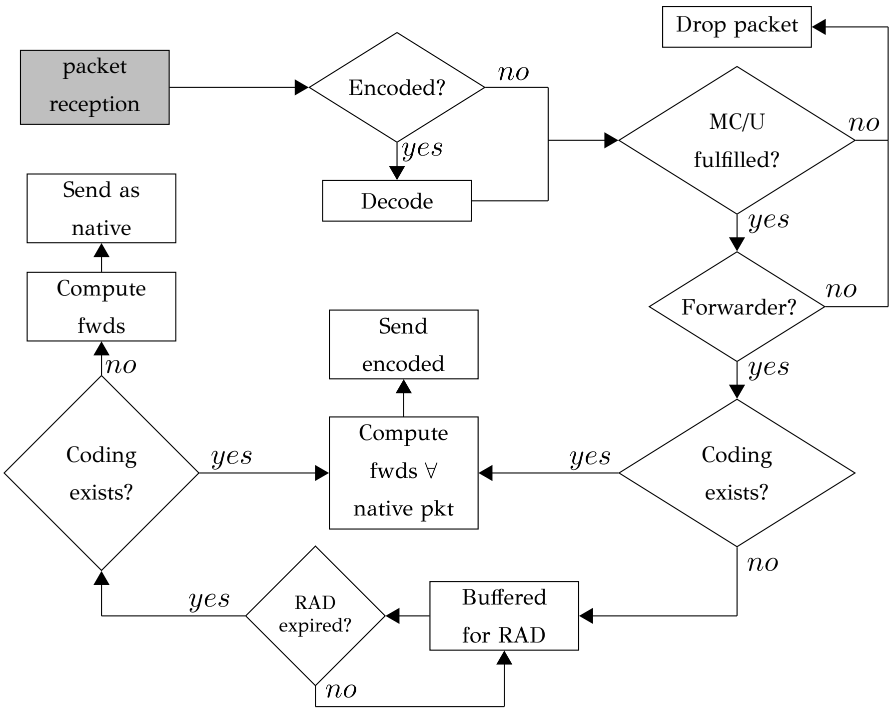


Figure 3.7: Flow diagram of a packet's life cycle in NOB-CR.

3.4.1 Basic operation

Fig. 3.7 describes the basic operation of NOB-CR by illustrating the life cycle of a packet at a node that implements the protocol. An incoming packet is first examined for deciding if it is a native or an encoded one. In the latter case, the packet is decoded to produce the native packets that it consists of. Then, for each native packet the receiving node examines: a) whether the packet meets the termination criterion conditions, and b) the set of forwarders that is piggybacked on the packet to determine if it is selected as a forwarder for the packet. If at least one of these tests is negative the packet is dropped and the process terminates. Otherwise, the coding opportunity detection process initiates. If a coding opportunity is detected, the set of forwarders is determined (please refer to Section 3.1.2 for details) for each native packet involved and then the encoded packet is created and immediately transmitted. In the absence of a coding opportunity, the received packet is temporarily buffered in the output queue for a randomly chosen time interval according to the RAD mechanism. This allows the packet to participate in subsequent coding inquiries. When the buffering interval expires the packet is examined for coding one last time before being transmitted as native.

Finding coding opportunities at an intermediate node strongly depends on the

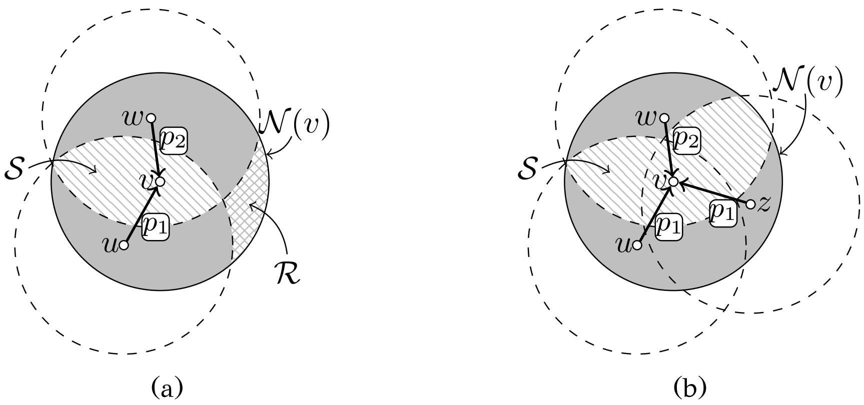


Figure 3.8: Searching for coding opportunities when v has received (a) one duplicate, or (b) more duplicates per packet.

packets that each neighbor has already received. To acquire such information, each node snoops all communication in the wireless medium [20, 21] and stores it in the neighbor reception table. However, the maintenance of this table comes at a significant cost. Keeping track of every packet received by each neighbor requires a considerable amount of storage. Likewise, updating the neighbor reception table on a packet arrival basis is a task that requires significant processing power. In order to avoid these costs, we introduce a new approach for finding coding opportunities. Our method operates without the need for a neighbor reception table. In fact, it uses neighborhood information that is available through the underlying broadcasting mechanism.

3.4.2 Lightweight detection of coding opportunities

To explain the method, let us use the example in Fig. 3.8a. In this, node v receives packets p_1 and p_2 from u and w respectively and checks for a coding opportunity. Recall that a coding opportunity exists only when all of v's neighbors can decode the prospective encoded packet $p_1 \oplus p_2$. In other words, all neighbors must know either p_1 or p_2, or both. Observe that the set of neighbors that can decode the packet is $\mathcal{N}(u) \cup \mathcal{N}(w)$. As a result, it suffices for v to confirm that the set $\mathcal{R} = \mathcal{N}(v) - \mathcal{N}(u) - \mathcal{N}(w)$ is an empty set to decide that $p_1 \oplus p_2$ is possible.

To increase the probability of producing an encoded packet our approach takes advantage of packet duplicates. Recall that a native packet waits in node v for a random time in order to find a coding opportunity. During this time v receives multiple copies of the native packet. Our observation is that each of these copies reaches a different part of v's neighborhood. As a result, it is possible to minimize $\mathcal{R}$ and thus increase the probability of finding a coding opportunity. Fig. 3.8b illustrates the advantages of considering packet duplicates. In the example, node v receives a duplicate of p_1 from node z while initially v received p_1 from node u. Taking into account the neighbors that indirectly received p_1 through node z, node v searches for coding opportunities by estimating the set $\mathcal{R}$=$\mathcal{N}(v)-\mathcal{N}(w)-(\mathcal{N}(u)\cup\mathcal{N}(z))$, which is now an empty set and therefore coding is possible. In general, when multiple duplicates of both p_1 and p_2 exist, v can detect coding opportunities using the set

$$\mathcal{R}=\mathcal{N}(v)-\mathcal{Z}_1-\mathcal{Z}_2 \tag{3.1}$$

where

$$\mathcal{Z}_m = \bigcup_{i\in\mathcal{H}_m} \mathcal{N}(i) \tag{3.2}$$

and $\mathcal{H}_m$ is the set containing all of v's neighbors that forwarded a copy of packet p_m.

An important feature of a coding process is to be able to find coding opportunities that involve more than two native packets, i.e., increase what is known as the *coding depth*. This feature is critical because it allows for further reduction of transmissions, thus improving energy efficiency. To illustrate that it is possible to use the proposed method to find coding opportunities involving multiple packets let us extend the example in Fig. 3.8b. Assume now that another native packet p_3 is available at node v and that we wish to check whether we can include it in the original coding $p_1 \oplus p_2$, i.e., create the encoded packet $p_1 \oplus p_2 \oplus p_3$. Recall that the prerequisite is that every node in $\mathcal{N}(v)$ should know at least two of the thee packets. Observe that nodes in the set $\mathcal{S} = \mathcal{N}(w) \cap (\mathcal{N}(u) \cup \mathcal{N}(z)) = \mathcal{Z}_1 \cap \mathcal{Z}_2$ have received both p_1 and p_2 therefore they fulfill the prerequisite. On the other hand, nodes in $\mathcal{N}(v) - \mathcal{S}$ (gray area in Fig. 3.8b) do not know both packets but have received either p_1 or p_2 (otherwise the coding of p_1 and p_2 could not be possible). Consequently, these nodes should known about p_3 in order for the triple coding to be possible. In other words, the set $\mathcal{R}$=$\mathcal{N}(v) - \mathcal{S} - \mathcal{Z}_3$ should be an empty set. The process can be repeated recursively to include more native packets in the encoding. In general, in order to include a native packet p_n in an encoding that already contains packets $p_1, p_2, \ldots, p_m$ node v should check whether

$$\mathcal{R} = \mathcal{N}(v) - \mathcal{S} - \mathcal{Z}_n = \mathcal{N}(v) - \bigcap_{j=1}^{m} \mathcal{Z}_j - \mathcal{Z}_n \tag{3.3}$$

is an empty set. Algorithm 3.3 presents the pseudocode of NOB-CR's coding procedure.

Algorithm 3.3 Pseudocode for detecting coding opportunities at node v.

DetectCodingOpportunities(packet p**, output Queue** $\mathcal{Q}$**)**

$\mathcal{Z}_p = GetReceiversOf(p)$
$\mathcal{S} = \mathcal{Z}_p$
$\mathcal{C} = \mathcal{N}(v) - \mathcal{S}$
$e = p$
for each native packet **q** $\in \mathcal{Q}$ **do**
  $\mathcal{Z}_q = GetReceiversOf(q)$
  $\mathcal{R} = \mathcal{C} - \mathcal{Z}_q$
  if $(\mathcal{R} = \emptyset)$ **then**
    $\mathcal{S} = \mathcal{S} \cap \mathcal{Z}_q$
    $\mathcal{C} = \mathcal{N}(v) - \mathcal{S}$
    $e = e \oplus q$
  end if
end for
return e

As we mentioned previously, our method renders the use of a neighbor reception table obsolete which significantly alleviates the related costs. Instead, we mostly rely on information already available through the underlying broadcast algorithm, i.e., neighborhood information. The only additional information that the method requires for a packet p is the set of previous hops $\mathcal{H}_p$, i.e., the nodes that forwarded a copy of p to v. This information can be used to estimate the set $\mathcal{Z}_p$ (procedure $GetReceiversOf(p)$ in the pseudocode) which consists of the nodes that have received p. This can be done using neighborhood information, i.e., $\mathcal{Z}_p = \bigcup_{\forall i \in \mathcal{H}_p} \mathcal{N}(i)$. Bear in mind that neighborhood information is updated on a periodic basis. Furthermore, each packet remains to the output queue for a limited period of time which is smaller than typical values for a neighborhood update interval. Therefore estimating the receivers of p by using $\mathcal{Z}_p$ is as accurate as the neighborhood information. However, note that due to periodic updating and mobility, $\mathcal{Z}_p$ is an approximation of the nodes that actually received p. In Section 3.5 we evaluate the impact of mobility on our coding approach and show that it is negligible even in networks of high node mobility. In addition, we examine in detail the pros and cons of using the proposed coding approach over the traditional one that utilizes a reception table.

Clearly, the advantage of our method is the limited cost for storing and updating $\mathcal{H}_p$. To explain this note that the basic data item for representing either a neighbor reception table or $\mathcal{H}_p$ is the id of a node. Now observe that for each packet p received by a node v the neighbor reception table may store up to $|N(v)|$ items in contrast to the $|\mathcal{H}_p|$ items stored in our approach. By definition, in a broadcast algorithm $|\mathcal{H}_p| \ll |N(v)|$. The total storage gain is $n_p \times (|N(v)| - |\mathcal{H}_p|)$, where n_p is the average number of packets for which reception information is stored at any given time. In Section 3.5 we show that in our simulation set-up the storage requirement may be reduced by three orders of magnitude. Another benefit stemming from the limited storage requirement is the positive impact on the processing cost for updating this information (e.g., locating the appropriate $\mathcal{H}_p$ and adding a node id). Last but not least, the proposed algorithm for detecting coding opportunities is entirely based on the manipulation of sets. Therefore it is possible to represent all sets using bitmaps and implement our algorithm as a sequence of fast bitmap operations.

3.4.3 Exploiting RAD to enhance the pruning process

The pruning efficiency of the underlying CDS-based algorithm is equally important to the coding process for minimizing transmissions. In general, algorithms that use the source-based CDS approach are considered to be the most efficient. In fact, NOB-CR builds on top of such an algorithm, i.e., PDP. Recall that in PDP a node v elects forwarders from $\mathcal{C}(v) = \mathcal{N}(v) - \mathcal{N}(u)$ in order to deliver a packet p to all nodes in $\mathcal{U}(v) = \mathcal{N}(\mathcal{N}(v)) - \mathcal{N}(v) - \mathcal{N}(u) - \mathcal{N}(\mathcal{N}(u) \cap \mathcal{N}(v))$, where u is the previous hop node that forwarded p to v. We take advantage of information already provided by the RAD mechanism to further enhance the pruning efficiency of PDP. More specifically, we make the observation that, when electing the forwarders, it is possible to take into account not only the previous hop node of p but all other nodes that relayed a duplicate of p while it was buffered due to the RAD mechanism. As a result, the set of nodes to be covered can be further reduced to:

$$\mathcal{U}(v) = \mathcal{N}(\mathcal{N}(v)) - \mathcal{N}(v) - \bigcup_{i \in \mathcal{H}_p} \mathcal{N}(i) - \bigcup_{i \in \mathcal{H}_p} \mathcal{N}(\mathcal{N}(i) \cap \mathcal{N}(v)) \tag{3.4}$$

where the set $\mathcal{H}_p$ is the set consisting of all previous hop nodes of p. The same idea can be applied to any source-based CDS algorithm that uses the same problem modeling such as DP, TDP and their derivatives. It is clear that reducing $\mathcal{U}(v)$ increases the probability of selecting fewer forwarding nodes from the candidate set $\mathcal{C}(v)$. Also note that the proposed method comes at no additional cost since the set $\mathcal{H}_p$ is used for detecting coding opportunities.

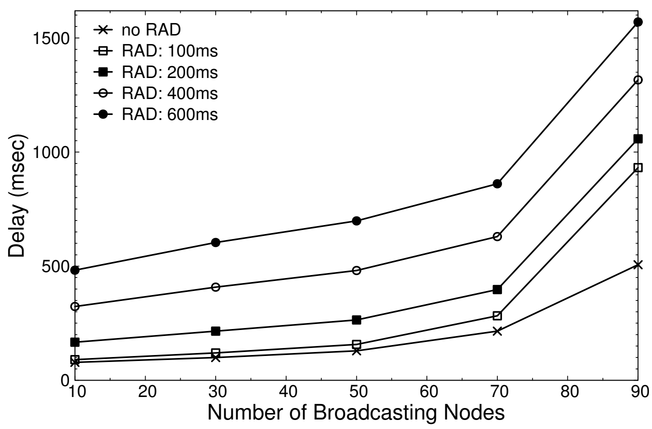


Figure 3.9: End-to-end delay performance of PDP MC/U under varying traffic load using different RAD intervals

3.4.4 Reducing delay through coded redundancy

The performance of XOR-based broadcast schemes heavily depends on the random assessment delay (RAD) applied before relaying a packet. In particular, high RAD values increase the coding opportunities and maximize the coding gain. This behavior was confirmed by the experimental results in Section 3.2.1 (Fig. 3.1 and Fig. 3.2) where increasing the RAD value from 200 to 400 ms substantially enhances CodeB's performance in terms of delivery ratio and energy efficiency. However, imposing a RAD to a packet in every node has a major impact on the end-to-end delay. Although RAD is usually short it results to an aggregation of a considerable end-to-end delay. To highlight this effect, we replayed the first experiment in Section 3.2.1 and recorded the end-to-end-delay for different values of RAD (Fig. 3.9). To avoid any interference caused by the performance degradation of the M/U criterion under high load we used the PDP algorithm with the MC/U criterion which can sustain performance in such conditions (Fig. 3.6, Section 3.3).

The results confirm our observation and expose the paramount importance of reducing the end-to-end delay when employing RAD. One way towards this direction is to increase the packet redundancy across the network. The rationale is that *using more duplicates per packet increases the probability of delivering a copy of the packet through a faster path*. However, the practice of increasing redundancy should be exercised with caution because it usually results in extra transmissions. This impacts the energy efficiency of the broadcast process as well as its delivery efficiency through the increase of collisions.

We propose a cost-free method for introducing packet redundancy across the network. This method, called *Coded Redundancy (CR)*, targets at increasing the duplicates of a packet and it is cost-free in the sense that it does not produce new transmissions. To accomplish that, CR introduces a new packet type called *gratis*. Gratis packets are non-coded packets for which the receiving node has not been selected for relaying them. Instead of dropping them, our method examines if these packets could be forwarded as part of already encoded packets, i.e., without cost. Packets already delivered to all neighbors of a node v do not qualify for marked as gratis because no delay improvement is feasible. Summarizing, node v can mark gratis packets using the following criterion:

Definition 3.1 (Gratis Marking Criterion). A native packet p is marked as gratis by a node v if there is at least one neighbor of v that has not received p and v is not a forwarder of p.

Note that it is possible for v to estimate whether a packet has not been received by all of its neighbors by utilizing a similar methodology as the one used for detecting coding opportunities (Section 3.4.2).

Many aspects of packet handling are the same for gratis and native packets. More specifically, in order to avoid loops, gratis packets are considered for forwarding only if they conform to the termination criterion. Accepted gratis packets are also temporarily buffered using the RAD technique. Then, if a coding opportunity is detected one or more gratis packets are relayed as part of an already encoded packet. Nonetheless, there are also differences in handling gratis and native packets. The first is that gratis packets are dropped if their buffering time expires without finding a coding opportunity. This is because they are meant to be forwarded only as part of an encoded packet. For the same reason gratis packets do not participate in the forwarding process, i.e., a node does not determine a set of forwarders for a gratis packet. Instead, as soon as a packet is marked as gratis it is always treated as gratis in subsequent hops. This approach guarantees that gratis packets are forwarded without any additional cost. Finally, each node applies the following rule upon reception of a gratis packet:

Definition 3.2 (Gratis Receiving Rule). The arrival of a gratis packet never triggers any modification to the termination criterion structures.

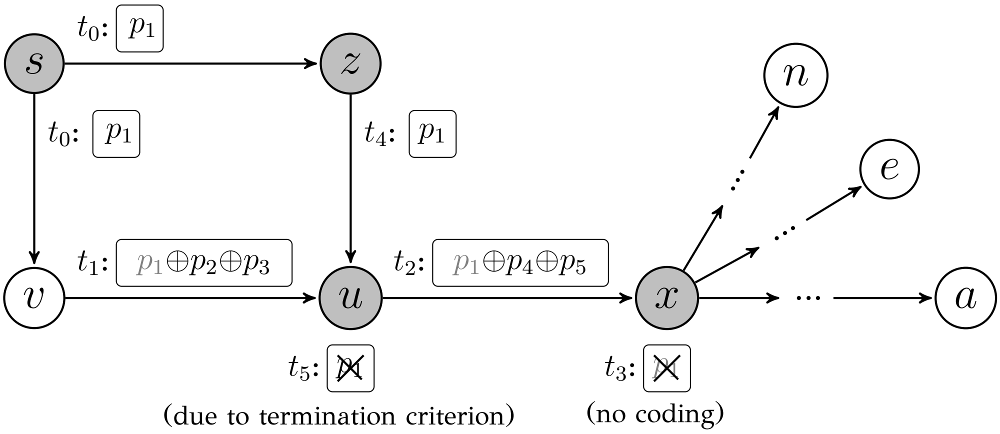


Figure 3.10: Example of failing to deliver packet p_1 to all network nodes when the "Gratis Receiving Rule" is not implemented.

This rule is directly associated with the termination criterion and indirectly affects the other packet duplicates that coexist in the network. If not applied, it could lead to situations where packets prematurely terminate their propagation in the network destroying the protocol's delivery efficiency. We further investigate the importance of this rule through an example. In Fig. 3.10, we monitor a packet p_1 that is propagated through a part of a network. Nodes in gray are selected as forwarders for p_1, while all other nodes act as passive receivers. Initially, at time t_0, nodes v and z receive a duplicate of p_1 from the source node s. Node v handles p_1 as gratis since it is not elected as a forwarder while for the opposite reason node z treats p_1 as native. At some point in time (t_1), node v detects a coding opportunity between the gratis p_1 and the already encoded packet $p_2 \oplus p_3$. As a result, it transmits the encoded packet $p_1 \oplus p_2 \oplus p_3$. Node u receives the encoded packet and successfully decodes it (assuming p_2 and p_3 are already known). Node u also handles the copy of p_1 as gratis. At time t_2 a coding opportunity for gratis packet p_1 at node u results in the transmission of a new encoded packet, i.e., $p_1 \oplus p_4 \oplus p_5$, where p_4 and p_5 are ordinary native packets previously received at node u. Node x receives the encoded packet and decodes it (assuming p_4 and p_5 are already known). Likewise, x handles p_1 as gratis searching for a proper coding opportunity to relay it. Assuming that p_1's buffering time expires with no coding opportunities, x drops p_1 (time t_3), terminating its dissemination to the rest of the network. At this point, the only way to deliver p_1 to nodes n, e and a is through the duplicate of p_1 that node z holds. At time t_4, the duplicate of p_1 is transmitted as a native packet because z is an elected forwarder for p_1. Node u, which is also an elected forwarder, successfully receives p_1 and has the opportunity to further relay it. However, its decision depends on the previous reception of the gratis copy of p_1 at time t_1. In particular, u employs the MC/U termination criterion which allows a node to relay only packets seen for the first time. In case that u has recorded the former arrival of p_1 at t_1 no forwarding is allowed due to the termination criterion. Consequently, the newly fetched duplicate of p_1 is dropped (time t_5). On the other hand, implementing the "Gratis Receiving Rule" resolves the situation. According to the rule, the former arrival of p_1 as a gratis packet is never registered by node u. As a result, the native copy of p_1 is forwarded to node x (time t_5). From that point, node x successfully propagates p_1 across all parts of the network.

The example in Fig. 3.10 also illustrates the benefits of using the coding redundancy technique. Packet p_1 reaches nodes u and x much faster when coding redundancy is utilized. More specifically, nodes u and x receive p_1 at time instances t_1 and t_2, respectively. On the contrary, without coded redundancy, p_1 reaches nodes u and x at t_4 and t_5, respectively.

The key functionality of the coding operation is the detection of coding opportunities. Extending this functionality to support gratis packets is not straightforward. This is because there are cases where gratis packets could destroy coding opportunities involving native ones, thus impairing the protocol's energy efficiency. Let us examine this problem through an example. Suppose that an intermediate node v is elected as forwarder for two native packets, i.e., p_1 and p_2, and that these packets can be mixed together (forming the encoded packet $p_1 \oplus p_2$) and forwarded with a single transmission. Things get complicated when a gratis packet p_3 is also present at v. Assuming that p_3 can be combined only with p_1 and that v chooses to combine p_1 with p_3 instead of p_2 then two transmissions are required, i.e., one for $p_1 \oplus p_3$ and one for p_2. To evade this problem, we introduce the following rule that every node applies when searching for coding opportunities.

Definition 3.3 (Gratis Coding Rule). Gratis packets participate in the coding process with lower priority than native packets.

Furthermore, in case that no coding opportunity is found among native packets then coding detection terminates without examining gratis packets. This policy in conjunction with the above rule prevents a gratis packet from directly being combined with existing native ones, leaving them available for possible future encodings. The only exception of combining a gratis packet with a native one is when the buffering time of the latter expires. In that case there is no possibility for the native packet to participate in any future coding opportunity. As a result, encoding the native packet with the gratis one does not destroy future coding opportunities.

3.5 Evaluation

To evaluate the performance of NOB-CR, we compare it with two algorithms. The first one is CodeB [21] which is the most representative of XOR coding-based broadcast algorithms. The second algorithm is PDP [27] which, despite the fact that does not use any type of coding, is well-known for its energy efficiency. We use two variants of PDP, namely PDP M/U and PDP C/U, in order to examine how the termination criterion affects the overall performance.

Set up and methodology: All investigated algorithms are implemented in the ns2 simulator [96], using the CMU extension. We present the average values over 20 independent simulation runs, each with a duration of 300 seconds. The confidence level, for all reported confidence intervals, is 95%.

Table 3.1: Simulation parameters

Simulation Time	300 sec
Number of Trials	20
Confidence Level	95%
Transmission Range (R)	250 m
Bandwidth	2 Mb/sec
Number of Nodes (N)	60 - 250
Avg. Neigh. Size	15, 30
Node Speed	0 - 20 m/sec
Broadcast Sessions (S)	10 - 90
Broadcast Rate (λ)	0.1 - 8 pkts/sec
Packet Size	256 Bytes
Hello Interval (t_H)	1 sec
Random Assessment Delay (t_{RAD})	100 - 600 msec

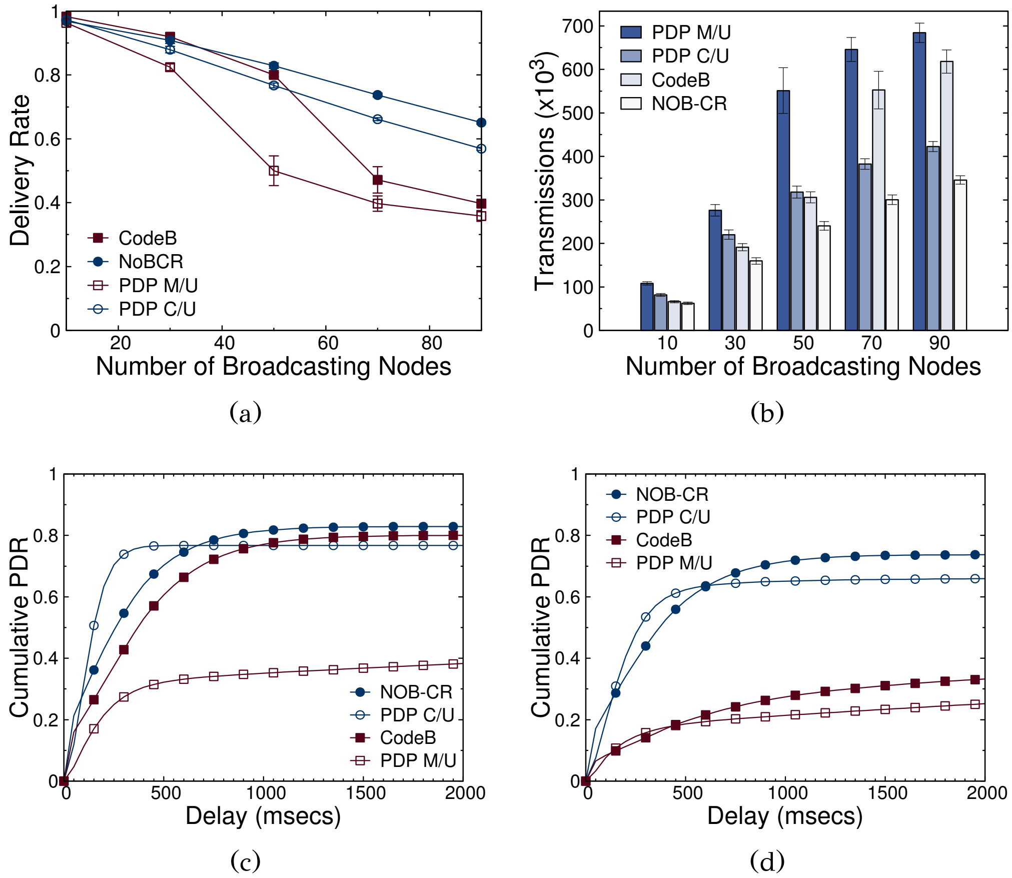


Figure 3.11: Performance for different levels of offered load in the "sparse" topology (N=100, max speed:1 m/sec, t_{RAD}=400msec): (a) Delivery rate vs broadcasting sources (b) Avg. number of transmissions vs broadcasting sources (c) Cumulative PDR vs end-to-end delay ($S=50$ sources) (d) Cumulative PDR vs end-to-end delay ($S=70$ sources).

Network model: The default number of nodes is 100, the propagation model is the TwoRay ground with a transmission range of 250m and the nominal bit rate is 2Mbps. The nodes move in a square area according to the Random Waypoint (RW) model [97]. To avoid transient artifacts in nodes' movement, we use the perfect simulation algorithm [99]. We examine two network topologies; "dense" and "sparse". Similar to [21], in the "dense" topology, the average neighborhood size is 30 while in the "sparse" topology it is 15. Note that we could not use a lower density in the "sparse" scenario since in such a case frequent partitions occur. Simulations confirmed that in the "sparse" scenario there exist many nodes (those moving near the boundaries) that experience very low connectivity. All algorithms collect neighborhood information by periodically exchanging hello messages with an interval (t_H) of

1 second.
Network traffic: Traffic is generated by broadcast sessions, each stemming from a different source node and starting at a random time. Although we use a variable number of sources, each one producing packets at a constant rate of $\lambda = 1$pkt/sec, the default value is 50. The size of each message is set to 256 Bytes.
Coding parameters: All coding schemes under evaluation use the RAD technique to maximize the probability of coding opportunities. According to RAD, each node delays every packet it receives for a random delay in $[0, t_{RAD}]$. The default t_{RAD} value in our simulations is 400 msec. Due to storage limitations, all coding based algorithms buffer incoming packets for a limited time interval (B_T) in order to enable decoding. This time interval is highly correlated with the time period for which information inside the neighbor reception table is available (R_T). We set both time intervals to 5 sec for the CodeB algorithm in order to increase the benefits of network coding and be realistic at the same time. On the contrary, we set only the B_T interval for NOB-CR algorithm since it operates without a neighbor reception table. After extensive experimentation, we found that a B_T value equal to 2 sec is sufficient for NOB-CR's encoding/decoding operation. The small B_T value utilized by NOB-CR is preferable because it allows for efficient management of the limited storage space in the network nodes.

Fig. 3.11 illustrates the performance of all investigated algorithms in the sparse topology and under different levels of offered load (variable number of broadcasting sources). As discussed in section 3.2.1, our experimental results reveal the ineffectiveness of the M/U criterion that induces the performance breakdown of the PDP M/U and CodeB schemes. More specifically, as the load increases the algorithms that utilize the M/U criterion lose their pruning efficiency producing a large number of

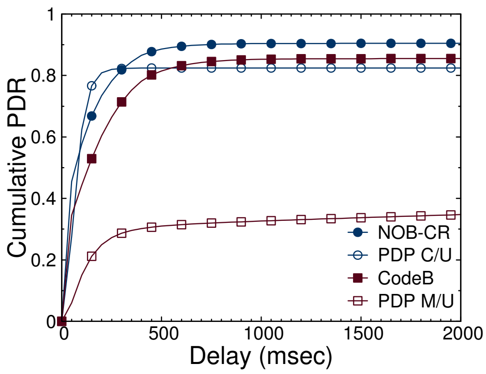
Table 3.2: Reduction (%) of transmissions for NOB-CR compared to other algorithms

# sources	PDP M/U	PDP C/U	CodeB
10	44.7%	27.0%	7.7%
30	44.1%	29.8%	17.9%
50	57.8%	26.9%	22.7%
70	55.1%	24.2%	45.5%
90	51.3%	21.1%	45.3%

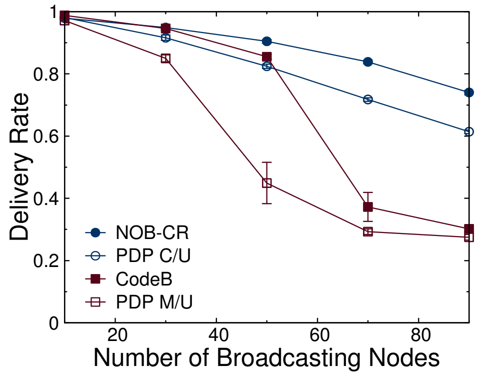
transmissions (Fig. 3.11b). PDP M/U fails to prune transmissions when the number of source nodes exceeds 30. On the other hand, network coding enables CodeB to maintain its pruning efficiency when traffic is produced by up to 50 sources (half of network nodes). However, as the congestion level increases, the excessive number of forwarding decisions taken by both algorithms induce transmission failures due to packet collisions. As a result, their delivery performance deteriorates (Fig. 3.11a). NOB-CR outperforms all algorithms both in terms of delivery efficiency and number of transmissions. Even in the extreme case of 90 broadcasting sources, which is close to the all-to-all communication paradigm, NOB-CR delivers $\sim$66% of the traffic while the M/U based schemes reach less than 40% of the network nodes (Fig. 3.11a). This justifies our approach to combine XOR network coding with an termination criterion other than M/U. At the same time, NOB-CR is exceptionally energy efficient. Table 3.2 presents NOB-CR's energy gains that derive from reducing the total number of transmissions. Against PDP variants, NOB-CR reduces the total number of transmissions by 21% in the worst case. Compared to CodeB, the energy gains become noticeable when the broadcasting sources are more than 30 (18% to 45%).

Fig. 3.11c and 3.11d depict the cumulative packet delivery ratio (PDR) versus the end-to-end delay, i.e., the fraction of packets delivered within a delay limit, when the broadcasting sources are 50 and 70, respectively. We choose this presentation style in order to capture both the delivery efficiency and the timeliness of each algorithm. Again, the results provide a confirmation of the ineffectiveness of the M/U criterion. Even when there are only 50 sources in the network (Fig. 3.11c) PDP M/U collapses while XOR coding allows CodeB to achieve a competitive performance. However, when the offered load increases (Fig. 3.11d), both schemes that use the M/U criterion deliver less packets with higher delay due to the increased number of transmission failures. NOB-CR outperforms both schemes not only in delivering more packets, but in delivering them faster. The reason is twofold; the utilization of the MC/U criterion and the coded redundancy mechanism. Interestingly, NOB-CR's delay profile is comparable to that of PDP C/U (especially in the case of 70 sources) despite the fact that simple PDP schemes operate without the RAD mechanism that significantly increases the end-to-end delay. Moreover, we found that under high load (>70 sources) NOB-CR is at least as fast as PDP C/U. This is due to NOB-CR's pruning process that efficiently decreases transmission failures allowing for timely packet delivery.

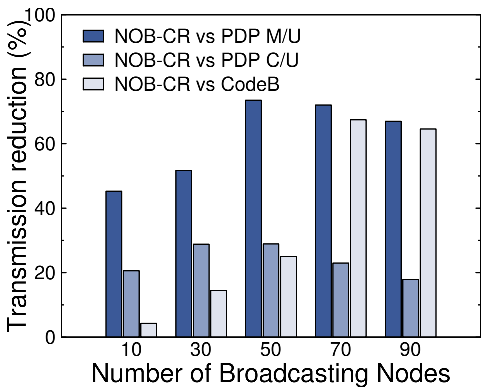
We also experimented on increasing the packet generation rate λ while keeping



(a)



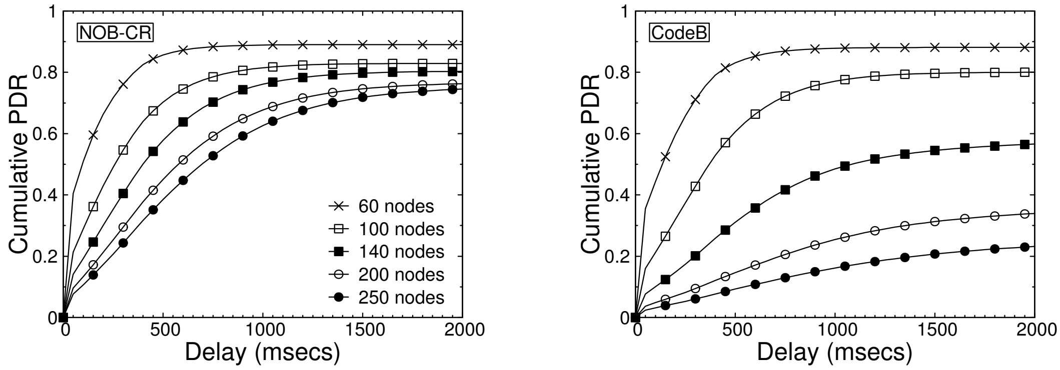
(b)



(c)

Figure 3.12: Performance for different levels of offered load in the "dense" topology (N=100, max speed:1 m/sec, t_{RAD}=400msec): (a) Cumulative PDR vs end-to-end delay (S=50 sources) (b) Delivery rate vs sources (c) Transmission reduction vs sources

the number of sources constant, e.g., $S = 10$ sources. In this way it is possible to change the offered load but limit the coding capability of algorithms. This is because coding opportunities heavily depend on the number of packet flows, i.e., the number of sources. The obtained results are qualitatively similar to the previous experiment therefore we do not include the corresponding performance plots. In summary, for a low λ, the energy gains for all coding enabled schemes are limited. This is because the number of coding opportunities is rather small since fewer packets coincide in the network. As the offered load increases, the benefits of network coding become more evident. However, after the breaking point of λ=5 packets per second the delivery performance of CodeB deteriorates as a result of the increased levels of congestion. On the other hand, NOB-CR exhibits a remarkable resilience to congestion achieving the best performance in terms of delivery ratio and energy efficiency.



(a) (b)

Figure 3.13: Performance when the network size scales up ("sparse" topology, S=50 sources, max speed:1 m/sec, t_{RAD}=400msec): Cumulative PDR vs end-to-end delay for a) NOB-CR, and (b) CodeB.

Next, we used a variable number of sources to test all schemes under different levels of offered load in the dense topology (Fig. 3.12). As expected, when the offered load is low, the delivery efficiency of all algorithms improves compared to the sparse topology (Fig. 3.12a). This is because the higher number of neighbors results in increased packet redundancy, making delivery more probable. Furthermore, the diameter of a denser network is smaller. Consequently, as depicted in Fig. 3.12a, all schemes deliver packets faster than in the sparse case (Fig. 3.11c). However, despite its positive effects, there is also a downside of the increased neighborhood size; under high load the probability of transmission failures due to collisions is higher. This is because more packet duplicates are created, resulting in congestion. As a result, schemes that do not efficiently prune transmissions collapse (Fig. 3.12b) as the offered load increases. Interestingly, the performance degradation is more acute and develops more quickly (lower traffic levels) than in the case of sparse topology because congestion is more severe. Fig. 3.12c illustrates the reduction of the total number of transmissions achieved by NOB-CR compared to all other schemes. As anticipated, the higher gains are witnessed when the offered load is high where NOB-CR prunes 60% more transmissions than CodeB. At the same time, it delivers over 40% more packets (Fig. 3.12b).

In the following we focus on the more challenging scenario of sparse networks. The next set of experiments assesses the performance of NOB-CR and CodeB when scaling the network up. Towards this direction, we conducted simulations with an

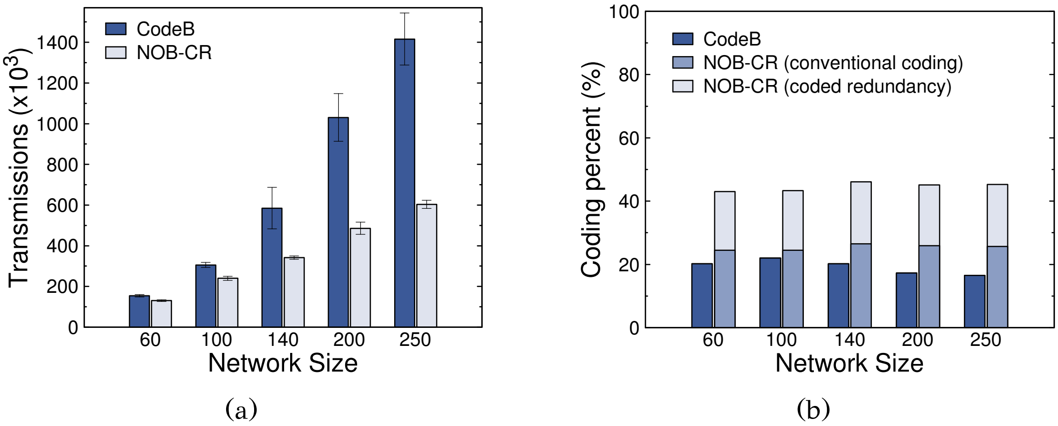


Figure 3.14: Performance when the network size scales up ("sparse" topology, S=50 sources, max speed:1 m/sec, t_{RAD}=400msec): (a) Avg. number of transmissions, and (b) Fraction of encoded packets vs network size.

increasing number of nodes. At the same time, we also expand the network area in which the nodes move in order to keep the average neighborhood size fixed. Fig. 3.13 illustrates the cumulative packet delivery ratio versus the end-to-end delay for various network sizes. NOB-CR and CodeB present a similar behavior when the network size is small, i.e., 60 nodes. As the number of nodes increases, the performance of CodeB quickly deteriorates and finally collapses when the network size exceeds 140 nodes (Fig. 3.13b). On the other hand, NOB-CR exhibits a remarkable durability and its performance, in terms of both delivery ratio and end-to-end delivery delay, degrades much more slower and smoother (Fig. 3.13a). The witnessed performance degradation for both schemes is reasonable since in our experiment we fix the network density. As a result, the diameter of the network increases with the number of nodes and therefore it is more difficult to reach some destinations. Notwithstanding, NOB-CR is very efficient in reducing transmissions (Fig. 3.14a) and therefore alleviates congestion. Thus, failures due to collisions are minimized and so is the impact of the increasing network diameter. For example, in the case of 60 nodes NOB-CR produces ~16% less transmission than CodeB, while in case of 140 nodes transmissions are reduced by $\sim$42%. This increasing difference in the pruning efficiency of the two schemes not only justifies the higher deliver ratio of NOB-CR but is also in accordance with the increasing difference in the delivery efficiency of the two schemes. Overall, NOB-CR loses less than 15% of its delivery efficiency in the most demanding scenario of 250 nodes. On the other hand, in the same scenario CodeB loses more than 60%.

Going back to Fig. 3.14a, it is worth pointing out that the pruning efficiency of NOB-CR is increasingly better compared to that of CodeB. As expected, the total number of transmissions increases because more forwarders are required for a bigger network with a larger diameter. However, NOB-CR manages to suppress this increase and therefore broaden its advantage over CodeB. The reason for this result is not only the better pruning operation of the MC/U termination criterion but also its more efficient coding operation. To illustrate this, we present in Fig. 3.14b the number of transmitted encoded packets as a percentage of the total (encoded and native) number of transmitted ones. Furthermore, in the case of NOB-CR we present two separate classes on encoded packets. The one consists of encoded packets containing at least one gratis packet (Coded Redundancy) while the other refers to typical encodings involving only native packets (conventional coding). Recall that besides the packets encoded with the conventional mechanism, those produced with the Coded Redundancy method may also reduce the number of transmissions. This is because the latter packets may also contain two or more native packets. Clearly, NOB-CR not only consistently performs about twice as many encodings as CodeB does, but its coding operation is also stable. In contrast, the percentage of coded packets for CodeB decreases in bigger networks which implies that the coding operation is hampered by the underlying broadcast mechanism. Another important finding is that NOB-CR also performs a significant number of encodings involving gratis packets. Although this type of encodings is important for reducing end-to-end delay (as discussed in Section 3.4.4), it is also beneficial for enhancing delivery efficiency because gratis packets increase packet redundancy across the network without any additional cost. Therefore, the Coded Redundancy mechanism also contributes to NOB-CR's superior delivery efficiency witnessed in Fig. 3.13.

In the last experiment we assess the delivery efficiency under different levels of mobility (Fig. 3.15). Clearly, increased mobility impacts the performance of both NOB-CR and CodeB. The reason is that both schemes rely on the PDP scheme. The latter uses neighborhood information for electing the optimal forwarders. This information becomes outdated more quickly when mobility increases. Note that, as discussed in Section 3.4.2, NOB-CR uses neighborhood information also for detecting coding opportunities while CodeB uses a neighbor reception table. Nonetheless, NOB-CR manages to outperform CodeB regardless of the mobility level and even in scenarios of very high mobility. More specifically, it exhibits faster packet delivery (Fig. 3.15)

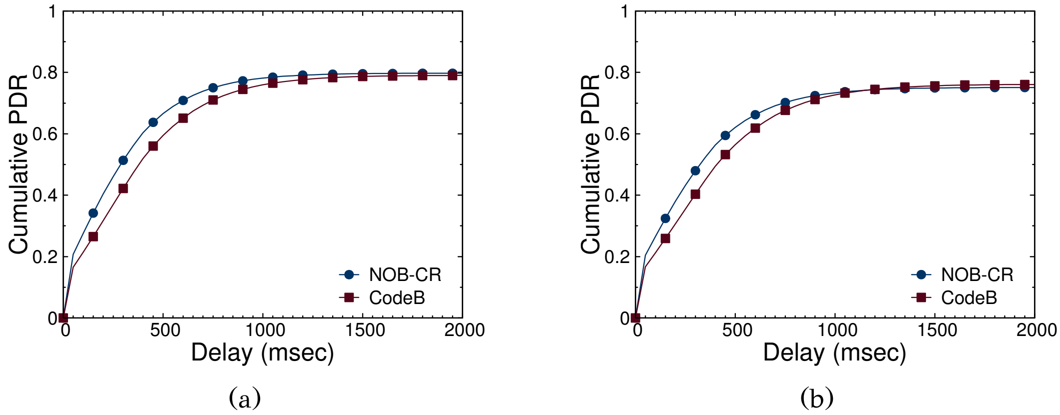


Figure 3.15: Performance under different levels of mobility ("sparse" topology, N=100, S=50 sources, t_{RAD}=400msec): Cumulative PDR vs end-to-end delay for Node speed: (a) $2-10$ m/sec (b) $10-20$ m/sec.

using more than ∼20% less transmissions compared to CodeB (Table 3.3).

Besides the performance evaluation of the complete NOB-CR algorithm it is interesting to shed some more light on the advantages and the limitations of the lightweight coding mechanism proposed in Section 3.4.2. In other words, we wish to investigate the storage and processing gains as well as the coding efficiency of the approach, i.e., the ability to find coding opportunities without those ending up in decoding failures, compared to the traditional approach that uses a neighbor reception table. To this end and in order to rule out any other interfering factor, we compare NOB-CR with a modified version of it that uses the typical neighbor reception table instead of the lightweight coding mechanism.

Clearly the advantage of the lightweight coding mechanism is the reduced storage and processing requirement as discussed in Section 3.4.2. To quantify this advantage we monitored the storage requirement for the two coding schemes both in a "sparse" and a "dense" network (Fig. 3.16). We express the storage requirement in terms of data items, where a data item represents the memory required for storing the id of a node. We follow this approach in order to have a fair comparison that does not depend

Table 3.3: NOB-CR's energy gains over CodeB under different mobility levels

Mobility	$0-1$ m/sec	$2-10$ m/sec	$10-20$ m/sec
Gain	21.45%	23.94%	27.83%

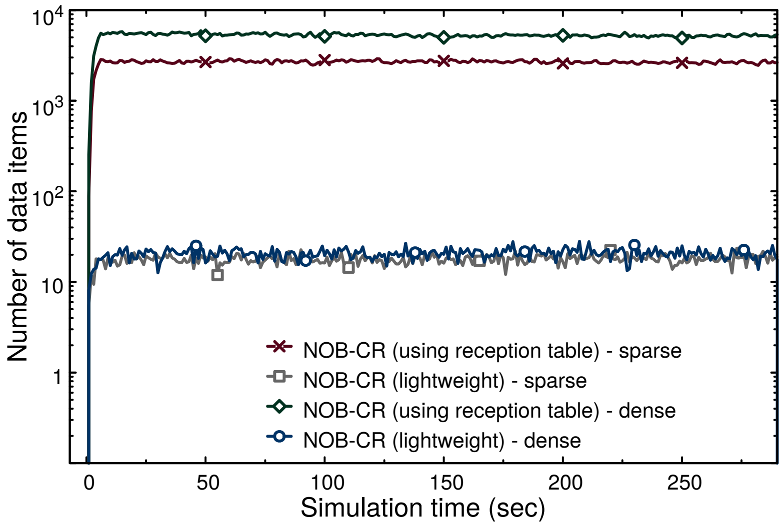


Figure 3.16: Average number of items stored by a node over simulation time when using neighbor reception table vs lightweight implementation for “sparse” and “dense” topologies (max speed: 1 m/s, N=100, S=50, t_{RAD}=400ms).

on the data representation method. Evidently, the storage demand for the lightweight approach is significantly smaller (up to three orders of magnitude) compared to the case of a neighbor reception table. As discussed in Section 3.4.2, this also has a positive impact on the required processing. An interesting and useful feature is that although the storage demand of the traditional coding approach increases (almost doubles) in a dense topology this is not true for the lightweight implementation. This is reasonable because in the latter case the storage demand depends on the rather stable number of received duplicates and not on the neighborhood size.

Regarding the coding efficiency and in order to understand the differences between the two approaches recall that the neighbor reception table is populated upon the reception of a packet, i.e., it uses the neighborhood information at the time of packet reception (e.g., t_0). Instead, our method uses the neighborhood state at a later time $t_1 > t_0$ (when a coding opportunity is present) as an estimation of the neighborhood at t_0. Clearly, this estimation becomes less accurate when mobility increases due to the increased invalidation rate of neighborhood information or when a packet waits a longer time for a coding opportunity (i.e., $t_1 - t_0$ increases). We expect the lightweight coding operation to be challenged in the aforementioned conditions so we investigate the extent at which this happens. First we compare the two schemes under different levels of node mobility (Fig. 3.17). Furthermore, we examine both “sparse” and “dense” topologies that correspond to different neighbor-

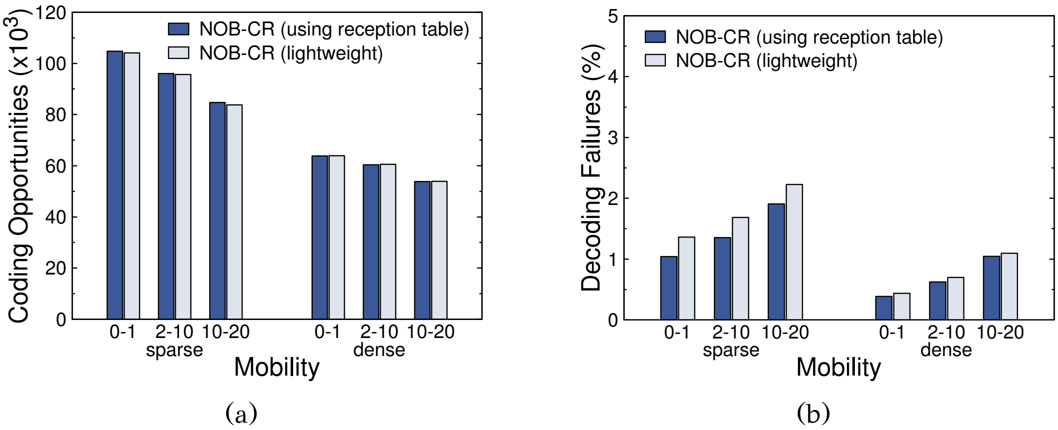


Figure 3.17: Coding Efficiency of lightweight implementation vs neighbor reception table with respect to mobility in "sparse" and "dense" topologies ($N=100$, $S=50$, t_{RAD}=400ms): (a) coding opportunities (b) percentage of decoding failures.

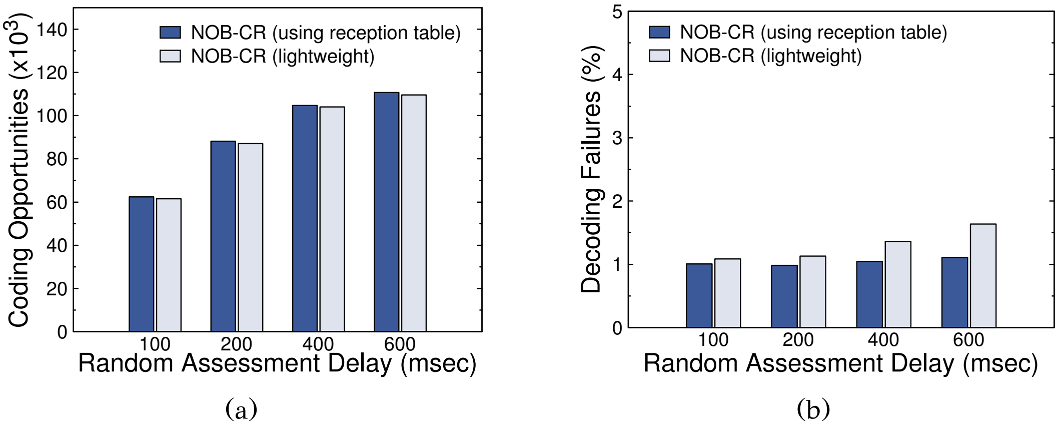


Figure 3.18: Coding Efficiency of lightweight implementation vs neighbor reception table with respect to RAD ("sparse" topology, max speed: 1 m/s, N=100, S=50): (a) coding opportunities (b) percentage of decoding failures.

hood sizes. As expected, using a neighbor reception table is slightly better than the proposed lightweight method from a coding point of view, i.e., in terms of both detected coding opportunities (Fig. 3.17a) and decoding failures (Fig. 3.17b). However, the difference is minor even in high levels of mobility. What is more important is that the slightly better coding operation of NOB-CR with reception table translates to a poor improvement in delivery ratio (Table 3.4) which peaks at $\sim$0.65%. This along with the advantages of the lightweight implementation justifies our approach to choose the latter over the traditional coding approach.

Table 3.4: Delivery Ratio Reduction (%) of NOB-CR compared to NOB-CR with neighbor reception table

Mobility		DR Reduction (%)	RAD	DR Reduction (%)
0-1 m/s	"dense"	0.1002	100 ms	0.1342
2-10 m/s		0.0686	200 ms	0.3426
10-20 m/s		0.0921	400 ms	0.5302
0-1 m/s	"sparse"	0.5302	600 ms	0.8232
2-10 m/s		0.5677		
10-20 m/s		0.6474		

Similar results are witnessed when we modify the RAD value, that is the maximum time that we allow a packet to wait for a coding opportunity (Fig. 3.18). As explained previously, in the lightweight approach late coding opportunities, i.e., those appearing significantly later than the reception of the involved packets, run an increased risk of resulting to a decoding failure. However, this does not significantly impact the performance (Table 3.4). Even when RAD is as high as $600\ ms$ the delivery ratio reduction is as low as $\sim 0.82\%$ compared to the traditional case. Note that $600\ ms$ is already a significantly high RAD value and results in increased end-to-end delay (Fig. 3.9). Using a higher value would further increase end-to-end delay thus destroying the broadcast process. Finally, we observed similar results when increasing the network size or the number of broadcasting sources.

3.6 Summary

XOR-based coding has been successfully used to enhance the energy efficiency of broadcasting in mobile ad hoc networks. We demonstrated, through extensive experimentation, that using the M/U termination criterion in the baseline broadcast algorithm severely impairs performance in several cases. Unfortunately, we found that alternative termination criteria that are proposed in the literature are not capable of efficiently supporting the coding process. As a result, we introduced a novel termination criterion that is fully compatible with XOR-based network coding. Furthermore, we revised some of the coding internals to enhance performance and at the same time reduce complexity. More specifically, we proposed a lightweight method for detecting coding opportunities that operates without a reception table. We introduced the concept of "Coded Redundancy" that reduces the end-to-end delay by increasing the packet redundancy across the network at no additional cost. Finally, we improved the forwarder election process of the proposed algorithm by exploiting information that was originally used only for coding purposes. The efficiency of NOB-CR, the algorithm that incarnates all the aforementioned modifications, was demonstrated through extensive simulations.

Chapter 4

Random Linear Network Coding over Deterministic Broadcasting

Random Linear Network Coding (RLNC) has been successfully used for efficient broadcasting in wireless multi-hop networks. In this chapter, we focus on the problem of multi-source broadcasting using RLNC in mobile ad hoc networks. Initially, we develop an analytical model which reveals that the usual approach to combine RLNC with probabilistic forwarding may significantly impact RLNC's performance. Motivated by this finding, we take the novel approach to combine the resilience offered by RLNC with the pruning efficiency of CDS-based broadcasting.

4.1 Preliminaries

Before delving into the theoretical analysis of RLNC and its comparison to XOR-based coding, we first provide a short description of their coding internals and discuss the related overhead introduced by each scheme.

4.1.1 RLNC in a nutshell

RLNC is based on the observation that a linear code, i.e., to linearly combine packets based on the theory of finite fields, is adequate for providing the benefits of network coding [28]. In order to practically implement RLNC, *native*, i.e., non encoded, packets need to be organized in groups, the so called *generations* [22]. Then, an encoded packet is produced as a linear combination of the native packets in a generation, using $\mathbb{F}_{2^s}$ arithmetic. That is, each native packet p_i is first partitioned into symbols of s bits and then the k-th symbol of the encoded packet $e(k)$ is calculated as $e(k) = \sum_{i=1}^{g} c_i p_i(k), \forall k$, where $p_i(k)$ is the k-th symbol of the i-th native packet and g is the number of packets in a generation. The set of coefficients $\langle c_1, c_2, \ldots, c_g \rangle$, which is called *the encoding vector*, is randomly selected from the finite field $\mathbb{F}_{2^s}$ and appended to the packet header. The random selection provides the required flexibility for distributed implementations. It is also sufficient since the probability of producing linearly dependent packets depends on the field size 2^s [29] and is negligible even for small values of s [100]. Decoding packets of generation i at node v is performed by means of a decoding matrix $\mathbb{G}_{v,i}$. The matrix is populated by *innovative packets*, i.e., the encoded packets that increase the rank of $\mathbb{G}_{v,i}$. Decoding is accomplished by performing the Gaussian elimination when $\mathbb{G}_{v,i}$ has a full rank. It is also possible to decode a subset of packets when a full rank sub-matrix of $\mathbb{G}_{v,i}$ exists (*partial decoding*). Furthermore, encoding at an intermediate node is possible without the need of decoding the native packets since a new encoded packet may be produced by linearly combining other encoded packets.

4.1.2 XOR coding principles

In XOR-based coding, each node collects information about the native packets received by its neighbors. The information is collected by overhearing the wireless medium and by exploiting local connectivity information. Let $\mathcal{B}_u$ denote the buffer containing the native packets received by node u and $\mathcal{B}_u^v$ denote v's view of the same buffer. A node u may choose a set of native packets $\mathcal{B}' \subseteq \mathcal{B}_u$ and produce an encoded packet, by using bitwise XOR, in the presence of a *coding opportunity*. This means that a set $\mathcal{B}' \neq \emptyset$ can be found such that, according to u's view, each node $v \in \mathcal{N}(u)$ has received at least $|\mathcal{B}'|-1$ of the native packets in $\mathcal{B}'$, i.e., $|\mathcal{B}_v^u \cap \mathcal{B}'| \geq |\mathcal{B}'|-1, \forall v \in \mathcal{N}(u)$. XOR-based coding works on a hop-by-hop basis, i.e., a receiver of an encoded packet should be able to decode it. Successful decoding depends on the consistency of $\mathcal{B}_v^u$, i.e., whether $\mathcal{B}_v^u = \mathcal{B}_v$. Decoding failures occur when $|\mathcal{B}_v \cap \mathcal{B}'| < |\mathcal{B}'|-1$ and result in the loss of all the encoded packets.

4.1.3 Complexity of coding schemes

Both RLNC and XOR-based coding entail some communication, processing and storage space overhead. About the communication overhead, both schemes assume that an encoding vector is included in the header of each encoded packet. The processing overhead in RLNC is related to the implementation of the Gaussian elimination. Its complexity on a matrix with rank r is $\mathcal{O}(r^3)$, however, implementing partial decoding can alleviate the decoding cost. On the other had, in XOR-based coding, the processing burden lies in finding coding opportunities. The optimal XOR-based algorithm is shown to be NP-hard, however, efficient suboptimal algorithms for finding coding opportunities have been proposed [21]. Finally, while in RLNC each node is required to store all packets in a generation, in XOR-based coding, each node should store a list of recently received packets (in order to enable decoding) along with information about the packets received by each of its neighbors. To summarize, in our view, none of the above schemes is profoundly better than the other, in terms of the related overheads. Furthermore, the actual cost of each scheme depends on the implementation specifics, making it impossible for a more detailed comparison. Nonetheless, we will show, throughout the rest of the manuscript, that we take all necessary action to minimize the cost of the proposed scheme, e.g., we enable partial decoding, minimize the size of encoding vectors, keep the generation size small, etc.

Table 4.1: Notation used in the Analysis

g	Generation size
$\mathbb{G}_{v,i}$	Decoding matrix of node v for generation i
N	Number of nodes in network
$\mathcal{N}(v)$	Set of node v's direct neighbors
ω	Probability of forwarding a message
ρ	Probability of transmission failure

4.2 Analysis of RLNC's coding features

As mentioned previously, the driving force of this work has been the observation that RLNC is capable of providing robust coding features. To validate this view, we develop an analytical model that portrays the performance of RLNC in the context of broadcasting. Before continuing with the analysis, we briefly describe the system model. Table 4.1 summarizes the notation used in this chapter.

4.2.1 System model

Network Model: We consider multihop wireless ad hoc networks. We model such a network as a random geometric graph (RGG) [101]. The nodes are deployed over an area $A \times A$. We focus on the generic approach of uniform node deployment which captures static and some cases of mobile networks (e.g., when node movement follows the random direction model [102]). Moreover, our study is valid for the node distribution resulting from the random waypoint movement model [97]. A link between a node pair (u, v) exists when the Euclidean distance $d(u, v)$ is smaller than a transmission range R. The neighborhood $\mathcal{N}(v)$ of a node v is the set of nodes connected to v with a link, i.e., $\mathcal{N}(v) = \{u \mid d(u, v) \leq R\}$.

Loss Model: The network consists of unreliable links. The transmission of a packet over a link fails with probability ρ, which is independent of other links. This assumption is common in the literature [64, 103] for wireless links without correlated shadowing and severe interference.

Broadcast sources: We assume that multiple sources exist in the network. Created packets are grouped in generations of size g. For each packet added to a generation, the source broadcasts an encoded packet that is a random linear combination of the generation contents.

Forwarding process: When receiving an innovative packet, each node implements a simple probabilistic forwarding protocol, i.e., forwards a new encoded packet with probability ω.

4.2.2 Distribution of the number of message copies

The properties of an RGG are critical for the performance of RLNC. More specifically, we will show that the performance of RLNC depends on the number of message copies that a node d receives when a source s broadcasts a message without using network coding. Let us model this number as a discrete random variable (RV), denoted as X. We aim at identifying a good approximation for the probability mass function (pmf) of X. We first assume lossless links (i.e., ρ=0) and later generalize our model to include the case of $\rho \neq 0$. First, note that X is conditional on the number of d's neighbors that receive at least one copy of the broadcast message. If Y is a RV representing the latter number, then X follows the binomial distribution with parameters Y and ω, i.e., $X \sim B(Y, \omega)$. This is because the forwarding decisions made by neighbors are independent. Then, we focus on research efforts that have established, by means of percolation theory, that probabilistic forwarding presents a bimodal behavior [104]. That is, if we consider the number of nodes (r) that receive the message, then, with high probability, either r=0 or $r = \alpha, \{\alpha \in \mathbb{N} : 0 < \alpha \leq N\}$. The probability that r has any other value is negligible. The actual probability of $r = 0$ (and the complementary of r=α) as well as α depend on the network properties. Moreover, in most cases, $\alpha \to N$, i.e., either none or nearly all the nodes receive the message [104]. By extending this finding, we make the observation that Y also exhibits a near bimodal behavior, therefore a good approximation for Y's pmf is:

$$\mathrm{P}\{Y = k\} = \begin{cases} \phi & k = 0 \\ 1 - \phi & k = |\mathcal{N}(d)| \\ 0 & \text{otherwise} \end{cases} \tag{4.1}$$

where $\{\phi \in \mathbb{R} : 0 \leq \phi \leq 1\}$. The rationale behind this approximation is simple; due to spatial proximity of the nodes that belong to $\mathcal{N}(d)$, all of them will lie either in the set of receivers or in the set of non-receivers with high probability. Using (4.1), it is easy to show[1] that:

$$\mathrm{P}\{X=k\}=\begin{cases}\phi+(1-\phi)(1-\omega)^{|\mathcal{N}(d)|} & k=0\\ (1-\phi)\omega^k(1-\omega)^{|\mathcal{N}(d)|-k} & 0<k\leq|\mathcal{N}(d)|\end{cases} \tag{4.2}$$

To validate this distribution, we need to examine it under various combinations of ω, $|\mathcal{N}(d)|$, the average node degree and the hop distance (H) between s and d. This is because ϕ, in analogy to the bimodal property [104], also depends on those parameters. Therefore, we adopt the following strategy; we simulate probabilistic broadcasting for various values of ω in RGGs deployed in areas of various sizes (we use the normalized value $\hat{A}=A/R$ to denote the size of the network area). Then, for each $\langle\omega,\hat{A}\rangle$ pair we execute 10^6 simulations. In each simulation we create a new RGG, randomly select a source-destination pair (s,d) and record the number of message copies received by d. For each combination of $\omega,\hat{A},H,|\mathcal{N}(d)|$, we construct the statistical pmf based on the frequency observed for each value of X. Let $\widetilde{\mathrm{P}}\{X=k\}$ denote this pmf. We approximate ϕ in (4.2) by solving $\phi+(1-\phi)(1-\omega)^{|\mathcal{N}(d)|}=\widetilde{\mathrm{P}}\{X=0\}$. Then, we calculate the total variation distance (d_{TV}) [105] between (4.2) and $\widetilde{\mathrm{P}}\{X=k\}$, i.e., the maximum difference between the probabilities assigned by the two distributions to the same event.

Table 4.2 reports d_{TV} values for networks with $\hat{A}=\{4,6,8\}$ and $N=100$. The lower value of $\hat{A}$ corresponds to relatively dense networks while the highest has been chosen so that the resulting networks are as sparse as possible but not partitioned with high probability [106]. We have obtained similar results for various values of N, however, for brevity, we report only the results for $N=100$. According to the presented results, (4.2) provides a satisfactory approximation for the purposes of the following analysis. As a final note, (4.2) can be generalized to include the case of transmission errors, i.e., when $\rho\neq0$. Simulation results (omitted for brevity) confirm that the approximation is still good if ω is replaced by $\omega(1-\rho)$. Furthermore, we have also obtained results confirming that (4.2) is still valid when nodes' positions follow the random waypoint distribution [97]. This is in accordance with a similar observation regarding the bimodal behavior of probabilistic broadcasting under the same node distribution [104].

[1] Observe that X can be seen as a set of Y i.i.d. Bernoulli RVs. Then, $G_X(z)=G_Y(G_B(z))$, where G denotes the probability generating function and B indicates a Bernoulli RV

Table 4.2: Total variation distance ($\times 10^{-2}$) of the Approx. Distribution

	$\hat{A}=4$				$\hat{A}=6$					$\hat{A}=8$				
ω	$\lvert\mathcal{N}(d)\rvert$	$H=2$	$H=3$	$H=5$	$\lvert\mathcal{N}(d)\rvert$	$H=2$	$H=4$	$H=6$	$H=8$	$\lvert\mathcal{N}(d)\rvert$	$H=2$	$H=4$	$H=7$	$H=9$
0.9		0.99	1.21	0.90		0.93	0.82	1.49	0.37		0.77	1.58	2.06	1.35
0.7		0.56	0.84	1.32		0.68	1.00	0.51	1.01		2.24	2.16	1.13	0.43
0.5	6	1.28	1.61	1.40	4	2.74	1.40	0.82	0.60	2	1.44	0.39	0.03	0.02
0.3		1.08	1.99	1.72		1.29	0.22	0.07	0.01		0.17	0.06	0.05	0.01
0.9		0.42	0.12	0.73		0.39	0.17	0.13	0.69		0.75	0.75	0.78	1.46
0.7		0.29	0.45	0.47		1.02	0.91	0.58	0.66		2.04	1.58	1.01	1.01
0.5	14	0.62	0.33	0.62	8	2.72	1.68	1.29	0.73	4	1.81	1.12	0.62	0.27
0.3		2.37	1.88	1.14		3.45	0.92	0.20	0.15		1.01	0.13	0.02	0.02
0.9		0.66	0.58	1.78		0.61	0.95	0.99	1.79		1.10	0.88	0.45	1.16
0.7		0.64	0.22	1.64		0.92	0.73	0.46	0.78		2.63	1.41	1.02	0.58
0.5	22	0.52	0.47	2.20	12	1.42	1.49	1.58	1.40	7	3.57	1.83	0.30	0.34
0.3		2.07	1.92	1.37		5.20	1.25	0.48	0.21		2.89	0.37	0.08	0.02

4.2.3 Delivery efficiency

The performance of RLNC depends on the ability of a node to fully or partially decode a generation, which in turn depends on the rank of the decoding matrix. We examine the usual approach in the literature, in which a source node transmits a new encoded packet each time a native packet is created and added in a generation. In the context of RLNC, each intermediate node, instead of forwarding a received encoded packet, creates a new one. As a result, each node will receive a number of encoded packets with a probability given by (4.2). The received encoded packets may increase the rank of the decoding matrix, depending on whether they are innovative or not. We assume that the delay from a source to a receiver is smaller than the time between the creation of two native packets, so that all the encoded packets, created after adding the $(k-1)$-th native packet, arrive before the ones created after adding the k-th. Then, the rank of the decoding matrix can be modeled as a stochastic process $Z = \{Z_k, k \in \mathbb{N}\}$, where the RV Z_k denotes the rank after a node d receives all the encoded packets created by the k-th native packet. Note that Z is memoryless because Z_k depends only on the total number of innovative packets received after $k-1$ native packets (i.e., Z_{k-1}). Therefore, Z is a discrete-time Markov chain and its state space is $[0, g] \in \mathbb{N}$.

In the following, we focus on analysing the best case performance of RLNC in order to illustrate its full potential for providing increased delivery efficiency. We

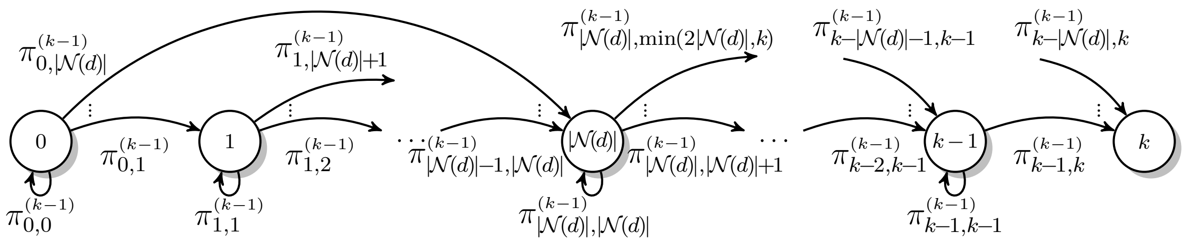


Figure 4.1: Proposed Markov chain at time $\big(k-1,k\big], k > |\mathcal{N}(d)|$.

discuss the case of non-optimal performance in Section 4.2.4. Suppose that, at time $k-1$, the rank of the decoding matrix of node d is i, i.e., $Z_{k-1}=i$, and that d receives $j-i \leq |\mathcal{N}(d)|$ encoded packets. If n_{in} denotes the number of the encoded packets that are also innovative, then $Z_k = i + n_{in}$. Note that $n_{in} \leq j-i$. Furthermore, $Z_k \leq k$ because at time k only k native packets have been added in the generation. This implies that $n_{in} \leq k-i$. Therefore, $n_{in} \leq \min\{j-i, k-i\}$. The best performance occurs when the rank of the matrix is maximal or, equivalently, n_{in} is maximal. When $j<k$, the best performance is when $n_{in}=j-i$, i.e., all the received encoded packets are also innovative. In this case, $Z_k = i+(j-i) = j$ and the transition probability from state i to state j is therefore equal to the probability of receiving $j-i$ encoded packets. However, when $j \geq k$, only $k-i$ out of the $j-i$ encoded packets are innovative because Z_k cannot exceed k. In this case, $Z_k = i+(k-i) = k$ and the transition probability from state i to state k is equal to the probability of receiving $k-i$ or more encoded packets. Summarizing, the transition probabilities in the interval $(k-1,k]$, $1 \leq k \leq g$ are:

$$\pi^{(k-1)}_{i,j} = \begin{cases} \mathrm{P}\{X=j-i\} & j-i \leq |\mathcal{N}(d)|, j<k, i<k \\ \sum\limits_{w=k-i}^{|\mathcal{N}(d)|} \mathrm{P}\{X=w\} & j-i \leq |\mathcal{N}(d)|, j=k, i<k \\ 0 & \text{otherwise} \end{cases} \tag{4.3}$$

For $k>g$, $\pi^{(k-1)}_{i,i}=1$ and $\pi^{(k-1)}_{i,j}=0, \forall j \neq i$ since after time $k=g$ no native packets are added in the generation. Note that the Markov chain is time-inhomogeneous. Fig. 4.1 illustrates the transition probabilities for the time interval $\big(k-1,k\big], k > |\mathcal{N}(d)|$. The initial distribution is $\mathrm{P}\{Z_0=0\}=1$ and $\mathrm{P}\{Z_0=i\} = 0, \forall i > 0$. Therefore:

$$\mathrm{P}\{Z_k = i\} = \sum_{w=0}^{g} p^{(k)}_{w,i} \mathrm{P}\{Z_0 = w\} = p^{(k)}_{0,i} \tag{4.4}$$

where $p^{(k)}_{0,i}$ is the element of table $\Pi^{(k)} = \boldsymbol{\pi}^{(0)}\boldsymbol{\pi}^{(1)} \cdots \boldsymbol{\pi}^{(k-1)}$ in the position $(0,i)$ and $\boldsymbol{\pi}$ are the transition matrices constructed using (4.3). Decoding is possible when $Z_k=k$

because k innovative packets are required for decoding the k native packets that exist in a generation at time $k^{\dagger}$. Furthermore, decoding of exactly k packets occurs when $Z_k = k$ but no further decoding is possible, i.e., $Z_w < w, \forall w > k$. As a result, the expected delivery rate is:

$$D_R = \frac{\sum_{k=1}^{g} \left[k\mathrm{P}\{Z_k = k\} \prod_{w=k+1}^{g} (1 - \mathrm{P}\{Z_w = w\}) \right]}{g} \tag{4.5}$$

where $\prod_{w=k+1}^{g}(1–\mathrm{P}\{Z_w=w\})$ is the probability that no decoding is possible for $w>k$.

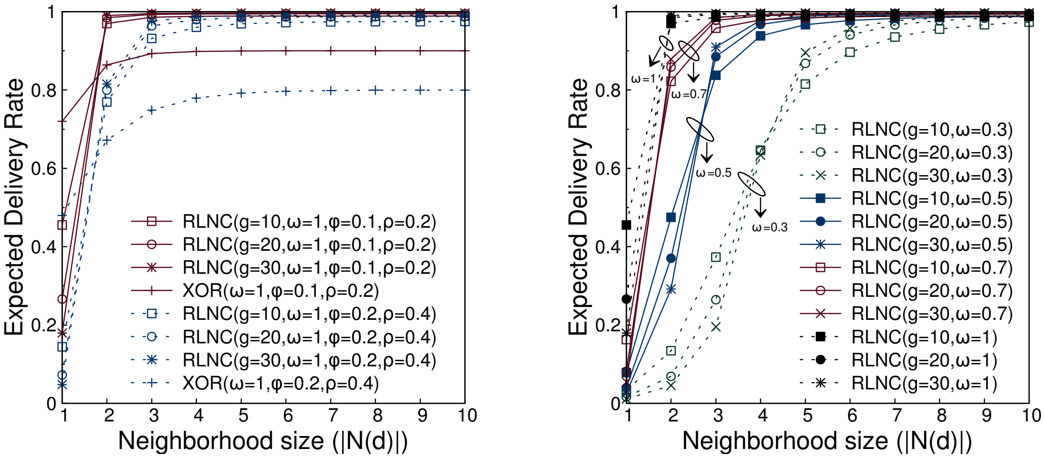
In XOR-based coding, packets are encoded under the requirement that each recipient node will decode at maximum one native packet. In other words, receiving an encoded packet is equivalent to receiving a copy of a native packet. Since receiving a single copy is enough, the expected delivery rate is $D_X = 1 - \mathrm{P}\{X = 0\}$. Note that, this is the best case performance as we do not take into account decoding failures.

4.2.4 Probabilistic broadcasting considered harmful

Fig. 4.2a illustrates the expected delivery rate for RLNC and XOR-based coding when combined with flooding ($\omega = 1$). More specifically, the delivery rate is plotted versus the node degree using different values of ϕ and ρ. RLNC exhibits high levels of resilience to transmission impairments and dominates XOR-based coding even when ϕ and ρ increase. A simple explanation is that using RLNC in broadcasting enables a node to exploit message redundancy (or equivalently path diversity) to recover not just a single packet but any packet from the generation. This is possible through the creation of new encoded packets in each intermediate node, which allows a node to receive a plethora of possibly useful packets. This also explains why RLNC fails when message redundancy is absent ($|\mathcal{N}(d)| = 1$). RLNC-based schemes should specially treat such cases. As a final note, the proposed Markov chain can be easily generalized to describe the cases that a node receives less than the maximum number of innovative packets per native one. Our results indicate that RLNC continues to dominate the best-case performance of XOR-based coding (for a wide range of ϕ and ρ values). The only exception is when a node receives, at maximum, only one innovative packet for each native, i.e., again when diversity is absent. Nevertheless, such a situation is highly unlikely.

The advantage of XOR-based schemes is the reduced cost, since coding is utilized

†We underestimate the decoding probability since decoding may be possible even if $Z_k < k$



(a) (b)

Figure 4.2: Analysis of RLNC's delivery efficiency: (a) comparison with XOR-based coding, (b) impact of probabilistic forwarding (ϕ=0.1,ρ=0.2).

towards reducing transmissions. Instead, RLNC-based schemes resort to probabilistic forwarding for reducing cost. Fig. 4.2b depicts the performance of RLNC when combined with probabilistic forwarding for different values of ω. The plotted results are in accordance with other reported simulation data [60]. Clearly, pruning transmissions significantly impairs RLNC's performance. This performance degradation has also been identified, implicitly [23] or explicitly [67], however the problem has been treated within the context of probabilistic broadcasting. We believe that the key factor for RLNC's performance degradation is the unsystematic way of pruning transmissions, which does not take into account information about connectivity. The strategy to prune transmissions based on heuristics that account for the node degree [23] is towards the correct direction. However, we feel that such heuristics should also take into account topology-related information of non-local scope. Some of this information is difficult to obtain and even if this was possible, it would require a complex analytical model to define the optimal ω. Therefore, *we opt for a more systematic and self-configuring pruning mechanism that takes into account the network topology*.

4.3 The synergy of RLNC and deterministic broadcasting

Following the previous observations, we adopt RLNC. Yet, contrary to the common approach, we implement it on top of a deterministic broadcast algorithm. We choose

Partial Dominant Pruning (PDP) [27] from the class of Dominant Pruning (DP) algorithms. DP algorithms distributively construct a CDS in order to broadcast messages. Our intuition is that the CDS will provide a topology-aware, self-configuring process for reducing transmissions. However, establishing this synergy, without damaging RLNC's coding efficiency, is not a trivial task. PDP's forwarding rules need to be redesigned so as to treat packets as members of a group, i.e., the generation. *Random Linear network coding over Dominant Pruning (RLDP)* incarnates the aforementioned concepts.

4.3.1 Basic concepts

In DP algorithms, a node v, with a message to broadcast, decides which of its neighbors should act as forwarders and informs them by piggybacking on the message the corresponding list, called the *forwarding set* $(fs(v))$. The process is then repeated by every forwarder until a *termination criterion* is met [27]. Forwarders should be elected so as to deliver the message to (or "cover" according to the set cover terminology) the set of nodes that lie exactly 2-hops away from v. This latter set is also called the universal set $\mathcal{U}(v)$, i.e., $\mathcal{U}(v)=\mathcal{N}(\mathcal{N}(v))-\mathcal{N}(v)$, where $\mathcal{N}(\mathcal{N}(v))$ is the set of nodes lying within 2-hops from v. The set of candidate forwarders $\mathcal{C}(v)$ consists of v's neighbors, i.e., $\mathcal{C}(v)=\mathcal{N}(v)$. Note that $\mathcal{U}(v)\subseteq\bigcup_{\forall u\in\mathcal{C}(v)}(\mathcal{N}(u)-\mathcal{N}(v))$ and that $\mathcal{C}(v)$ can be seen as a set of sets if each node $u\in\mathcal{C}(v)$ is replaced by $\mathcal{N}(u)-\mathcal{N}(v)$, thus the set cover problem. The problem is solved using the well-known greedy set cover (GSC) algorithm [92], however other approximation algorithms exist [25, 95]. PDP makes the observation that, when v receives a message from u, both $\mathcal{C}(v)$ and $\mathcal{U}(v)$ can be reduced by eliminating the nodes covered by u, i.e., $\mathcal{C}(v)=\mathcal{N}(v)-\mathcal{N}(u)$ and $\mathcal{U}(v)=\mathcal{N}(\mathcal{N}(v))-\mathcal{N}(v)-\mathcal{N}(u)-\mathcal{N}(\mathcal{N}(u)\cap\mathcal{N}(v))$.

Broadcasting is inherently coupled with some degree of message redundancy, i.e., a node receives multiple copies of a message, due to path diversity. RLNC takes advantage of this property to enhance error resilience. The idea is to allow forwarders to create new random linear combinations of the generation contents so that a node receives many different encoded packets. Proposed algorithms [23,60,61] use RLNC and build a generation using packets from the same source (intra-source coding) or from different sources (inter-source coding). In the context of RLNC-based broadcasting, inter-source coding operates on an end-to-end basis similar to the intra-source one, i.e., packets are linearly encoded at the source, re-encoded at intermediate nodes and only decoded at the destinations after traveling multiple hops. The only difference is in the composition of the generation. Therefore, this implementation of inter-source coding is also oriented towards resilience to transmission failures. The idea of mixing packets from different sources/sessions is not new and has been extensively used in the literature of network coding based unicast routing [107]. In this line of research inter-source coding is also used for enhancing multi-hop communication but it works on a hop-by-hop basis, i.e., requires decoding and re-encoding at each hop. This also applies to hybrid approaches that combine intra- and inter-source coding [108–112], where the former works on an end-to-end basis while the latter on a hop-by-hop one. This hop-by-hop coding approach is reasonable because mixing packets from different flows (i.e., traveling between different source/destination pairs) is meaningful at flow intersection points. Packets from different flows should then be decoupled in the next (probably non common) hop in order to be delivered to the different destinations. On the contrary, in broadcasting packets from different sources are destined to every node in the network, i.e., they share the same set of destinations. Therefore coding and decoding of packets can be performed at the communication end points (i.e., in an end-to-end fashion).

4.3.2 Coding rules

RLDP adopts *inter-source coding* in the light of empirical evidence which demonstrates that it increases the coding efficiency when combined with intra-source coding compared to the case that only intra-source coding is used [61]. However, in this case the problem of generation management is not trivial. In contrast to the usual approach, which is to operate inter-source and intra-source coding in parallel [23, 60, 61], in RLDP, each source can add only one packet in each generation. In other words, we adopt inter-source coding but do not allow intra-source coding. We call this strategy *strictly inter-source coding (SIS)*.

Our approach stems from the observation that, in the context of multi-source energy efficient broadcasting, intra-source coding may pose performance issues for poorly connected source-receiver pairs. To understand this recall that the performance of RLNC degrades under poor connectivity because of the limited message redundancy experienced by a receiver. In inter-source coding however, various sources enroll in a generation. While some sources may form poorly connected pairs with a given destination, it is possible that the opposite holds for some other. This latter set of sources can compensate for the limited redundancy provided by the former one, thus increasing the decoding probability. In other words, inter-source coding exploits what we call "spatial diversity", i.e., the fact that sources located in various parts of the network can provide different levels of message redundancy to a specific destination. Clearly, when the number of sources reduces, the coding gains of SIS deteriorate because it is more probable that a destination will experience poor connectivity with all the sources. In the case of intra-source coding, the only way to overcome the problem of a poorly connected source-destination pair is to employ "*temporal redundancy*", i.e., to allow each intermediate node to subsequently transmit more coded packets. Although this is an efficient approach for unicast scenarios, in the case of broadcasting it results in rapidly increasing the overall number of transmissions because a non trivial percentage of the nodes in the network act as forwarders. Besides the apparent impact on the energy cost, our observation is that this approach may also significantly increase the end-to-end delay. The reason is the elevated number of collisions that can delay the decoding of a generation. Consequently, we have chosen to rule out intra-source coding and focus on SIS. We validate the effectiveness of our approach in various settings (Section 4.5 and Section 4.6), including scenarios with limited number of sources. Of course, in the extreme case of a single source the only way to enjoy the benefits of coding is to use the intra-source approach. A node can use its decoding matrix to detect such cases and switch to a strictly intra-source operation. We do not examine this scenario since we focus on the multi-source case. As a final note, adopting SIS also allows us to simplify the generation management, i.e., make it easier for source nodes to collectively agree on the grouping of packets into generations. We discuss generation management in detail in Section 4.4.

Similar to every RLNC scheme, each node maintains a decoding matrix for each known generation. A generation is considered known if the node either created it or has received at least one encoded packet belonging to it. After creating a new native packet, the source node either starts a new generation or chooses from the set of known ones in order to add the packet (we discuss in detail this issue in Section 4.4). Then, it immediately creates and transmits a new encoded packet. The rationale of this strategy is twofold; first it aims to ensure that the new information carried by the native packet will be propagated through the network with minimum delay. Second, it facilitates partial decoding, which reduces end-to-end delay. To understand this, bear in mind that non zero rows of a decoding matrix correspond to innovative packets while non zero columns correspond to native packets. Consequently, the strategy of immediately transmitting a new encoded packet, increases the probability that the decoding matrix contains a full rank square submatrix, thus enabling partial decoding.

A node can attempt to perform partial decoding instead of waiting for the decoding matrix to become full rank. Deleting the decoding matrix and the packets of a generation is an important decision for managing storage. A frequently used practice is to employ feedback information that allows a receiver to indicate that it has successfully decode a generation, e.g., [67,72]. However, such approaches require an extremely large number of control messages in the context of broadcasting. This is because all nodes act as receivers and most of them are involved in the forwarding process. Proposed techniques for reducing control messages, e.g., using local scope advertisements [67,73], are not suitable for mobile networks. A more flexible approach is to allow a node to define a time threshold after which a generation is deleted. The threshold represents the maximum acceptable delay for receiving a packet and can be adjusted to also take into account each node's storage profile.

Besides being a receiver, a node may be required to act as an intermediate and forward an encoded packet after receiving an innovative one. In that case, the node forwards a new encoded packet. The latter can be created using the packets received up to that time without the need to decode the native ones. In the following, we delineate the conditions under which a node should act as a forwarder.

4.3.3 Forwarding rules

In order to achieve the synergy of RLNC and PDP, we need to enable the *propagation of generations through the CDS* formed by PDP. Note that, in DP algorithms, a node v reacts to the reception of a packet only if it has been selected as a forwarding node. Furthermore, recall that, in RLNC, only a subset of the encoded packets, the innovative ones, carry useful information about a generation. Therefore, the first intuitive approach is to adopt the following forwarding strategy:

Definition 4.1 (Innovative-based criterion). A forwarding node produces and transmits a new encoded packet iff it receives an innovative packet.

In the context of DP, the Innovative-based criterion is actually a termination criterion, i.e., the execution of the algorithm stops when a non innovative packet is received. This criterion is the analogous of the stopping conditions adopted by schemes that implement RLNC on top of probabilistic broadcasting [23, 61]. Given the Innovative-based criterion, we can prove the correctness of RLDP[3], i.e., that all network nodes can fully decode a generation in a lossless network. First, we prove that:

Lemma 4.1. *Every node v, which is not the source node of a native packet q, receives* at least *one innovative packet after q is added in a generation.*

Proof. The source node s, after adding q to a generation i, defines a forwarding set $fs(s)=\{f_1, f_2, \ldots\}$ and transmits a new encoded packet $e_{s,i}$. Every node $v \in \mathcal{N}(s)$ will receive this packet, which is innovative since it "contains" q. Furthermore, the solution of the set cover problem guarantees that, given a node $u \in \mathcal{N}(\mathcal{N}(s))$, there is at least one forwarder $f \in fs(s)$ that covers u. Since $f \in \mathcal{N}(s)$, it will receive the innovative packet $e_{s,i}$ and will transmit a new encoded packet $e_{f,i}$. As a result, each node $u \in \mathcal{N}(\mathcal{N}(s))$ will receive at least one encoded packet $e_{f,i}$. The first of these packets is clearly innovative since it "contains" q. The same reasoning can be used in subsequent hops to include all network nodes. □

Moreover, we can prove that:

Lemma 4.2. *Every node v, which is not the source node of a native packet q, receives* exactly *one innovative packet after q is added in a generation.*

Proof. According to Lemma 4.1, if g native packets are added in generation i, then each node v will receive $g' \geq g$ innovative packets. It suffices to show that $g'=g$. Note that the row rank of $\mathbb{G}_{v,i}$ (which equals g') cannot exceed the column rank (which equals g), i.e., $g'>g$ is not possible. □

We use this lemma to prove that:

Theorem 4.1 (Correctness of RLDP). *Every node can decode a generation in a lossless network.*

Proof. Lemma 4.2 secures that, any non source node v will receive exactly g innovative packets for a generation of size g, thus $\mathbb{G}_{v,i}$ has a full rank. Furthermore, each source

[3] We assume that the probability of producing linearly dependent encoded packets is negligible [100]. This assumption is common in the related literature.

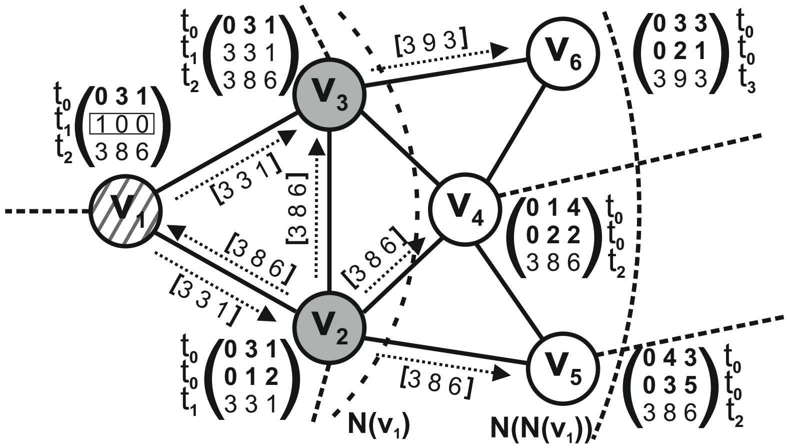


Figure 4.3: Example of broadcasting with RLDP (Single-Innovative criterion).

s will receive exactly g–1 innovative packets, one for each native packet added by the other sources. This is sufficient since s only needs to decode g–1 packets. □

According to Lemma 4.2, the Innovative-based criterion is equivalent to the strategy of forwarding one encoded packet for each native one added in a generation. However, in the presence of transmission errors, if a native packet is added in a generation, a node v will receive more than one innovative packet. This happens when the rank of its decoding matrix is lower than the rank of the decoding matrices of its neighbors. Using the Innovative-based criterion in such cases will result in v transmitting more than one encoded packet for each native one. To explain the situation, let us examine the example in Fig. 4.3. In this example, we monitor the decoding matrices ($g = 3$) in a part of a network. At some point in time (t_0), the generation contains already two native packets (added by some other nodes in the network). As a result, each node has received at most two innovative packets and populated its matrix accordingly (entries marked with t_0). Note that, due to transmission errors, v_1 and v_3 have received only one innovative packet. At some point, v_1 acts as a source, adds a packet in the generation and after selecting the forwarding set (in this case $fs(v_1)$=$\{v_2, v_3\}$), transmits an encoded packet (the transmitted innovative packets are illustrated with dashed lines along with the corresponding encoding vectors). Both v_2 and v_3 receive an innovative packet and update their matrices (entries marked with t_1). Then, v_2 (a forwarder that received an innovative packet) transmits an encoded packet which is received by v_3, v_4 and v_5 (entries marked with t_2). Note that v_3 receives two innovative packets (one from v_1 and one from v_2) and, according to the Innovative-based criterion, should transmit two encoded packets. We make the observation that, in the context of Dominant Pruning, *not all innovative packets need to result in the transmission of a new encoded packet*. In fact, we introduce the following policy:

Definition 4.2 (Single-innovative criterion). A forwarding node produces and transmits a new encoded packet only for the first innovative packet that is received as a result of the addition of a native packet in a generation.

The rationale of this policy is clear and, in part, is expressed by Lemma 4.2 and Theorem 4.1; in the absence of transmission errors only one innovative packet per native is adequate while, in the presence of transmission errors, *a node should only rely on the path diversity provided by the network to recover from transmission errors*. To further explain, let us go back to the example of Fig. 4.3. When the Single-innovative criterion is used, v_3 receives two innovative packets and decodes the generation. However, v_3 will transmit only one new encoded packet. Note that this new packet is enough for v_6 to decode the generation (entry marked with t_3). Furthermore, observe that v_6 actually receives two encoded packets (the second one is from v_4 and is not illustrated since it is not innovative). If the rank of v_6's decoding matrix was initially one, both of the received packets would be innovative. Therefore, v_6 could take advantage of path diversity and decode the generation. Clearly, there is still the probability that a node will not be able to decode a generation. In general, this probability increases for nodes with low connectivity.

One solution to eliminate failures would be to allow a node to relax the Single-innovative criterion based on the connectivity of its neighbors or even based on feedback information. We further examine this solution in Section 4.6. For now, we refrain from investigating the impact of such methods, as well as the related cost, since our primary objective is to illustrate that using deterministic broadcasting, even without such methods, results in less decoding failures compared to a probabilistic scheme.

An important issue is how to implement the Single-innovative criterion. To do so, we need to provide some kind of association between a native packet q and the innovative ones produced after node s adds q in a generation i. Since in RLDP a node adds only one native packet into a generation, this task can be tackled by using the value pair $\langle s, i \rangle$, i.e., the source address and the generation id, which is contained in a packet's header. Another requirement is to allow a forwarding node

Algorithm 4.1 Pseudocode of RLDP's forwarding procedure.

RLDP(prev_node u**,cur_node** v**,packet** p**,generation_id** gid**)**

if (!Innovative(p)) **then**
    DropPacket(p)
end if
UpdateDecodingMatrix(p)
if ($v \notin p.forwarders$ || !Single-innovative($p.src, gid$) **then**
    DropPacket(p)
end if
$newp$=RandomLinearCoding(gid)
$fwset$=GSC($N(v)$, $N(N(v))$, u)
$newp$.set($fwset$)
transmit $newp$

v to track whether an innovative packet with the same value pair $\langle s, i \rangle$ has already been received. The most efficient way is to use direct addressing [92] due to the fast dictionary operations. The space complexity of such an approach ($\mathcal{O}(g)$ for a generation of size g) is reasonable since the generation size is usually kept low in order to reduce the decoding cost. The forwarding procedure of RLDP is illustrated in algorithm 4.1.

4.4 Distributed generation management

In order to practically implement RLNC, packets should be grouped into generations. This task, known as *generation management*, is easier to handle when intra-source coding is used since the required decisions involve a single node and thus are made locally. In this case, the main challenge is to select the generation size g so that it maximizes a chosen performance feature (e.g., [113]). However, generation management becomes more complicated in the case of inter-source coding since the sources should agree on a common grouping of packets in a distributed fashion. In the following, we focus on the challenges that arise from this requirement:

1) Decide which generation to choose for adding a native packet and when to start a new generation: Choosing a generation is the first important decision to make because it affects the overall performance. Observe that, when inter-source coding is used, a node v may be unaware of the existence of a generation or have incomplete view of the number of packets added in it. The reason is that when another node u adds a packet to an existing generation or starts a new one, node v becomes aware of this after receiving an encoded packet produced from this generation. Therefore, the challenge is to ensure that the number of packets added in each generation will be close to the predefined size g. Let $\mathcal{GS}_v$ denote the set of generations which are known to v (i.e., v has received at least one encoded packet from or created the generation) and their size, according to v's view, has not exceeded the size g. The common approach is that a source s will add a new packet to a randomly chosen generation from $\mathcal{GS}_s$ [23, 60]. Another approach is to choose from a subset of $\mathcal{GS}_s$ which contains generations initiated from nodes that lie certain hops away from s [23, 61]. A new generation is started if $\mathcal{GS}_s = \emptyset$ [23, 60, 61] or when the chosen generation already contains a packet from s [60]. All the proposed strategies aim at reaching the predefined generation size, in order to increase performance [60]. However, under transmission errors, a large generation size increases the decoding delay. The reason is that it takes longer to collect the number of encoded packets that is required for decoding. Following this observation, we opt for reduced delay. Therefore, in RLDP, a source s adds a new packet to the most recently seen generation, if this belongs to $\mathcal{GS}_s$ (algorithm 4.2 presents the process for selecting a generation). If no such generation is found or the selected generation already contains a packet from s (strictly inter-source coding), then a new generation is created. Note that, the size of the produced generations will not necessarily be close to g. However, we believe that this will not have a significant impact on the decoding efficiency. This intuition is based on reported empirical data [23, 60], also confirmed by the analysis in Section 4.2, which indicate that a relatively small generation size is enough for providing the coding benefits. We confirm our intuition through simulation in Section 4.5.

2) Provide an addressing scheme for packets within a generation: Another problem, although rarely discussed in the literature, is to uniquely identify packets within a generation. To understand this requirement, recall that each encoded packet carries an encoding vector, i.e., the coefficients $\langle c_1, \ldots, c_g \rangle$ used to mix the native packets. In order for decoding to be possible, it is necessary that all nodes will be able to agree on and use the same mapping between native packets and coefficients. This is a challenging task in a distributed environment. A practical solution is to provide a unique id for each native packet, so as to enable sorting based on this id, and associate it with the corresponding coefficient. The simplest way to accomplish this is by using the pair $\langle node\_id, seq\_num \rangle$, where seq_num is a sequence number generated locally at the source and $node\_id$ is the source address [61]. The use of seq_num enables two packets from the same source to coexist in a generation. RLDP takes a simpler approach. Since strictly inter-source coding is used, there is no way that two native packets from the same source will reside in the same generation. Therefore, only $node\_id$ can be used for uniquely identifying a packet in the generation. Our strategy, besides using a smaller identifier for packets, does not involve any overhead for managing sequence numbers.

3) Provide an addressing scheme for generations: The next important task is to uniquely identify generations so that each node can decide to which generation an encoded packet belongs to. The problem arises when a source, due to the incomplete knowledge of existing generations, uses an id to start a new generation without knowing that this has already been used by another source. Consequently, a node may receive two encoded packets with the same generation id but constructed using different native packets. The downside is that it is possible to destroy the one-to-one mapping between the native packet ids and the coding coefficients. To tackle the problem, the usual approach is that a node will randomly choose the generation id [23, 60, 61]. Choosing from a sufficient large space minimizes the probability that two different nodes will choose the same generation id. In RLDP however, two sources can use the same generation id without destroying the aforementioned mapping. This is because, in any case, a generation will contain, at maximum, only one native packet per source and this will be uniquely identified by $node\_id$. Therefore, when a source starts a new generation, increases by one the most recently seen generation id and uses this as the new id (algorithm 4.2). This strategy allows sources to coordinate their views about the generations in use, thus enabling them to effectively populate the generations with packets. Eliminating the random selection of generation ids allows the use of a smaller address space.

Algorithm 4.2 Pseudocode for selecting a generation in RLDP.

GetGeneration(set $\mathcal{GS}_s$,generation_id $last\_seen$**)**

if ($last\_seen \in \mathcal{GS}_s$) **then**
  if (AlreadyUsed($last\_seen$) == TRUE) **then**
    $last\_seen = last\_seen + 1$ //create a new generation
  end if
else
  $last\_seen = last\_seen + 1$ //create a new generation
end if
return $last\_seen$

4.5 Evaluation of RLDP

To evaluate RLDP's performance, we compare it with two algorithms. The first one, proposed in [23], is the most representative of RLNC-based algorithms. In the following, we will use the term RLNC to refer to this algorithm. The second algorithm, CodeB [21], utilizes XOR-based coding. Regarding RLNC, we use two variants, namely RLNC^D and RLNC^G. The first, uses the distributed generation management described in [23]. In the second, we assume that each node has global coding information, i.e., perfect knowledge of the coding status of other nodes. This scheme achieves the optimal allocation of packets across generations. Although it is unrealistic, we use it to illustrate the performance bounds of RLNC. Furthermore, RLNC employs the forwarding heuristic described in [23, Algorithm 6B] with k=2. We chose this setting after extensive experimentation which showed that it yields the best performance in our experiments, i.e., it results in the best possible trade-off between delivery efficiency and the number of forwards.

Set up and methodology: All investigated algorithms are implemented in the ns2 simulator [96], using the CMU extension. Furthermore, RLNC and RLDP were implemented based on the network coding ns2 module [114]. We present the average values over 20 independent simulation runs, each with a duration of 900 seconds. The confidence level, for all reported confidence intervals, is 95%.

Network model: The default number of nodes is 100, the propagation model is the TwoRay ground with a transmission range of 250m and the nominal bit rate is 2Mbps. The nodes move in a square area according to the Random Waypoint (RW) model [97]. To avoid transient artifacts in nodes' movement, we use the perfect simulation algorithm [99]. We examine two network densities; "Dense" and "Sparse".

Similar to [21], in the “Dense” topology, the average neighborhood size is 30 while in the “Sparse” topology it is 15. Note that, we could not use a lower density in the “Sparse” scenario since, in such a case, frequent partitions would occur. Simulations showed that in the ‘Sparse” scenario, there exist many nodes (those moving near the boundaries) that experience very low connectivity. All algorithms collect neighborhood information by periodically exchanging hello messages with an interval of 1 second.

Network traffic: Traffic is generated by broadcast sessions, each stemming from a different source node and starting at a random time. The size of each message is set to 256 bytes. Furthermore, both the number of sources and the maximum generation size are fixed to 30. We chose the generation size after extensive experimentation, which showed that using a larger size does not improve performance but rather increases the related costs. We used a GF of size 2^8.

Fig. 4.4a and 4.4b depict the cumulative packet delivery ratio (PDR) versus the end-to-end delay, i.e., the cumulative fraction of native packets received by a node within a delay limit, for “Sparse” and “Dense” networks. We only consider decodable packets for calculating PDR. Moreover, for the calculation of a packet’s end-to-end delay we use the time instant that decoding of this packet becomes feasible. However, we do not consider the decoding delay because it depends on the implementation specifics of each scheme, thus making it impossible for a fair comparison. We choose the aforementioned presentation style in order to capture both the delivery efficiency and the timeliness of each algorithm. The results provide a confirmation of the effectiveness of random linear network coding. Both RLDP and RLNC^G outperform CodeB. The main reason is that XOR-based coding schemes introduce delay in order to detect coding opportunities. As expected, in the “Sparse” topology, the performance of all schemes degrades. For CodeB, a low density topology reduces the coding opportunities. As a result, more transmissions occur (compare Fig. 4.4c and 4.4d) and increase the probability of collisions. In the case of random linear coding, the witnessed degradation is in accordance to the analysis in Section 4.2 because in low density topologies the average neighborhood size is smaller. Nonetheless, RLDP outperforms both RLNC^D and RLNC^G, which uses global knowledge. This justifies our approach to combine random linear coding with deterministic broadcasting. Note that, in sparse topologies, RLNC^D fails to keep up with other schemes. This highlights the importance of distributed generation management. Also, observe that, RLDP’s generation management

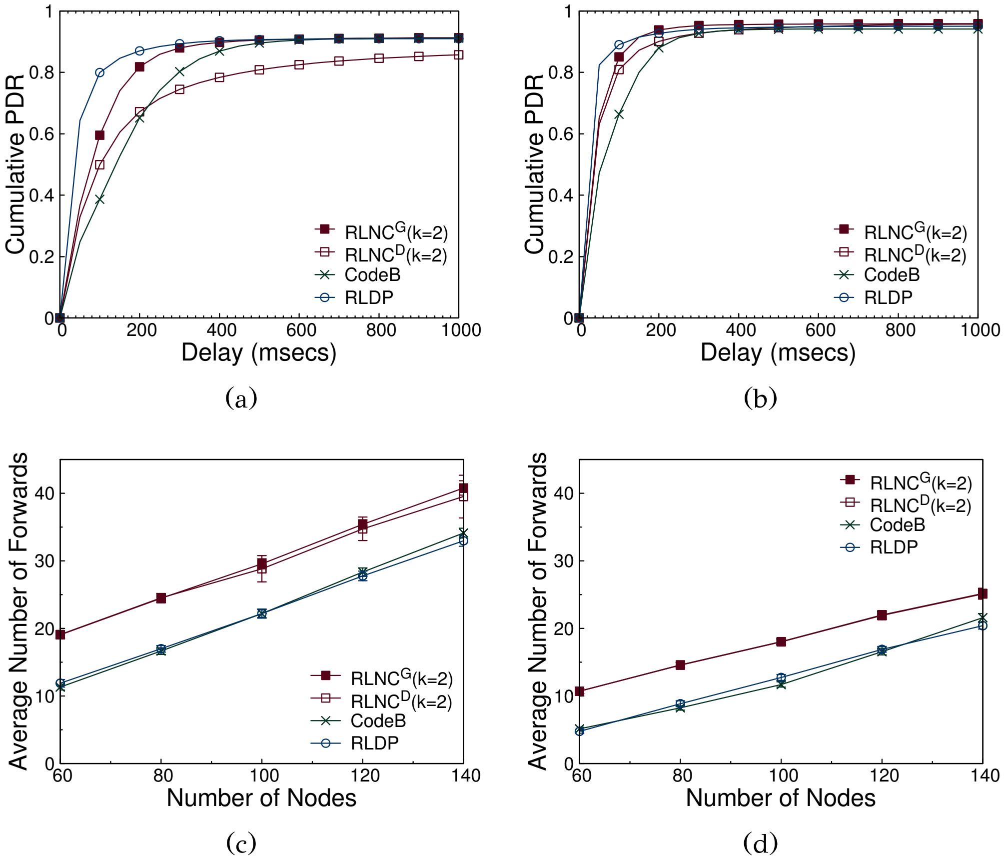
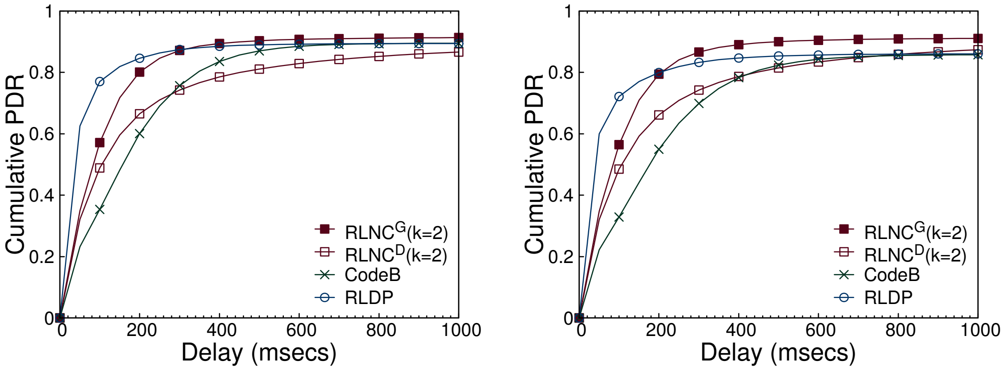


Figure 4.4: Performance for different network densities ($\lambda = 1$ pkt/sec/source, max speed:1 m/sec): (a) Cumulative PDR vs delay ("Sparse", $N = 100$) (b) Cumulative PDR vs delay ("Dense", $N{=}100$) (c) Avg. number of forwards vs number of nodes ("Sparse") (d) Avg. number of forwards vs number of nodes ("Dense").

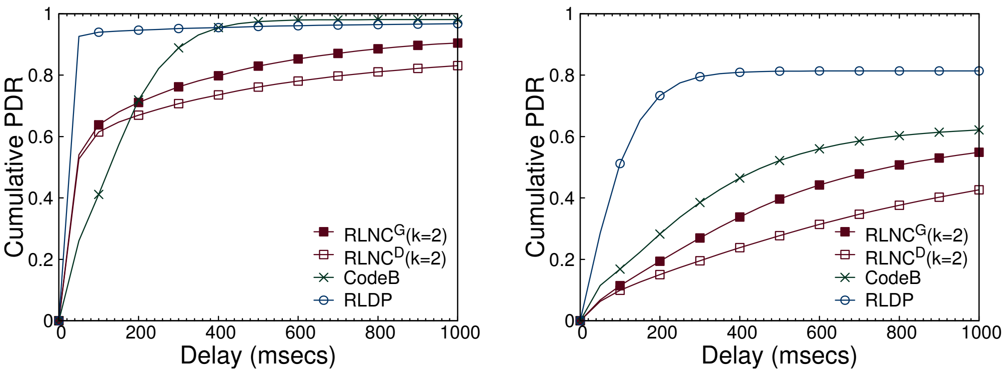
does not compromise the coding gains. We tested networks of various sizes (from 60 to 140 nodes) and found qualitatively similar results. Fig. 4.4c and 4.4d illustrate the average number of forwards versus the network size for "Sparse" and "Dense" networks. The results confirm the intuition that the CDS, used by RLDP to forward messages, provides an efficient pruning process. More specifically, RLDP manages a reduction from 17% to 38% in "Sparse" and from 19% to 56% in "Dense" networks, compared to RLNC variants. Interestingly, RLDP performs similar to CodeB, despite the fact that the latter uses coding for reducing transmissions.

In the following experiments, we only examine the delivery efficiency since we observed similar findings as far as the number of forwards is concerned. Furthermore, we focus on the more challenging scenario of "Sparse" networks. Fig. 4.5 presents the delivery efficiency under different levels of mobility. Clearly, increased mobility



(a) (b)

Figure 4.5: Cumulative PDR vs delay (λ=1 pkt/sec/source, “Sparse” topology). Node speed: (a) $2-10$ m/sec (b) $10-20$ m/sec.



(a) (b)

Figure 4.6: Cumulative PDR vs delay (max speed:1 m/sec, “Sparse” topology): (a) $\lambda=0.1$ (b) $\lambda=2$ pkts/sec/source.

levels impacts the performance of RLDP and CodeB. The reason is that both schemes use deterministic broadcasting, which is affected by topology variations. Moreover, mobility also increases the decoding failures in CodeB since successful decoding depends on the accuracy of information about the neighbors’ coding status. On the contrary, RLDP minimizes the impact of mobility on the deterministic broadcasting algorithm due to the use of random linear coding. Both RLNC^D and RLNC^G are virtually unaffected by mobility. This is attributed to the higher message redundancy produced by the probabilistic forwarding scheme. Nevertheless, message redundancy results in a significantly increased cost (more than 30% compared to RLDP) in terms of transmissions. In any case, observe that only the unrealistic RLNC^G outperforms RLDP when mobility is very high.

In Fig. 4.6, we evaluate the algorithms for different levels of traffic load. Under low traffic (Fig. 4.6a), RLDP outperforms all algorithms. CodeB needs to wait an increased amount of time in order to find coding opportunities since fewer packets coincide in the network. Both RLNC variants suffer from increased delay as they need more time to fill the generations. On the other hand, RLDP outperforms all schemes because its generation management is oriented towards reducing delay. The tradeoff is a reduced number of packets allocated to each generation (refer to Section 4.4). However, this does not impair the delivery efficiency. When congestion levels increase (Fig. 4.6b), the performance of all algorithms degrades. However, RLDP exhibits a remarkable resilience due to the combination of deterministic broadcasting and random linear coding. The former reduces the levels of congestion and thus decreases the probability of collisions. The latter uses path diversity to enhance delivery efficiency. Both mechanisms are equally important. CodeB and RLNC variants fail because they use only one of them.

4.6 Coping with poorly connected nodes

The analysis in Section 4.2 has demonstrated that RLNC's performance deteriorates in poorly connected nodes, i.e., nodes with a small number of neighbors (or equivalently limited message redundancy), even in the absence of probabilistic forwarding (refer to Fig. 4.2a where $\omega = 1$). The phenomenon intensifies as the probability of transmission failures ρ increases. Therefore, it becomes evident that *we should enhance message redundancy in a topology-aware fashion*, i.e., increase the number of encoded packets received by poorly connected nodes. Before implementing such a strategy, we make two important observations. The first is that, in RLDP, the message redundancy experienced by a node v depends on the number of forwarders that are located in $\mathcal{N}(v)$ (let $|\mathcal{N}^{\mathrm{F}}(v)|$ denote this number). This is in contrast to gossip-based forwarding, where the message redundancy depends on $|\mathcal{N}(v)|$. Consequently, a node v for which $|\mathcal{N}^{\mathrm{F}}(v)|$ is small is considered as poorly connected. The second important observation is that only forwarding nodes can enhance message redundancy because they are the only ones that transmit. Thus, any action for enhancing message redundancy should be taken by forwarders.

To tackle the aforementioned problem, the first important question is whether a forwarding node should try to enhance the message redundancy experienced by its neighbors or not. Our approach is that this should be done by a forwarding node that is a:

Definition 4.3 (Border Forwarder). A forwarding node that is the only one to cover one or more network nodes.

Equivalently, a forwarding node f_i is a border forwarder when there exists at least one neighbor v for which f_i is the only forwarder in $\mathcal{N}(v)$, i.e., $|\mathcal{N}^{\mathrm{F}}(v)| = 1$. More formally, $f_i \in fs(u)$, where $fs(u)$ is the forwarding set constructed by node u, is a border node iff:

$$\exists\, v \in (\mathcal{N}(f_i) - \mathcal{N}(u)) : v \notin \mathcal{N}(f_j), \forall f_j \neq f_i \in fs(u) \tag{4.6}$$

The rationale of using border forwarders is straightforward; a border forwarder should take action because it is responsible for delivering the message to at least one poorly connected node v because $|\mathcal{N}^{\mathrm{F}}(v)| = 1$. The advantage of this approach is that, at the same time, there is a high probability that other nodes, located in the neighborhood of the border forwarder, are also experiencing relatively poor connectivity and could benefit from its actions. Let us examine the example in Fig. 4.7a. Node v_6 is only covered by v_3, therefore v_3 is a border forwarder. At the same time, node v_4, although covered by two forwarders (v_3 and v_2), is relatively poorly connected, i.e., $|\mathcal{N}^{\mathrm{F}}(v_4)| = 2$. Consequently, both v_6 and v_4 could benefit if v_3 decides to act for enhancing message redundancy.

In order for a forwarding node f_i to identify itself as a border forwarder, it suffices to check whether there exists a node $v \in (\mathcal{N}(f_i) - \mathcal{N}(u))$ such that:

$$\forall f_j \neq f_i \in fs(u), f_j \notin \mathcal{N}(v) \tag{4.7}$$

Note that all required information for performing this test, i.e., $fs(u)$, $\mathcal{N}(u)$ and $\mathcal{N}(v), \forall v \neq u \in \mathcal{N}(f_i)$ is available to f_i. However, this test is based on local scope information, therefore f_i may falsely identify itself as a border forwarder.

To illustrate this, let us examine the example in Fig. 4.7b. Again, v_3 identifies itself as a border forwarder although v_6 is now covered by two forwarders (v_3 and v_4). This happens because v_3 is not aware of the fact that v_5, at a later time, chooses v_4 as a forwarder. Nonetheless, we do not wish to eliminate the occurrence of such events. The reason is that, although a false decision does not indicate a node covered by a single

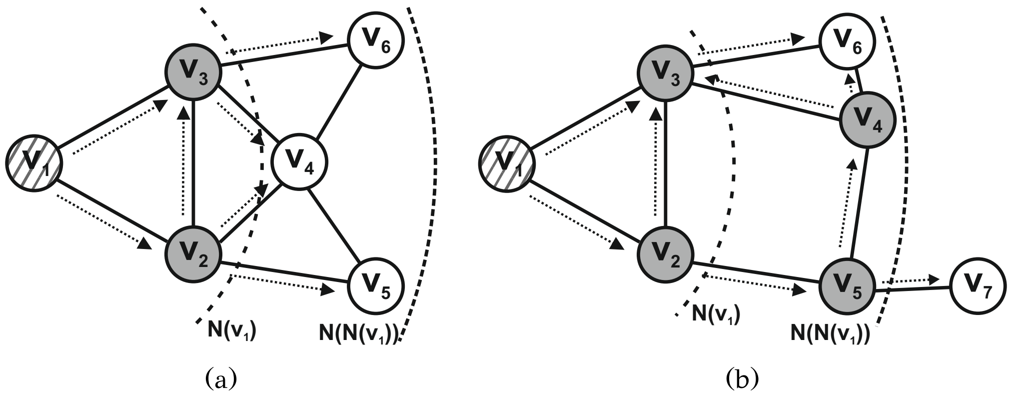


Figure 4.7: Example of a node (v_3) a) correctly and b) falsely identifying itself as a border forwarder.

forwarder, it is highly correlated with the identification of poorly connected nodes, which are also in need of enhanced message redundancy. For example, although v_6 is not anymore covered only by v_3, it is still a poorly connected node.

After deciding which forwarding node should act, the next important decision is what policy it should implement to increase the message redundancy. A first simple approach would be to transmit multiple encoded packets each time the Single-Innovative criterion is activated, i.e., when the first innovative packet is received as a result of the addition of a native packet in a generation. The main drawback of this approach is that it is not straightforward how to derive the appropriate number of encoded packets. Therefore, we take a more elaborate approach. We let border forwarders to relax the Single-Innovative criterion and instead implement the Innovative-based one. More specifically, we introduce the following termination criterion:

Definition 4.4 (Hybrid-Innovative criterion). A forwarding node implements the Innovative-based termination criterion if it is a border forwarder and the Single-Innovative criterion in any other case.

There are two advantages in this approach. The first is that it is topology-aware due to the use of border forwarders. The second is that it can adapt to network conditions. When the probability of transmission failures is small most packets are delivered and the rank of a node's decoding matrix is close to the maximum. Thus, the number of innovative packets decreases and transmissions are suppressed. On the other hand, when loss rate is high, the rank of many nodes falls behind the maximum possible rank, therefore more packets are innovative. As a result, this termination criterion will result in more packets being sent to poorly connected nodes.

4.6.1 Evaluation of RLDP-HI

To evaluate the new termination criterion and compare its performance to the other algorithms, we experiment by introducing transmission failures in the channel. More specifically, we use the error model in ns2 [96]. The model defines a loss rate l as the result of channel impairments. When a node transmits a message, each of its neighbors receives the message with a probability $1 - l$. We tested the algorithms for values of l from 0 to as high as 0.4. Note that l only captures the packet losses due to channel impairments while ρ refers to all packet losses, including those owned to collisions or even stale neighborhood information. Therefore, $\rho \geq l$ and $l = 0$ does not imply that packets losses do not occur. In the experiment, we vary l because the packet losses due to collisions and stale neighborhood information depend on traffic levels and node mobility, respectively, thus it is impossible to quantify ρ.

Fig. 4.8a-4.8c illustrate the cumulative PDR vs the end-to-end delay for all schemes and for various values of l. We use RLDP-HI to denote RLDP with the Hybrid-Innovative termination criterion. The results confirm our approach. Introducing the Hybrid-Innovative criterion to RLDP improves the performance even when $l = 0$ (recall that even in this case $\rho \geq 0$, so there is room for improvement). RLDP-HI presents a remarkable resilience to transmission failures and it is outperformed only by RLNC^G and only when l is as high as 0.4. The latter is a reasonable result since RLNC^G features the unrealistic scenario of perfect knowledge about generations. On the other hand, all fully distributed schemes experience a higher performance degradation because the increased loss rate has an impact on the accuracy of a node's local information about generations. In any case, RLDP-HI outperforms all other fully distributed schemes, including CodeB. Although CodeB does not use packet generations, its performance declines for different reasons. Recall that, in CodeB, a node v maintains information about the packets received by another node u (previously denoted as B_u^v) in order to identify coding opportunities and secure that successful decoding is possible. Nonetheless, packet losses significantly invalidate the information in B_u^v, thus leading to decoding failures.

As expected, introducing the Hybrid-Innovative termination criterion results in

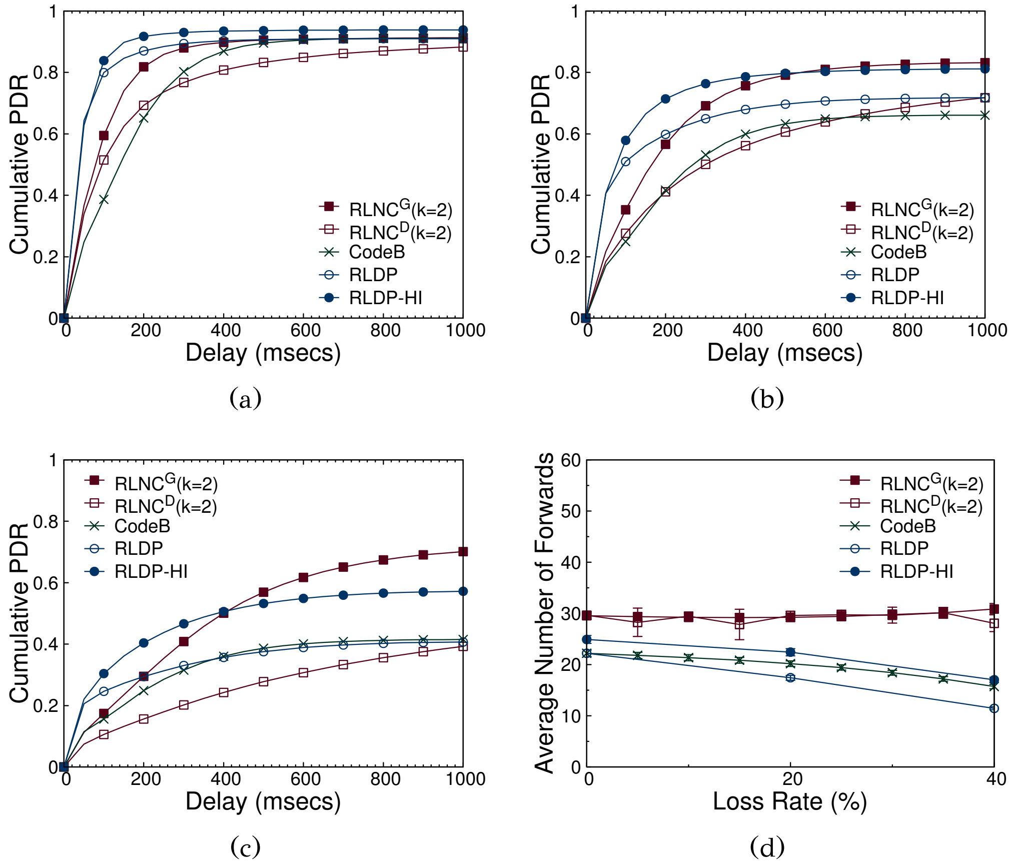


Figure 4.8: Performance for different loss rates ("Sparse", N=100, λ=1 pkt/sec/source, max speed:1 m/sec): (a) Cumulative PDR vs delay (Loss rate 0%) (b) Cumulative PDR vs delay (Loss rate 20%) (c) Cumulative PDR vs delay (Loss rate 40%) (d) Avg. number of forwards vs loss rate.

more transmissions (Fig. 4.8d). Nevertheless, the increase is minimal, proving the efficiency of the criterion. This result is more impressive if we bear in mind that the number of transmissions and the delivery rate are correlated; dropping a packet aborts future transmissions, thus creating a bias in favour of the other schemes. Indeed, RLDP-HI performs close to CodeB, in terms of transmissions, but at the same time improves the delivery rate by $\sim$3% when $l = 0$ and $\sim$17% when $l = 0.4$. Besides being efficient due to its topology-awareness, the Hybrid-Innovative criterion also presents a remarkable adaptability to the loss rate, i.e., it performs equally well, in terms of transmissions, for small and high loss rates. This is a confirmation of the rationale that led us to the introduction of the Hybrid-Innovative termination criterion. Moreover, this performance characteristic renders RLDP-HI as the best solution regardless of the loss rate.

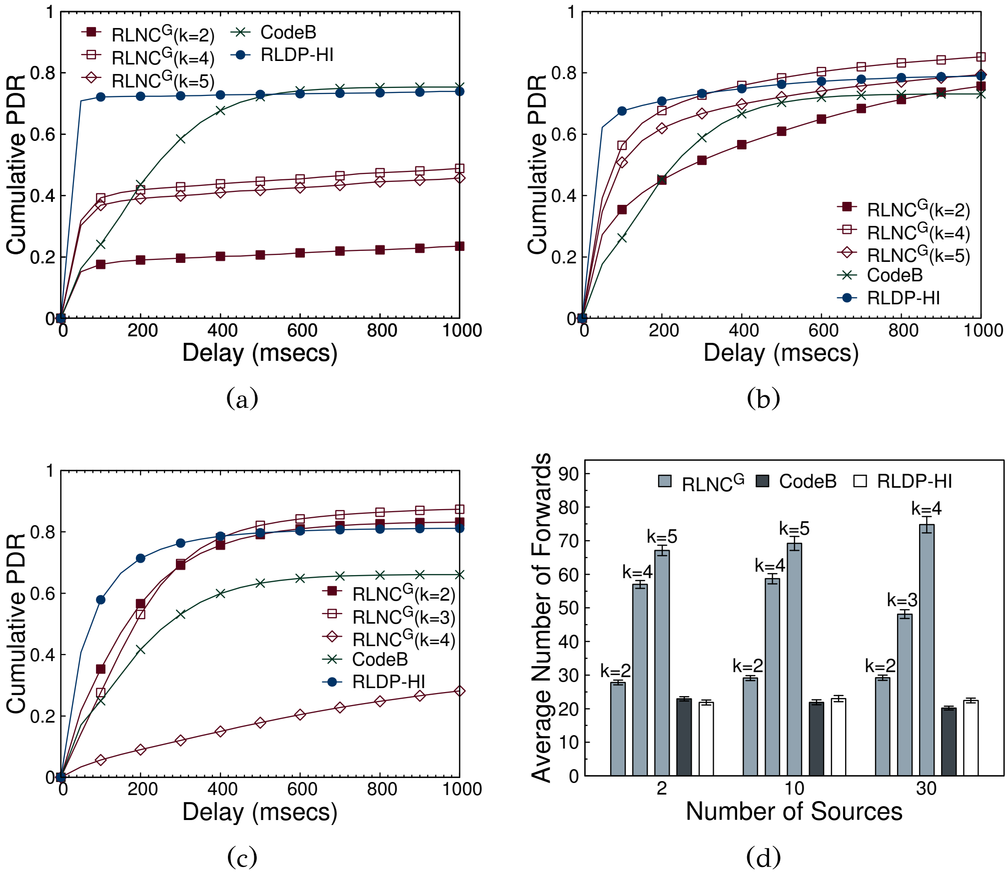


Figure 4.9: Performance for different source count(λ=1 pkt/sec/source, max speed:1 m/sec, N=100, Loss Rate 20%, "Sparse" topology): (a) Cumulative PDR vs delay (2 sources) , (b) Cumulative PDR vs delay (10 sources), (c) Cumulative PDR vs delay (30 sources) and (d) Avg. number of forwards.

Next, we tested all schemes by changing the number of sources that are present in the network. As discussed in Section 4.3.2, the source count affects the performance of RLDP. Obviously, the impact is more severe in the presence of losses since message redundancy deteriorates. Therefore, we have chosen to present the results in the presence of channel loss rate ($l = 0.2$) and found analogous results for values of $l \in [0, 0.4]$. Fig. 4.9 presents the performance of RLDP-HI, RLNC^G and CodeB when there exist 2, 10 or 30 sources in the network. We also present the performance of RLNC^G for various values of k. As expected, the performance of RLDP-HI, in terms of the cumulative PDR vs the end-to-end delay (Fig. 4.9a-4.9c), degrades as the number of sources decreases. This is reasonable because less packets are included in a generation and therefore there are less opportunities to exploit the "spatial diversity" that we discussed about in Section 4.3.2. However, the degradation is limited since less traffic results in less packet collisions. Interestingly enough, a similar, but more severe, performance degradation is witnessed for RLNC^G regardless of the value of k. Although part of this degradation (mostly in the case of 2 and 10 broadcasting sources) can be attributed to the time required for filling a generation, the major reason is related to intra-source coding and the use of temporal redundancy. Recall that RLNC^G implements both inter- and intra-source coding. When the number of sources decreases, the coding process resembles a pure intra-source approach since more packets from the same source are included in a generation. As a result, the role of "temporal redundancy" (i.e., the ability of an intermediate node to transmit more encoded packets), which is necessary for coping with losses in this case, becomes more critical in decoding a generation. However, in the context of broadcasting, the "temporal redundancy" comes at the cost of delay, thus the performance degradation. To explain this, observe that in broadcasting many forwarders may find themselves within each other's transmitting range. Consequently, there is an increased probability of collisions which can delay the decoding of a generation because a destination may need to wait for subsequent transmissions in order to receive the required amount of encoded packets. Reasonably, increasing the "temporal redundancy" (through k) improves the performance of RLNC^G but there are two downsides. The first is that there is a limit for k after which no improvement is possible and the performance actually degrades (compare for example $k = 4$ and $k = 5$ in Fig. 4.9a). The reason is that the impact of collision-related failures increases to a level that invalidates the benefits of redundancy. The second and more critical disadvantage is that any performance improvement comes at the expense of a surge in cost (Fig. 4.9d). In the case of CodeB, the performance actually degrades when the number of sources increases. At first, this seems surprising since packets from more sources provide more coding opportunities. However, the higher traffic load increases the collision-related transmission failures. Besides the fact that XOR-based schemes are not error resilient, transmission failures also invalidate the information used to make coding decisions, thus leading to decoding failures. Finally, note that, RLDP-HI either outperforms all schemes or it performs close to RLNC^G but with much less cost although the latter is rather unrealistic and features a much better performance compared to the more realistic RLNC^D. However, we chose RLNC^G in order to focus on the performance characteristics of intra-source coding and rule out other factors related to the distributed implementation.

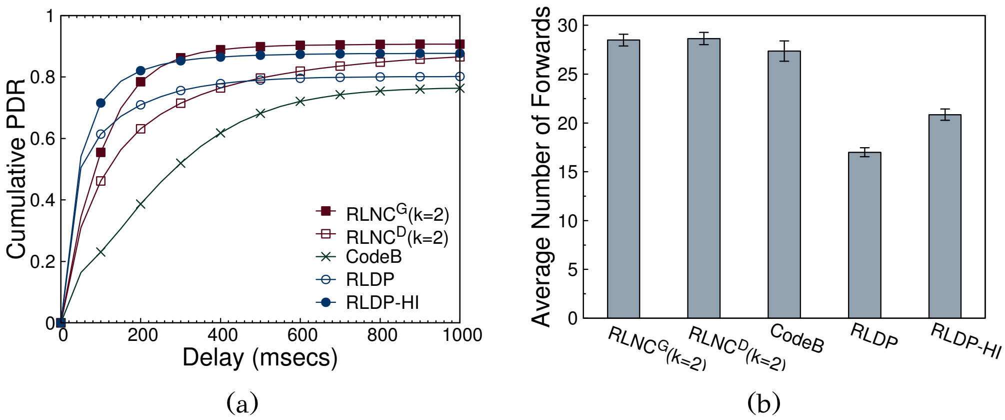


Figure 4.10: Performance of all schemes under Rayleigh fading (λ=1 pkt/sec/source, max speed:1 m/sec, N=100, "Sparse" topology): a) Cumulative PDR vs delay, (b) Avg. number of forwards

Finally, we conducted a set of experiments with the presence of channel fading using the well-known Rayleigh model. The model is appropriate for environments with many obstacles that block the line-of-sight between the transmitter and the receiver. In this experiment we do not introduce errors using an error model. Instead, transmission failures occur due to fading and are more frequent as the distance between the communicating nodes increases. Fig. 4.10a depicts the cumulative PDR vs the end-to-end delay. The performance of all schemes degrades compared to the case that there is no fading (Fig. 4.8a). The reasons are the same as those explained in the experiment with the uniform error model. Still, RLDP-HI outperforms all distributed schemes while its performance is comparable to that of RLNC^G. CodeB experiences a notable increase of the average number of forwards (Fig. 4.10b). This is because the effective transmission range is smaller than 250m since more distant nodes experience very poor link quality. As a result, the underlying PDP algorithm uses more forwarders to cover the same network area. Although both RLDP and RLDP-HI also rely on PDP, their efficient termination criteria allow them to suppress transmissions and outperform both RLNC and CodeB.

4.7 Summary

Random linear network coding is used to enhance the resilience of protocols to packet losses. We proved, through analysis, that we need to utilize a topology-aware algorithm in order to maximize its benefits. To this end, despite the common approach in the literature, which is to use random linear coding on top of probabilistic forwarding schemes, we chose the synergy with a CDS-based broadcast algorithm. Furthermore, we proposed an extension of the basic algorithm in order to enhance topology-awareness and cater for poorly connected nodes, especially when the packet loss rate is high. We demonstrated, through simulation, the efficiency of both approaches. Moreover, we provided a distributed mechanism for managing generations. The mechanism does not compromise the coding efficiency even in cases of high mobility and increased packet loss rate.

Chapter 5

Energy-efficient Routing in Opportunistic Networks

In order to cope with the intermittent connectivity of OppNets, routing schemes utilize the "store-carry-and-forward" paradigm constructing end-to-end paths gradually across time. Nodes store packets until they come in contact, i.e., move into the communication range of, other nodes. The dominant routing strategy uses packet replication to enhance the delivery efficiency. However, this strategy can lead to the creation of an excessive number of replicas thus exhausting the limited node resources such as energy and storage capacity. In this chapter, we present a simple yet efficient replication-based routing strategy that significantly improves the energy efficiency without compromising the overall performance in terms of delivery rate and delay.

5.1 Preliminaries

Replication-based strategies have been extensively used [1, 31–36, 78] in the context of routing in opportunistic networks in order to tackle the problem of intermittent connectivity, i.e., the absence of end-to-end paths. The idea is simple; spreading more replicas increases the probability that a node carrying the packet will meet, i.e., move into the communication range of, the destination. Although this strategy achieves high performance in terms of delivery rate and delay, this comes at the cost of more transmissions and increased storage requirements.

Controlling the level of replication allows for a trade-off between delivery rate and cost (both energy and storage related). To this end, utility-based replication [32–34, 36] is probably the most appealing strategy due to its capacity to adjust to diverse network characteristics. The idea here is to introduce a utility metric that captures the fitness (or quality) of a node for delivering and/or forwarding the packet and then create replicas by comparing the utility metrics of the nodes in contact. As discussed in section 2.2, there exists a diverse range of metrics [32–34, 36, 78–85] that are constructed from a node's feature such as the frequency or the regularity of its contacts, its importance in a social context, etc. Although the choice of the utility metric significantly impacts performance, it is common ground that, regardless of the metric used, utility-based replication frequently involves a high cost due to increased packet replication.

In order to reduce replication without significantly impacting the delivery rate, Delegation Forwarding (DF) [36] exploits the knowledge about past replication decisions by enabling each node to record the highest utility among its past contacts. This recorded value is the node's perception of the highest utility in the network, therefore no replication is performed if the contacting node has a lower utility. To further reduce the routing cost, Chen et al. [115] extend the delegation criterion by probabilistically pruning the set of possible packet carriers. However, this approach can lead to performance degradation in terms of packet delivery. Furthermore, determining the optimal probability for ignoring a possible carrier of the packet is a quite challenging task that strongly depends on the network characteristics. Gao et al. [116] also address the elimination of packet redundancy. However, their work considers only a subclass of destination independent utility metrics that are constructed as the sum of a node's ability to reach every other network node.

Our work builds on the premises of Delegation forwarding [36] due to its efficiency and its generic nature which allows it to be implemented with virtually any utility metric. We aim at further minimizing the replication cost without however sacrificing effectiveness by probabilistically suppressing replication. Instead, we take a deterministic approach and reduce the cost by taking advantage of the cooperation between nodes.

5.2 Problem formulation and motivation

To clearly demonstrate the motivation of our work, we first examine the approach taken in [36] which is to model the replication process using the well-known problem of optimal stopping theory known as the hiring problem [117]. The problem concerns a small start-up company with the ambition to develop into a colossal and successful enterprise. To this end, the company interviews candidates in order to expand its work force and maximize the average employee quality. The only constraint is that the decision to hire or not a candidate must be instantaneous. According to this modeling, a node carrying a packet corresponds to the interviewer while the nodes that it contacts correspond to the possible employees. Furthermore, note that each node receiving a packet copy may immediately also replicate this packet. This, in the context of the hiring problem, is equivalent to the scenario where each chosen applicant immediately acts as a job interviewer and meets new applicants. In other words, the company's hiring process contains multiple interviewers that evaluate candidates in parallel. One of the solutions to the hiring problem is known as the max strategy [117]. This strategy dictates that each candidate that has better quality from all current employees qualifies for a post in the company. Its main drawback is that it results in hiring a candidate more and more rarely as the time goes on [117]. However, in the aforementioned model, this drawback is eliminated by the parallel interviews which lead to a speed-up in the hiring rate [36]. This is why Delegation Forwarding (DF) [36] implements a policy for producing replicas which resembles the max strategy, i.e., a node receives a copy if its utility is higher than the highest recorded utility so far.

We make the observation that DF's policy on packet replication deviates from the max strategy producing unnecessary packet copies. More specifically, there are cases where nodes with lower utility than the highest in the network receive a replica. This is because each carrier node in DF generates and forwards packet replicas using its local view of the maximum utility. Instead, the replication process could be significantly improved if nodes are allowed exchange their view's. This is equivalent to say that, in the context of the hiring problem, the interviewers are able to exchange information about the hired employees. To this end, we propose a new routing strategy that, without incurring additional cost, exploits the recurring contacts between the packet carriers in order to coordinate them to an up to date view of the highest utility seen in the network. Moreover, we provide a lightweight extension based on bloom filters that further improves the energy efficiency by allowing nodes to deny receiving replicas for which they were previously rejected.

5.3 Coordinating replication decisions towards energy efficiency

We introduce the Coordinated Delegation Forwarding (COORD) algorithm which targets at reducing packet redundancy across the network. In COORD, each network node makes replication decisions by taking into account not only its own perception of the highest utility in the network but also the highest utility as perceived by other nodes. To accomplish this, the proposed algorithm utilizes the recurring contacts between nodes as well as contacts between nodes that already carry a packet replica. Fig. 5.1 illustrates an example where node v utilizes its recurring contact with node u in order to update its view regarding the highest utility in the network. This is in contrast to the common case where recurring contacts are never exploited. After the contact, both nodes use the same coordinated utility value, i.e., U'_{max}, to perform replication and avoid unnecessary future replications.

5.3.1 Protocol overview

COORD works in cooperation with any utility metric that portrays the node's fitness for delivering and/or forwarding a packet. In addition to U_v, which corresponds to v's value for a utility metric, COORD introduces the concept of:

Definition 5.1 (Coordinated Threshold: $c\tau^{p}_{v,t}$). The highest utility value among the nodes that carry a packet p which is known to node v at time t.

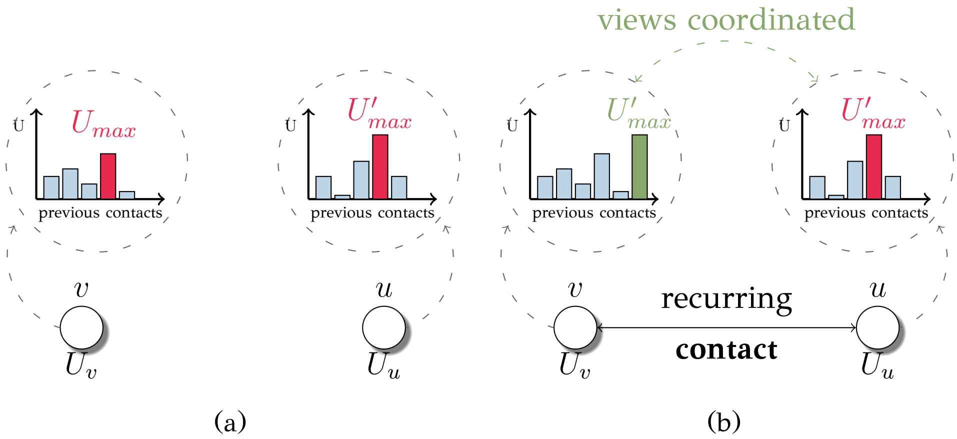


Figure 5.1: Example where node v exploits its recurring contact with node u in order to update its perception of the highest utility value in the network. Status of both nodes: (a) before the recurring contact arises, and (b) after the recurring contact occurs and coordination of the nodes' views takes place.

Each node maintains a coordinated threshold for each packet that it carries. This threshold is initialized with the utility of the node ($c\tau^p_{v,t} = U_v$) when the packet is initially received. Then, replication is performed by comparing the coordinated threshold of the node carrying the packet and the utility value of the candidate node. In the event of a contact between nodes v and u at time t, node v executes the forwarding procedure described in algorithm 5.1. More specifically, a packet p, carried by v but not by u, is replicated only if $c\tau^p_{v,t} < U_u$ (line 5, algorithm 5.1), i.e., only if node u's utility is higher than the highest utility known to v. Furthermore, the coordinated threshold is updated, i.e., $c\tau^p_{v,t} \leftarrow U_u$, since U_u is the new highest value known to v.

The innovation of COORD is that a node v, when in contact with node u, is able to take advantage of the coordinated threshold of u for each packet that both nodes carry (line 2, algorithm 5.1), i.e., $c\tau^p_{v,t} \leftarrow c\tau^p_{u,t}$ if $c\tau^p_{u,t} > c\tau^p_{v,t}$. This is done in order to allow v to increase its own coordinated threshold and therefore reduce future replication.

To illustrate the rationale behind COORD's approach, let us express more formally the update process of the coordinated threshold. Let K_v denote the set of v's contacts up to time t. Furthermore, let $<u, T>$ denote a contact of v with node u at time T. Then, the coordinated threshold of v for packet p can be expressed as:

Algorithm 5.1 COORD's forwarding procedure when node v encounters u at time t.

for every packet $p \in Buf_v$ **do**
    if $p \in Buf_u$ **and** $c\tau^p_{v,t} < c\tau^p_{u,t}$ **then**
        $c\tau^p_{v,t} \leftarrow c\tau^p_{u,t}$
    else
        if $c\tau^p_{v,t} < U_u$ **then**
            Forward p to node u
            $c\tau^p_{v,t} \leftarrow U_u$
        end if
    end if
end for

$$c\tau^p_{v,t} = \max_{\forall <u,T> \in K_v} \{c\tau^p_{u,T}, U_v, U_u\} \tag{5.1}$$

Note that U_v is the maximum value when the packet is first received by v while U_u is the maximum value when v replicates the packet to u (in this case it is also $c\tau^p_{u,T} = U_u$). Moreover, observe that the approach of DF is equivalent to the following update process:

$$\tau^p_{v,t} = \max_{\forall <u,T> \in K_v} \{U_v, U_u\} \tag{5.2}$$

where $\tau^p_{v,t}$ denotes the corresponding threshold in the case of DF. It is clear from (5.2) that in DF v uses only information regarding the utility of the nodes it meets. On the contrary, in COORD, v is able to exploit the utility of nodes which has never contacted. This is possible through $c\tau^p_{u,T}$ since it contains a "summary" of the utilities of u's contacts which, in general, are different from v's contacts. In other words, COORD exploits the recurring meetings between nodes in order to disseminate the highest utility seen across the network. To further illustrate the advantages of our approach, consider the example in Fig. 5.2. The figure is composed of two parts; the replication tree of a packet p originated at node S and the contact table containing all node contacts that occurred from p's generation until its delivery. At time t_3 node C, which has a high utility value, receives p. DF uses this information to suppress future replication performed only by node S. On the other hand, COORD exploits the recurring contacts, at time instances t_4 and t_7, to coordinate the views of nodes C,S,A and B. More specifically, A and B update their threshold values for packet p

					COORD		DF	
time	v	u	U_v	U_u	$c\tau_v^p$	$c\tau_u^p$	τ_v^p	τ_u^p
t_1	S	A	0.2	0.3	0.2	0.3	0.2	0.3
t_2	S	B	0.2	0.5	0.3	0.5	0.3	0.5
t_3	S	C	0.2	0.9	0.5	0.9	0.5	0.9
t_4	S	A	0.2	0.3	0.9	0.3	0.9	0.3
t_5	A	I	0.3	0.5	0.9	0.5	0.3	0.5
t_6	I	E	0.5	0.6	0.9	0.6	0.5	0.6
t_7	A	B	0.3	0.5	0.9	0.5	0.5	0.5
t_8	B	G	0.5	0.8	0.9	0.8	0.5	0.8
t_9	B	H	0.5	0.9	0.9	0.9	0.8	0.9
t_{10}	I	F	0.5	0.7	0.9	0.7	0.5	0.7

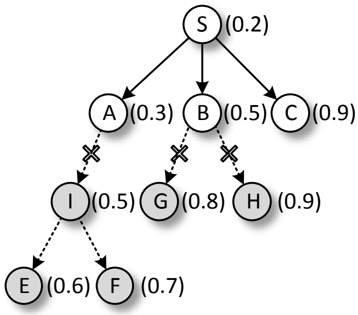


Figure 5.2: Example where COORD achieves a reduction of $\sim$55% in the number of replications compared to DF. The table describes the node meetings with the corresponding utility values, while the replication tree shows the replications performed by both algorithms.

to 0.9 although they never meet C. As a result, the probability of performing future replication is also reduced for A and B. The outcome is a cost reduction of $\sim$55%. Finally, it is important to note that our approach comes at no additional cost, as no additional storage or transmissions of control packets are required compared to DF. The nodes in contact just need to exchange, for the packets that both carry, their coordinated thresholds instead of exchanging their utilities.

5.3.2 Cost analysis

Similar to the plethora of multi-copy routing protocols we define the routing cost as the total number of packet replications that occur across the network. This definition captures both the number of transmissions, which is associated with node energy, as well as the packet load which is correlated to the storage ability of nodes. Without loss of generality, in the following we focus on the case of routing a single packet p. Furthermore, we assume that for a node v both U_v and $c\tau_{v,t}$ are normalized and take values in $[0, 1]$. Since DF is proved to have a lower cost than the Compare & Replicate scheme [36], we focus on showing that COORD's cost is lower than that of DF. To this end, we first show that:

Lemma 5.1. *The coordinated threshold for a packet p at a node v is always greater or equal to the corresponding threshold of DF, i.e.,* $c\tau^p_{v,t} \geq \tau^p_{v,t}, \forall t$.

Proof. We prove this Lemma by induction. Let T_0 denote the time of the contact over which node v first receives p. Then, according to (5.1) and (5.2), $c\tau_{v,T_0} = \tau_{v,T_0} = U_v$[1] since v receives a copy only if U_v is the highest value. Let T denote the time that the (k-1)-th contact occurs while T' denotes the time of the k-th contact and u denotes the contacting node with utility U_u. Furthermore, for simplicity, we use τ to denote the threshold of u regardless of whether COORD or DF is used. Note that if u already has a copy of p then, by definition, $\tau \geq U_u$ otherwise τ is undefined. Since in both algorithms the threshold of a node is updated in a contact basis, it is sufficient to show that if $c\tau_{v,T} \geq \tau_{v,T}$ then $c\tau_{v,T'} \geq \tau_{v,T'}$. Fig. 5.3 illustrates all the cases that, according to (5.1) and (5.2), result in updating $c\tau_{v,T'}$ and/or $\tau_{v,T'}$:

- Case A: Node u already has a copy of p and $U_u \in [0, \tau_{v,T}]$ while $\tau \geq c\tau_{v,T}$. In this case the new thresholds are $\tau_{v,T'} = \tau_{v,T}$ and $c\tau_{v,T'} = \tau$. In other words, the coordinated threshold is updated while the same does not happen for the DF case and as a result $c\tau_{v,T'} \geq \tau_{v,T'}$.

- Case B: Node u does not carry a copy of p and $U_u \geq c\tau_{v,T}$. The new thresholds are $c\tau_{v,T'} = \tau_{v,T'} = U_u$.

- Case C: Node u already has a copy of p and $c\tau_{v,T} \leq U_u \leq \tau$. After the update process $\tau_{v,T'} = U_u$ and $c\tau_{v,T'} = \tau$, therefore $c\tau_{v,T'} \geq \tau_{v,T'}$.

- Case D: Node u does not carry a copy of p and $U_u \in [\tau_{v,T}, c\tau_{v,T}]$. The update process will result in $\tau_{v,T'} = U_u$ and $c\tau_{v,T'} = c\tau_{v,T}$ and consequently $c\tau_{v,T'} \geq \tau_{v,T'}$.

- Case E: Node u already has a copy of p and $\tau_{v,T} \leq U_u \leq \tau \leq c\tau_{v,T}$. After the update $\tau_{v,T'} = U_u$ and $c\tau_{v,T'} = c\tau_{v,T}$ and as a result $c\tau_{v,T'} \geq \tau_{v,T'}$.

- Case F: Node u already has a copy of p and $\tau_{v,T} \leq U_u \leq c\tau_{v,T} \leq \tau$. The updated thresholds are $\tau_{v,T'} = U_u$ and $c\tau_{v,T'} = \tau$ which results in $c\tau_{v,T'} \geq \tau_{v,T'}$.

Note that in all of the aforementioned cases $c\tau_{v,T'} \geq \tau_{v,T'}$. Furthermore, in all other cases, the utility levels remain the same and consequently again $c\tau_{v,T'} \geq \tau_{v,T'}$. $\square$

With the help of Lemma 5.1 we can now prove that:

Theorem 5.1. *The routing cost of COORD is lower or equal to the routing cost of Delegation forwarding.*

[1]Hereafter, for simplicity, we omit the superscript p in the related notation.

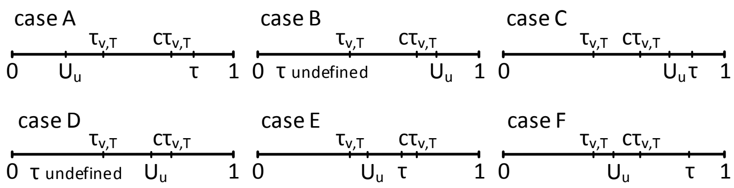


Figure 5.3: Node v encounters node u: cases that result in an update of $c\tau_{v,T'}$ and/or $\tau_{v,T'}$

Proof. If N is the number of nodes in the network then the average cost for routing a single packet p is defined as:

$$\mathcal{R} = 1 + \sum_{\forall v}(1 - \mathrm{pn}_v)$$

where pn_v is the probability that a network node v will not receive a copy of p. This probability is equal to the probability that the utility of v is lower than the lower threshold among the nodes that v meets, i.e.:

$$\mathrm{pn}_v^{\mathrm{C}} = \mathrm{P}[\min_{\forall<u,T>\in K_v}\{c\tau_{u,T}^{p}\} \geq U_v]$$

in the case of COORD while in the case of DF:

$$\mathrm{pn}_v^{\mathrm{D}} = \mathrm{P}[\min_{\forall<u,T>\in K_v}\{\tau_{u,T}^{p}\} \geq U_v]$$

Hence, it suffices to show that $\mathrm{pn}_v^{\mathrm{C}} \geq \mathrm{pn}_v^{\mathrm{D}}, \forall v$ or equivalently:

$$\min_{\forall<u,T>\in K_v}\{c\tau_{u,T}^{p}\} \geq \min_{\forall<u,T>\in K_v}\{\tau_{u,T}^{p}\}$$

which is directly derived from Lemma 5.1 since the lemma holds for all nodes. □

Note that $\mathcal{R}^{\mathrm{C}} = \mathcal{R}^{\mathrm{D}}$, i.e., the routing cost of COORD is equal to that of DF, only when a packet is replicated to nodes with increasingly higher utility even if the replication is performed by different nodes. It is clear that this is a rather unlikely case.

5.4 Evaluation

In this section we evaluate the performance of COORD under various opportunistic environments. To this end, we developed an event-driven simulator, called Adyton [118], that operates on a contact basis and is capable of processing real world

Table 5.1: Properties of opportunistic traces

Trace Name	# Nodes	Duration (days)	Area
Infocom '05 [121]	41	3	conference
Sigcomm '09 [123]	76	3.7	conference
MIT Reality [125]	97	283	campus
Milano pmtr [127]	44	18.9	campus
Cambridge upmc [129]	52	11.4	city

traces [119]. Adyton includes a plethora of routing protocols and real-world contact traces, while it also provides several congestion control mechanisms and buffer management policies. More information on Adyton's implementation and features is presented in section 6.1.

5.4.1 Simulation environment

We use three classes of traces, each one corresponding to an opportunistic environment of different scale. The first class consists of two conference traces, the Infocom'05 [120, 121] and the Sigcomm'09 [122, 123]. The second class consists of campus traces, where the participants are students and faculty members that move in a larger area than in the conference case. We use the well known MIT Reality dataset [124, 125] and the Milano pmtr dataset [126, 127]. The latter utilizes a short beaconing scheme to achieve a more fine-grained view of contact records. Finally, the last dataset is the upmc/content [128, 129], collected in the city of Cambridge, UK. This is actually a city-level trace in which the network expands to the city limits. Table 5.1 summarizes the characteristics of the used traces.

As previously mentioned, COORD is designed to synergistically operate with any utility metric of the literature. Nevertheless, the choice of the corresponding utility metric has a significant impact on COORD's performance gain. To map such dependencies, we assess the performance of COORD when combined with various utility metrics both destination dependent and independent ones. Furthermore, we experiment on hybrid utility metrics that are composed of multiple individual destination dependent and independent metrics. More specifically, we use the following utility metrics:

- LTS [32, 80]: This is a destination dependent metric with values in $[0, 1]$. It is calculated as $1/(1 + \text{LastTime})$, where LastTime is the elapsed time since the last contact with the destination.

- ENC [34,78]: This metric captures the total number of contacts with all network nodes (destination independent).

- SPM [83, 84]: Social Pressure Metric is a destination dependent metric that captures the friendship between network nodes. It is estimated locally by each node using the frequency, the longevity and the regularity of past node contacts.

- PRoPHET [81, 82]: PRoPHET is destination dependent metric utilized by the well-known PRoPHET algorithm. It is based on the delivery predictability that measures the probability of encountering a node. The metric is calculated locally at each node and is updated on a contact basis. Moreover, it is enhanced with an aging mechanism as well as the transitive property.

- SimBetTS [33]: SimBetTS is a hybrid metric that is composed of five individual utility metrics; one destination independent and four destination dependent. The destination independent metric is Betweenness Centrality [86] that measures to what extend the node lies on the shortest paths from all nodes to all other. The distributed version of this utility, i.e., Ego Betweenness [85], is calculated using the local contact graph (i.e., ego network) of each node. The local contact graph is also used to calculate the Similarity metric that is destination dependent and measures the number of common neighbors between two nodes. The remaining metrics are tie strength indicators [87] that measure how strong or weak is the relationship among network nodes. The frequency metric is based on the number of encounters of node with the other network nodes, the intimacy/closeness metric uses the duration of the encounters between network nodes, while the recency metric is based on the amount of time passed since the last contact between two nodes.

The reported results are obtained as the average of 50 repetitions. In each repetition we randomly select the source/destination pair and the generation time for each packet. Furthermore, packets are generated uniformly in the interval during which both the source and the destination are present in the network. To avoid statistical

Table 5.2: Average packet delay of COORD normalized to that of DF (traffic load: 5000 packets, storage capacity: unlimited)

Trace	LTS	ENC	SPM	PRoPHET	SimBetTS
MIT Reality	1.016	1.019	1.020	1.017	1.081
Milano pmtr	1.003	1.001	1.000	0.999	1.003
Infocom '05	1.006	1.003	1.005	1.003	1.029
Sigcomm '09	1.002	0.989	0.996	1.001	1.011
Cambridge upmc	1.0	0.956	1.001	1.001	1.022

bias, we use a warm-up and a cool-down period during which packets are not generated. We chose the duration of each period to be 20% of the total trace duration in order to perform simulations when the network is in steady state. In all cases, the confidence interval (with a 95% confidence level) of the reported results is less than 0.9%.

5.4.2 Simulation results

We conducted three sets of simulations to examine the performance gains of COORD over DF. For the comparison, we use the following three performance indexes: the delivery ratio (i.e., the fraction of generated packets delivered to their destination), the routing cost (i.e., the total number of transmissions) and the average packet delay.

In the first set of simulations, we assess the performance of COORD and DF under a diverse range of real world traces. We assume that the storage capability of each node is unlimited. We will explore the limited storage case in the third experiment. Fig. 5.4 illustrates the performance of both COORD and DF for the five different opportunistic traces. Each of the first five plots corresponds to a different trace. It depicts the delivery ratio vs the routing cost for both DF and COORD when various utilities are used. The last plot in Fig. 5.4 presents an overview of COORD's routing cost gains, i.e., its routing cost normalized to that of DF. Note that for the first five plots in Fig. 5.4 the best performing protocol is the one that lies closer to the top left corner of the graph, i.e., it achieves the highest delivery ratio and the lowest routing cost. COORD clearly achieves a remarkable routing cost reduction compared to DF. More specifically, the resulting gain ranges from 10% to 60% depending on the utility metric and the trace under consideration. As previously discussed, slowing

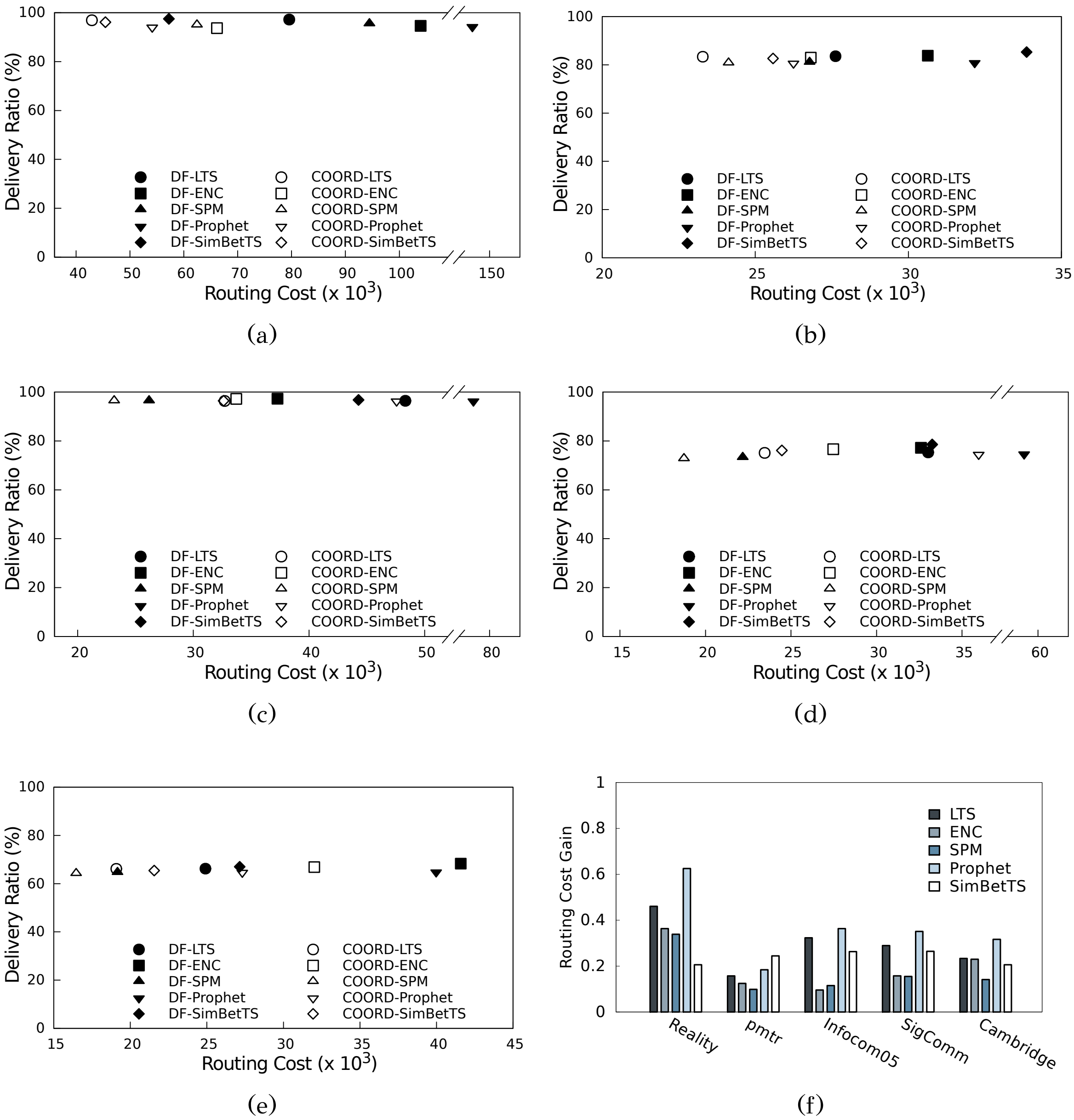
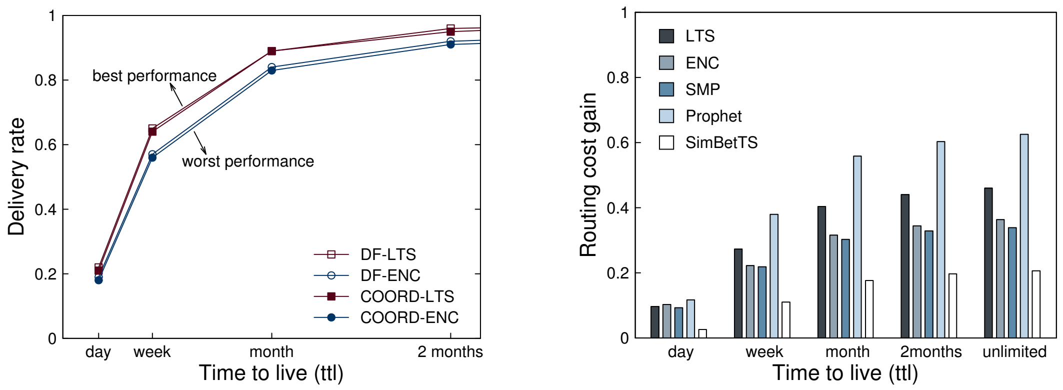


Figure 5.4: Performance comparison of COORD and DF under different opportunistic environments (traffic load: 5000 packets, storage capacity: unlimited): (a) MIT Reality (b) Milano pmtr (c) Infocom '05 (d) SigComm '09 (e) Cambridge upmc (f) Total gain in the routing cost

down packet replication, and thus pruning transmissions, could result to performance degradation in terms of delivery ratio and end-to-end delivery delay. Interestingly, COORD's cost gains come with no or minor impact on the delivery ratio and the end-to-end delay. More specifically, in the case of destination dependent metrics (LTS, SPM and PRoPHET) there is no apparent impact on the delivery ratio. Likewise, the increase of the average end-to-end delay is less than 2% (Table 5.2). In the case of destination independent metrics (ENC), there is a marginal impact (less than 2% in the worst case) on both delivery ratio and the end-to-end delay. To explain this, recall that COORD avoids replicating packets to nodes with low utility values assuming that a low value corresponds to a limited capability for delivering the packet. However, when a destination independent metric is utilized this is not always the case, e.g., a node that has a secondary role in maintaining network's connectivity (low utility value) may have an increased capability to deliver a specific packet. However, this can hardly be considered as a shortcoming of COORD at it is actually a well-known disadvantage of destination independent metrics. In the case of hybrid metrics (SimBetTS) the impact on the delivery ratio and the end-to-end delay is insignificant for most of the traces under consideration. In the worst case there is a delivery ratio reduction of less than 3.2% while in most cases the increase in average delay is less than 2.9%. The only exception is the MIT Reality trace where the average packet delay increases by 8.1% (Table 5.2). We attribute this behavior to the SimBetTS metric which is a sum of other metrics (both destination dependent and independent ones). Keeping just a single coordinated threshold $c\tau$ fails to correctly identify good forwarding opportunities. To better understand this, consider a scenario where nodes v and u meet. Assume that the used utility is a hybrid one with two components, one destination dependent and the other independent. Now assume that node v has a coordinated threshold $c\tau_v$ that is better than u's utility value, i.e., $c\tau_v > U_u$. As a result, node u will never receive a packet replica from v. However, this is not always the best decision. The high value of $c\tau_v$ may be the result of a high destination independent component while U_u may contain a high destination dependent component. In this case, replicating the packet to node u could be advantageous. To overcome this problem, a better approach would be to utilize one coordinated threshold for each component rather than one coordinated threshold for the hybrid utility. We plan to investigate such an approach in the future.

In the second set of simulations we explore the performance gains of COORD over DF when a Time-to-Live (TTL) deletion mechanism is utilized. The key idea behind this mechanism is to discard packets that exceed a predefined time limit. Deleting old packet replicas indirectly bounds replication and reserves network resources. However, predefining a proper TTL value is very hard since it directly affects the delivery efficiency of the routing algorithm. In the following, we will focus on the MIT Reality



(a) (b)

Figure 5.5: Performance comparison of COORD and DF under different TTL values (real-world trace: MIT Reality, traffic load: 5000 packets, storage capacity: unlimited): (a) Delivery rate (b) total gain in the routing cost

trace although we observed similar findings for all other traces. Fig. 5.5a illustrates the correlation between TTL value and packet delivery for COORD and DF. We present results for the best (LTS) and worst (ENC) performing utility metric. In both cases, small TTL values (equal to one day) result in poor delivery performance (∼20%) while larger TTL values (larger than a month) allows the delivery of most packets. Interestingly, COORD exhibits the same delivery efficiency as DF does regardless of the TTL value. At the same time it achieves a remarkable reduction of routing cost that in some cases is close to ∼60% (Fig. 5.5b). Overall, the results confirm that COORD successfully prunes redundant packet replications even when small TTL values limit the replication process. For larger values of TLL COORD manages a substantial routing gain that becomes more evident as TTL increases.

Despite the technological advancements in hardware the required memory for storing packets still raises concerns. Furthermore, using a large storage space increases the processing requirements as packet handling becomes more challenging. Therefore, in the last set of simulations we examine COORD and DF in a network of nodes with limited storage capacity. Stored packets are served using the FIFO policy and the oldest packet is discarded when the storage buffer gets full. Fig. 5.6a demonstrates the delivery efficiency of COORD and DF for different values of storage capacity (in packets). Again, only the best and worst performing utility metrics are presented since all other utilities demonstrate a similar behavior. As expected, the storage restrictions severely affect the delivery performance in all cases. Finite

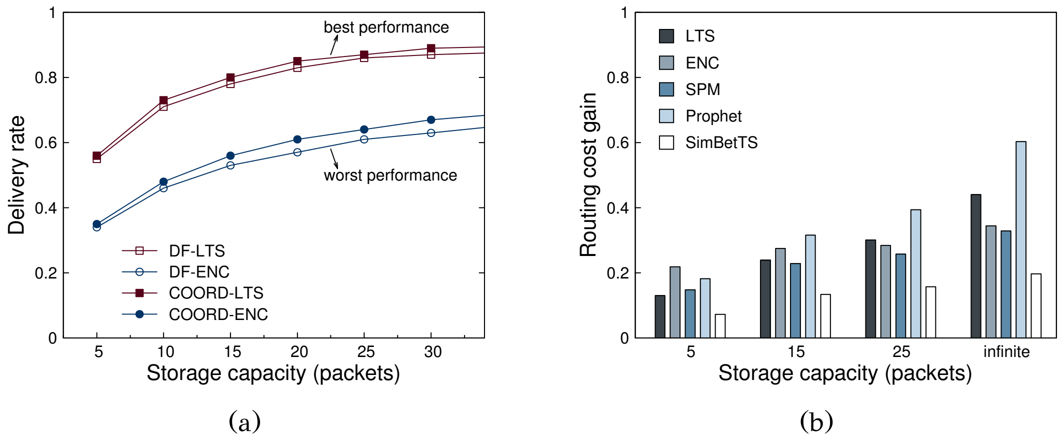


Figure 5.6: Performance comparison of COORD and DF under varying storage capacity (real-world trace: MIT Reality, traffic load: 5000 packets): (a) Delivery rate (b) total gain in the routing cost

storage buffers limit the number of packet replicas across the network decreasing the overall delivery probability. Observe however that COORD appears to outperform DF in terms of both delivery efficiency and routing cost. While the gain in cost is well expected based on COORD's design, the improvement of delivery efficiency is somehow less clear. Since COORD creates less replicas the available storage space in each node is higher. Consequently, less packets are dropped and thus they have an increased probability of being delivered. When the storage capacity is minimal (5 packets) COORD and DF perform similar because in this case the storage capacity is so limited that every node is under congestion even with minimal packet replication. On the other hand, COORD performs increasingly better as the storage capacity increases. However, we have observed that when the storage capacity increases beyond a limit COORD and DF converge again to the same delivery efficiency. This is reasonable because in this case the packet dropping rate becomes negligible.

5.5 Reducing the routing cost through past exclusions

So far we have focused on the synchronization of nodes that make replication decisions. However, a closer look at the replication process reveals that non-carrier nodes, i.e., nodes that do not hold a replica, could also be part of the synchronization process instead of playing a passive role just waiting for carriers to make replication decisions. This would be beneficial to the replication process since non-carriers nodes can provide valuable information to the carrier ones. Such information is whether the non-carrier node has been rejected as a carrier of a packet in the past. To understand how this information could be of use let us consider a simple scenario of a node v coming into contact with node u. Suppose that v would normally replicate the packet to u, i.e., $U_u \geq c\tau_v$. If some node z rejected replication to u in the past this means that $c\tau_z > U_v$ and therefore a node with a higher utility value already exists in the network. Therefore v should cancel replication. For efficiency, instead of burdening the carrier node we implement the aforementioned policy at the non-carrier side by allowing it to reject a packet replica according to the following criterion:

Definition 5.2 (Exclusion criterion). Node u refrains from receiving a copy of packet m, if there exists a node z that, in the past, denied u the replication of that packet.

We call the extended algorithm COORD-EC (COORD with exclusion criterion). Note that the exclusion criterion is fully compatible with COORD's strategy to replicate packets only to nodes that hold a better utility value than the highest utility value in the network. Its novelty is that it allows carrier nodes to coordinate indirectly by using non-carrier ones. This ability is more critical when carriers have inconsistent views of the highest utility value in the network. This is a rather common case due to the infrequent contact rate in such networks. By engaging non-carrier nodes in the replication process the rate of useful contacts increases, allowing a more complete coordination that results in better replication decisions.

To better understand the benefits using the exclusion criterion, consider the example depicted in Fig. 5.7. Initially, nodes v and z hold a replica of packet m, while node u is unaware of m. Furthermore, assume that z's view of the highest utility in the network is $c\tau_z = 0.9$ while v's is only 0.5. Node u has the opportunity to become a carrier for packet m at time instances t_1 and t_2 ($t_2 > t_1$) when it meets nodes z and v, respectively. Node z rejects packet replication in v at t_1 since v has a smaller utility value than z's threshold, i.e., $U_u < c\tau_z$. In case that the exclusion criterion is not utilized (Fig. 5.7a), node v copies m to node u at t_2 due to its inconsistent perception of the highest utility in the network. On the contrary, when the exclusion criterion is used (Fig. 5.7b), node u exploits its previous rejection from z and refuses a replica of m from v, thus saving an extra transmission.

There are two requirements for implementing the exclusion criterion. The first

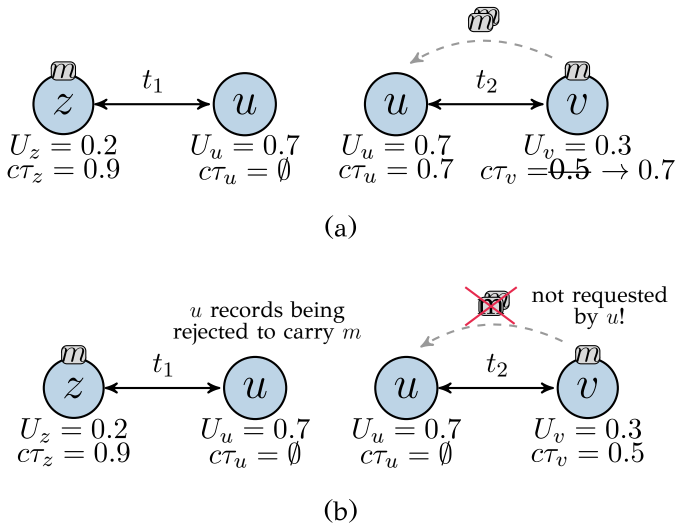


Figure 5.7: Example of the replication process under (a) COORD and (b) COORD-EC where the exclusion criterion is used.

is that a node should be able to identify packets for which was not selected as a carrier. Recall that such information is not distributed during the anti-entropy session [1], typically used by all multi-copy routing protocols at the beginning of a contact. Fortunately, obtaining such information requires no additional exchange of control messages. Bear in mind that during the anti-entropy session a node typically receives a list of packets carried by the encountered node in order to determine the set of unknown packets. By comparing this list with the list of packets that it eventually receives, the node can infer the set of packets for which it was not selected as a carrier. In the example of Fig. 5.7, not receiving packet m during the contact with z allows node u to infer that z had denied the replication of m. The second and more challenging requirement relates to storing and efficiently accessing what we call the *denial list*, i.e., a list containing all the packets ids for which a node was not selected as a carrier. The size of such a list calls for an efficient and cost effective representation method. To this end, we utilize Bloom filters [130], a data structure of low storage requirements that provides fast membership queries. We examine the use of Bloom filters and its implications in the following.

To explore the gains of using the exclusion criterion, we run a series of simulations using Adyton [118]. We use two versions of COORD; one that corresponds to the basic protocol as described in section 5.3 (identified as COORD) and one that utilizes the exclusion criterion (identified as COORD-EC). For brevity, we only report our

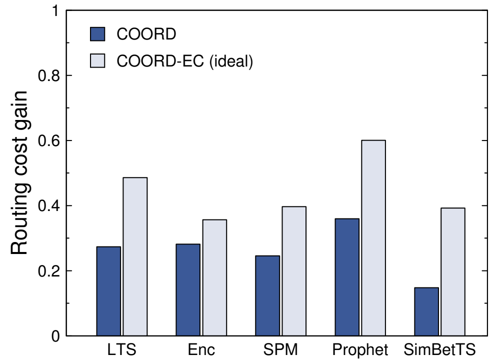


Figure 5.8: Routing gain of COORD and COORD-EC compared to Delegation Forwarding (trace: MIT Reality, traffic load: 5000 pkts, storage capacity: 20 pkts)

Table 5.3: Average packet delivery ratio (%) of DF, COORD and COORD-EC (trace: MIT Reality, traffic load: 5000 pkts, storage capacity: 20 pkts)

Protocol	LTS	ENC	SPM	PRoPHET	SimBetTS
DF	82.73	57.45	76.49	74.53	82.14
COORD	84.74	60.52	78.27	78.81	81.08
COORD-EC	84.07	62.10	78.91	77.39	80.86

Table 5.4: Average packet delay (days) of DF, COORD and COORD-EC (trace: MIT Reality, traffic load: 5000 pkts, storage capacity: 20 pkts)

Protocol	LTS	ENC	SPM	PRoPHET	SimBetTS
DF	6.10	7.32	6.03	6.01	6.10
COORD	6.29	7.18	6.20	6.24	6.30
COORD-EC	6.32	6.97	6.16	6.29	6.20

findings using the MIT Reality real-world trace but we observed qualitatively similar results for the other traces as well. In our simulation scenario we generate 5000 packets with random source/destination pairs. Each packet replica has a maximum TTL value equal to 20% of the total trace duration, which in the MIT Reality case is approximately two months. Furthermore, in order to be more realistic, we limit the storing ability of nodes to a maximum of 20 packets. When the storage buffer of a node is depleted, then the oldest packet is discarded.

Fig. 5.8 illustrates the routing cost gain of COORD and COORD-EC compared to Delegation Forwarding. In this experiment we use a version of COORD-EC where we assume an ideal data structure for implementing the denial list, i.e., every node can store an unlimited number of packet identifiers. This allows us to examine the upper performance bound when we use the exclusion criterion. Later on, we examine the performance of a more realistic version of COORD-EC. The results justify our approach; COORD-EC exhibits an enhanced pruning efficiency which results in an up to 25% improvement compared to COORD. With respect to Delegation Forwarding, COORD-EC produces 35% to 60% less transmissions, depending on the utility metric in use. Reducing packet replicas also alleviates the storage requirements at each node. As a result, nodes use the saved space in their buffers to store other replicas, improving, in most cases, the overall delivery rate (Table 5.3). However, for some utility metrics the exclusion criterion introduces a slight increase in delivery delay (Table 5.4). Not only this increase is negligible but to this extent it is also reasonable if we bear in mind that the efficiency of a utility metric is limited. This means that there are some rare cases where forwarding replicas to nodes with smaller utility values can lead to faster delivery paths. However, such cases should be tackled by the utility metric itself and not by the utility-based replication mechanism.

As previously discussed, the challenge in COORD-EC lies in the efficient implementation of the denial list. We wish to use limited space for storing such a list of packet identifiers and at the same time quickly access the list for making a decision on whether an identifier is in it or not (membership query). A Bloom filter is a data structure that can provide these operational characteristics. More specifically, a Bloom filter can be used for representing a set summary. It consists of an array of M bits and a small number of k hash functions. An element (packet id in our case) is added by setting to 1 the k bits identified by the k hash functions when the input is the element itself. The membership query is fast ($O(k)$) since it only involves the use of the k hash functions. The trade-off for the space efficiency of Bloom filters is their probabilistic nature, i.e., there is always a *false positive* probability f_p when performing a membership query. That is, the response that an element exists in the filter may be erroneous. In our context, this corresponds to an erroneous reply that a packet identifier exists in the denial list. It is clear that such false positives may impair the routing process by erroneously abolishing essential replication decisions. For that reason, the false positive probability f_p should be kept as low as possible. Note that f_p depends on k, M as well as the maximum number of elements stored in the filter El_{max} [130]:

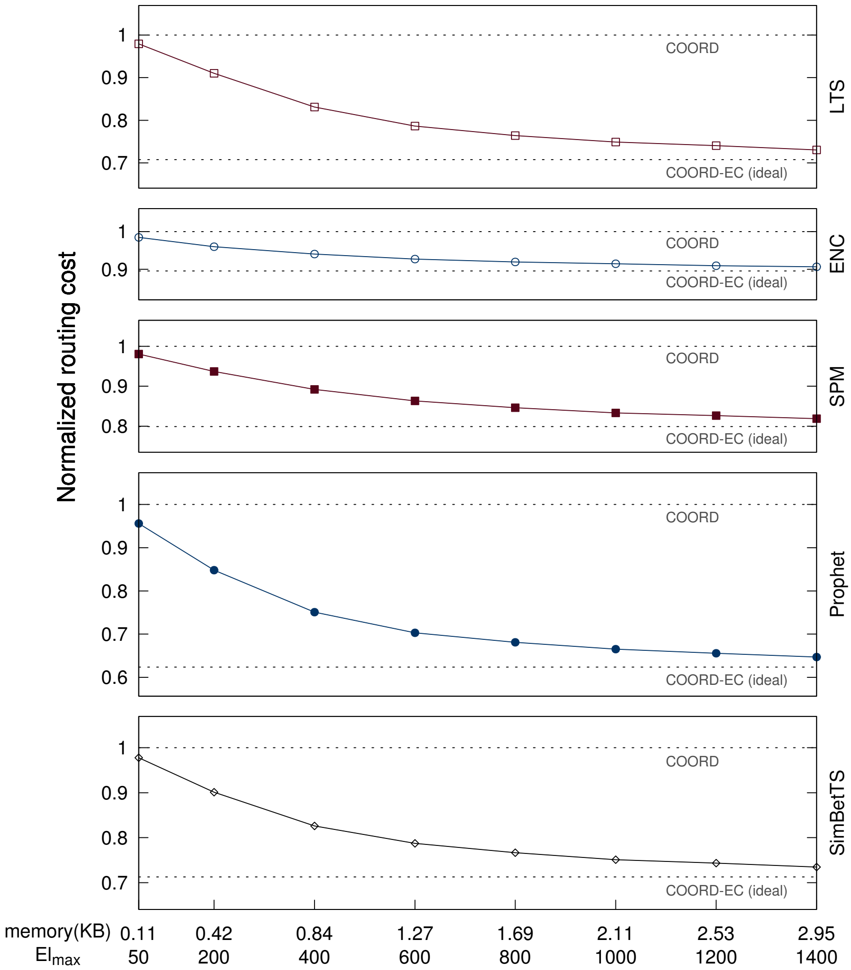


Figure 5.9: Routing cost of COORD-EC (with bloom filters) normalized to that of COORD under varying Bloom filter size and for various utility metrics (trace: MIT Reality, traffic load: 5000 packets, storage capacity: 20 packets, false positive probability: 0.1%)

$$f_p \simeq (1 - e^{-\frac{kEl_{max}}{M}})^k \tag{5.3}$$

The actual false positive probability is less than the one defined in the previous equation if the number of elements stored in the filter is smaller than El_{max}. Keeping in mind that usually a node has a predefined storage space, we can use the previous equation to determine the maximum number of packet ids (El_{max}) that can be stored in the Bloom filter in order for f_p not to exceed a predefined value. Since a denial list is continuously populated with more packet ids, we decided to reset the Bloom filter each time a node accumulates $El_{max}+1$ packet ids. This approach guarantees that the false positive rate never exceeds the predefined value therefore the impact in the replication process is limited. On the other hand, deleting old packet entries means that the replication process for those packets switches to the typical COORD operation, i.e., the replica pruning capability for those packets deteriorates. Therefore, it is clear that, for a given false positive rate, the size of the filter (or equivalently the maximum number of elements) determines COORD-EC's pruning efficiency. The greater the size the better the pruning capability and vise-versa.

To study the aforementioned trade-off we use the previous simulation environment and examine the routing cost of COORD-EC with Bloom filters of different sizes. Recall that according to (5.3), for a given false positive rate, the filter size determines the maximum number of stored elements. We use a maximum false positive rate of 0.1% for securing minimal impact on the algorithm's delivery efficiency. Furthermore, we choose $k = 5$ as a realistic setting for quick membership decisions in a real-life scenario. Fig. 5.9 illustrates the routing cost of COORD-EC normalized to that of COORD for different sizes of the Bloom filter and for different utilities. In the x-axis, along with the filter size we also report the corresponding El_{max} value. For comparison, we also report the performance of the ideal case of COORD-EC, i.e., the one that every node stores all packets ids. Note that both COORD and the ideal case of COORD-EC do not use Bloom filters so their performance is unaffected (both displayed in dashed lines). Clearly, when the Bloom filter size is small, COORD-EC tends to behave similar to COORD. This is reasonable since in this case only 50 packet ids can be stored in the denial list. As a result, the list is frequently reset and a node has little information for denying redundant replicas. On the other hand, as the Bloom filter size increases COORD-EC converges to the performance of the ideal case. A bigger filter size corresponds to an increased El_{max}. As a result, packet ids are deleted more infrequently and a node has more time to use the information for rejecting replicas. Even when the filter is reset, the deleted information is probably not of use since it refers to old packets for which the replication process is probably already completed. Interestingly, in all cases of our simulation scenarios, a Bloom filter of $\sim 2.95\text{KB}$ can deliver a performance close to that of the ideal case while a filter as small as $\sim 1.5\text{KB}$ is sufficient for providing most of the performance improvement. As a final note, it is important that we can enjoy such gains using filters of relatively small size even when we choose a low false positive rate such as 0.1%. This secures a minimal impact on the protocol's delivery efficiency which is confirmed by our results.

5.6 Summary

We proposed an energy-efficient multi-copy routing strategy that can work in synergy with virtually any utility metric in OppNets. The algorithm allows nodes to coordinate their views regarding the replication process and in this way achieves to significantly reduce the energy consumption at intermediate nodes without sacrificing its efficiency in terms of delivery rate and end-to-end delay. We demonstrated, through analysis and extensive simulations in a diverse range of opportunistic environments, the performance gains of the proposed algorithm over delegation forwarding, which is one of the state-of-the-art algorithms in this category. Furthermore, we provided an extension to the proposed algorithm that further improves the energy efficiency by allowing nodes to deny receiving replicas for which they were previously rejected in their past contacts.

Chapter 6

Other Challenges in Opportunistic Networks

In this chapter we discuss some challenging issues faced when conducting research on opportunistic networks. In particular, we focus on methods and tools for evaluating the performance of routing protocols, techniques for producing large scale traces and congestion control mechanisms for opportunistic networks. We present our solutions for each of the aforementioned issues. Specifically, we present a simulation tool specially designed to operate on real-world traces, a paradigm for generating large scale traces using real-world ones and a congestion control algorithm that provides an effective trade-off between fairness and efficiency.

6.1 Performance evaluation through simulation

Over the past decade a diverse set of routing algorithms has been proposed in the literature of opportunistic networks. Building large scale testbeds for evaluating these algorithms is hard since a large number of devices and human participants are required. Furthermore, the absence of a killer application for opportunistic networks [131] render the deployment of real-world testbeds even more difficult. For these reasons, the research community focused on simulation for evaluating the performance of the proposed routing algorithms.

The initial simulators used for evaluating routing algorithms for OppNets where the ones used for simulating networking protocols in MANETs. The most common simulation tools of this category, e.g., ns2 [96], use synthetic mobility models to generate the node movement. These tools were gradually replaced when the first real-world experiments on exploring the human interactions appeared. During these experiments, special devices are distributed among students/faculty or conference attendees and monitor the interactions between them. The contacts between the participants along with their corresponding duration constitute what is known as a real-world trace. Since existing tools (used for simulating MANETs) are not capable to process real-world traces, researchers turned to non-generic custom simulators. However, this approach received a lot of criticism since reproducing and comparing the results from different custom simulators is very difficult. Furthermore, in most cases custom simulators are limited to specific algorithms and real-world traces. A simulation tool that gained popularity and used in many works over the last years is ONE [132]. This simulator, implemented in Java, is able to generate and simulate node movement using various synthetic movement models. Furthermore, it can import mobility data from real-world traces or other mobility generators. However, transforming real-word traces to a format compatible with ONE is left to the user. As a result, each work in the literature that utilizes ONE for evaluation purposes uses its own method for importing real-word traces. In most occasions, information about the conversion method used is absent. This complicates the process of reproducing the simulation results.

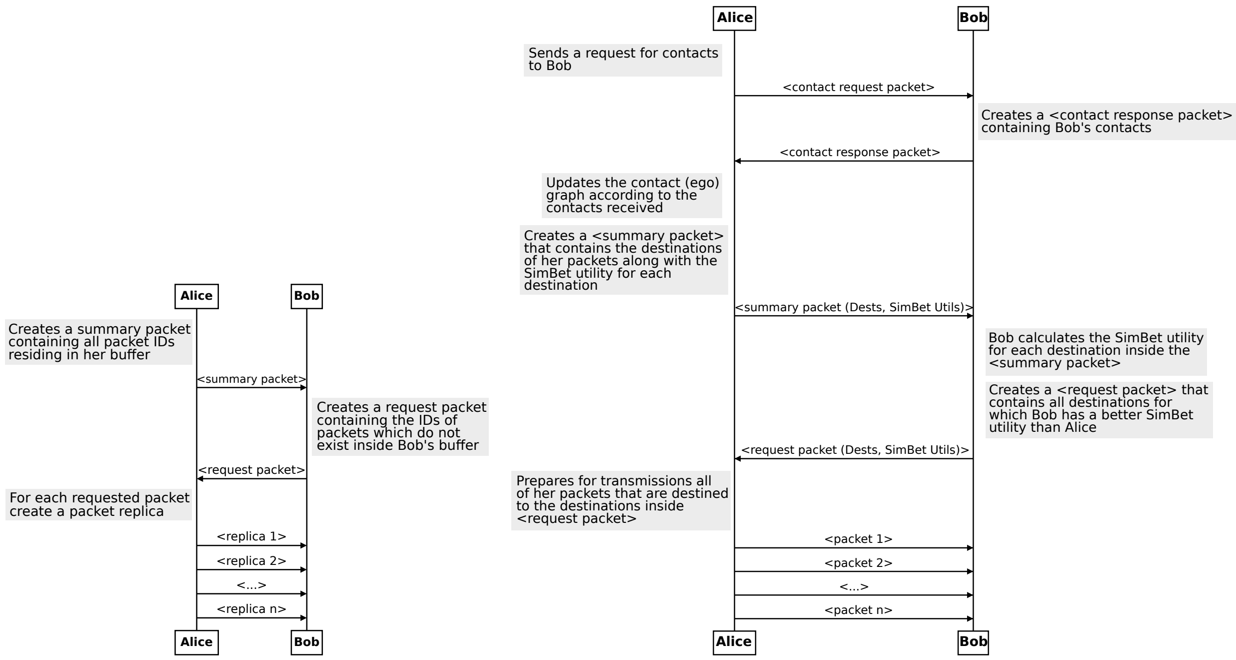
Following a different approach from the existing tools, we implemented Adyton [118], a new simulation framework. Adyton is specially designed for evaluating networking algorithms using real-world traces. Instead of relying on node move-

Table 6.1: Real-world traces supported by Adyton

Trace Name	Environment	Participants	Duration (days)	Type
Intel [121]	lab	9	4.15	encounters
Cambridge haggle [120,121]	lab	12	5.27	encounters
Infocom ’05 [121,133]	conference	41	2.94	encounters
Infocom ’06 [121]	conference	98	3.91	encounters
Sigcomm ’09 [122,123]	conference	76	3.71	encounters
Lyon [134,135]	school	242	1.35	encounters
MIT Reality [124,125]	campus	97	282.74	encounters
Milano pmtr [127,136]	campus	44	18.90	encounters
UCSD [137]	campus	266	78	AP-based
Dartmouth [138,139]	campus	739	119	AP-based
Nodobo [140,141]	campus	27	143.89	encounters
SASSY [142,143]	campus	25	74.23	encounters
Cambridge upmc [129,144]	city	52	11.43	encounters
Rollernet [145,146]	city	62	0.12	encounters
DieselNet [147,148]	city (buses)	37	123.1	encounters
Cabspotting [149,150]	city (taxis)	536	23.97	GPS-based

ments, Adyton operates directly on node contacts provided in almost all real-world traces. For each contact in the trace Adyton simulates the communication among the participating nodes and applies the forwarding/replication decisions of the corresponding routing protocol. Since real-world traces have different formats, Adyton provides the necessary tools to process the original traces and transform them in a common compatible format. Table 6.1 summarizes the real-world traces currently supported by Adyton. We gathered and processed several real-world traces of different scale ranging from lab-level to city-level opportunistic environments. Moreover, Adyton is able to operate on a wide range of synthetic traces as well. More specifically, it supports traces produced by BonnMotion [151] which is a well-known tool for creating and analyzing synthetic mobility scenarios proposed in the literature.

As previously mentioned, Adyton operates on a contact-basis which facilitates an implementation based on the well known event-driven simulation model. This model is known to be faster and provide more accurate results compared to time-



(a) (b)

Figure 6.1: Packets exchanged during a node contact between Alice and Bob when Epidemic routing [1] is utilized.

driven simulation. We followed the same approach as most of the existing simulation tools for opportunistic networks focusing on the network layer and abstracting the physical and MAC layers. Furthermore, in the current version of Adyton, we assume that, during a contact, nodes are able to exchange an unlimited number of packets.

In contrast to existing simulation tools, Adyton captures more than the data packets exchanged among the network nodes. It also simulates the extra communication required for sharing vital routing information, e.g., nodes' utility values. This type of communication is simulated using control packets which are exchanged during a contact and before any data transmission takes place. The number of control packets as well as their contents depend on the corresponding routing protocol. For example, Fig. 6.1 depicts two sequence diagrams illustrating the packets exchanged during a contact when Epidemic routing [1] and SimBet [79] protocols is utilized, respectively. Before any data packet transmission, both protocols exchange additional routing information. In Epidemic routing (Fig. 6.1a), nodes exchange two control packets which incarnate what is known as anti-entropy session. On the other hand, SimBet (Fig. 6.1b) requires twice more control packets than Epidemic routing. The first half updates the contact graph of the node that initiated the contact, i.e., Alice.

Table 6.2: Routing protocols implemented in Adyton

Routing protocol	Strategy	Utility-based
Direct [88]	single-copy	–
Epidemic [1]	multi-copy	–
PRoPHET [81,82,152]	single-copy	✓
SimBet [79]	single-copy	✓
SimBetTS [33]	multi-copy	✓
Bubble Rap [153,154]	single-copy	✓
Spray and Wait [31]	multi-copy	–
LSF Spray and Wait [32]	multi-copy	✓
MSF Spray and Wait [32]	multi-copy	✓
LSF Spray and Focus [35]	multi-copy	✓
Encounter-Based Routing [34]	multi-copy	✓
Delegation Forwarding [36]	multi-copy	✓
Coordinated Delegation Forwarding [155]	multi-copy	✓
Compare and Replicate	multi-copy	✓
PRoPHET Spray and Wait	multi-copy	✓

The second half is used for advertising Alice's packets along with their corresponding utility values that allows Bob to request the packets for which is better forwarder than Alice. It is clear that capturing the overhead introduced by control packets is important since this overhead is protocol-specific. Table 6.2 summarizes the routing protocols currently implemented in Adyton. In contrast to the existing tools, Adyton includes a plethora of routing algorithms composing of both single-copy and multi-copy schemes. Furthermore, Adyton's design facilitates the incorporation of new routing algorithms providing the means to easily reuse existing code, e.g., implementation of utility metrics.

Besides selecting a real-world trace and a routing protocol the user can further customize the simulation by changing the traffic generation scheme and setting the storage capacity of the network nodes. For storage capacity of finite size Adyton provides further options for customization. More specifically, the user can select among a list of drop policies [156,157] that define which packet is dropped when a node's buffer gets full. Also, the user is able to set a scheduling policy [157,158] which determines the transmission sequence of data packets at the sending nodes. Moreover, Adyton offers a diverse range of congestion control mechanisms[1] [159–163] that have been proposed in the literature. In case of multi-copy algorithms, Adyton supports different deletion mechanisms, i.e., the mechanism for deleting the packet replicas in the network when one of them reaches the destination. The implemented deletion mechanisms include multiple variations of the TTL approach and the VACCINE scheme [164].

Adyton's source code is publicly available under GPLv3 on GitHub[2]. Currently, we work towards lifting some of the assumptions made during the initial design, e.g., allow the user to adjust the sending capabilities (data rate) of the network nodes.

6.2 Large scale synthetic traces

Real-world traces are extensively used in the field of opportunistic networks. They are valuable tools for capturing and analyzing the social ties between the participants that in most cases are humans. Furthermore, they are widely used for evaluating the performance of networking mechanisms. As a result, real-world traces significantly influence the design of protocols. However, most real-world traces are of small scale. As a result, an open research problem is to produce large scale traces of human contacts in order to assess the scalability of the algorithms that provide communication in opportunistic networks.

Towards this direction, the research community focused on generating synthetic traces [165] based on modelling user mobility. All proposed schemes model mobility as a result of human activities and social ties. In order to increase realism, recent approaches target at inheriting certain characteristics that appear in real-world traces [166,167]. The synthetic traces produced in this way exhibit a power law with an exponential decay dichotomy distribution of inter-contact periods, similar to the one observed in real traces [168], as well as a similar distribution of contact duration.

Although reproducing certain characteristics that appear in real-world traces is essential, there are additional characteristics of equal importance that should be care-

[1] In the current version of Adyton some of the scheduling policies and congestion control mechanisms are supported only by single-copy routing protocols. Their use in multi-copy protocols is still experimental.

[2] https://github.com/npapanik/Adyton

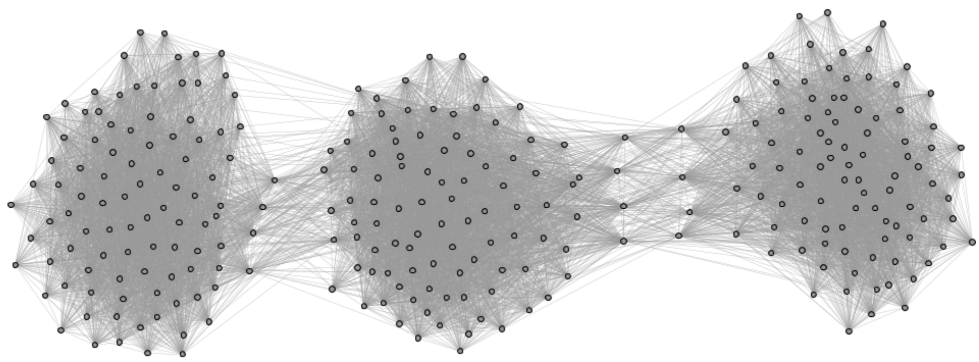


Figure 6.2: Visualized example of CrossWorld output using three building blocks based on MIT Reality trace

fully examined when generating large scale synthetic traces. Motivated by research efforts that reveal the “small world” phenomenon when examining the shortest path in real-world traces [169–171], we focus on the degree of separation between network nodes. This is a structural network characteristic which is defined as the minimum number of hops required for delivering a message from one node to another. After statistically analyzing the hop count of the minimum hop path in various real-world traces, we observed that all traces exhibit a low degree of separation, i.e., starting from a node all other nodes can be reached within two hops in most cases. We attribute this finding to the nature and characteristics of the experimental networks used to collect the real-word traces. In all examined cases the experiment for collecting the human contacts: (i) takes place in a limited geographical area, e.g., university campus, and (ii) the participants are related at least by a “loose” relationship, e.g., some type of enrolment in the same campus. However, it is reasonable to assume that large scale networks will not simply involve more nodes but also larger geographical areas and nodes will not share any kind of social relationship or be geographically co-located. In fact, the statistical analysis of large scale location traces of mobile users has confirmed that individuals live and travel in different confined regions [172]. Therefore, it is reasonable to expect that large scale opportunistic networks will exhibit a larger degree of separation.

Current synthetic mobility models are not designated to produce large scale traces with increased degree of separation. Therefore, we designed a new paradigm tailored for this task, called CrossWorld [173]. Instead of modelling user mobility, CrossWorld uses existing real-world traces as the building blocks of a larger network. Each block represents a network that evolves in a specific geographic area and its users are related at least by a “loose” relationship, e.g., people enrolled in a campus. Some users have multiple enrolments in different small scale networks, being in this way the “glue” that brings together the building blocks. The building blocks in our model can be seen as groups of users that move in a confined region and happen to be located in the same area and/or share some kind of social relationship [172]. At the same time, users with multiple enrolments may be associated with individuals moving in larger regions, therefore travelling over longer distances and producing contacts with nodes in different building blocks. Fig. 6.2 visualizes an example of a CrossWorld trace consisting of three building blocks in a linear arrangement. In this example, each building block is based on the MIT Reality real-world trace [124, 125] resulting in an opportunistic network composed of 291 nodes.

CrossWorld produces large scale synthetic traces that preserve a set of attributes that have been witnessed in the real-world traces. More specifically, the generated traces exhibit: (i) a high clustering coefficient, (ii) a power law distribution of inter-contact time with an exponential decay dichotomy, and (iii) a distribution of contact duration similar to that of real-world traces. However, the most important feature of CrossWorld is its capability to produce large scale traces with a configurable degree of separation. Moving a step further, we set up an experimental study assessing the performance of state-of-the-art routing protocols under large scale synthetic traces with varying degree of separation produced by CrossWorld. Our experimental results emphasize the need for designing new routing approaches for large scale opportunistic networks. More specifically, we reveal that current schemes face significant performance challenges when the degree of separation increases.

6.3 The need for congestion control

The prominent routing approach in Opportunistic networks is utility-based routing. Network nodes exploit the contacts that occur between them in order to make greedy decisions based on a utility metric that captures the fitness of each node for delivering and forwarding packets to their destinations. Despite its efficiency, this approach is known to cause the phenomenon of highly unbalanced loading of the nodes [174]. This results in a small subset of nodes handling most of the traffic load in the whole network. In the case of a realistic opportunistic network, where the resources of nodes are limited, the aforementioned phenomenon has a severe impact on routing performance because it induces a large number of packet drops. As a result, one of the challenges in opportunistic networks is to design congestion control mechanisms that prevent packet drops and leave the routing efficiency unaffected.

Several approaches that aim to minimize the performance degradation due to congestion have been proposed [175]. These approaches can be classified into those applicable to multi-copy routing protocols and those tailored for single-copy ones. The approaches in the first class benefit from the existence of multiple packet replicas that mitigate the effects of packet drops to some extent. In the latter case, the problem of congestion control is more challenging since a single packet drop has an immediate impact on the delivery efficiency. Two types of single-copy congestion control approaches can be identified: (i) those that aim at enhancing fairness [160], and (ii) those that aim at avoiding storage congestion and strive for a better delivery performance [159]. The rationale in the first category is that energy is a limited resource therefore fairness is required in using the nodes as relays. However, fairness is achieved at the expense of delay. The idea behind the second category is to exploit alternative paths in order to bypass the area of congestion. However, this approach is usually poor in fairness and at the same time induces an increased cost in terms of transmissions in order to reduce delay and vice versa.

Motivated by the observation that a profitable trade-off between fairness and efficiency can be achieved, we devise a novel congestion control mechanism that maximizes the delivery efficiency while achieving a low end-to-end delay at the expense of a reasonably low cutback in fairness. The proposed algorithm called Congestion Control with Adjustable Fairness (CCAF) [163] incarnates the described functionality in a generic manner that can be incorporated into virtually any utility-based routing protocol. CCAF provides, through a tunable parameter, a trade-off between fairness and end-to-end delay without impacting and in some cases improving the delivery ratio. Furthermore, we provide a method for dynamically adjusting the tunable parameter based on the social ties among the network nodes. Each network node acts as a relay and offers its resources to carry packets destined to its friends rather than strangers. Delivering a packet to a friend is easier than delivering it to a stranger resulting in nodes carrying packets for smaller amounts of time and significantly improving the routing performance. Our approach is also suitable for cases of social selfishness in opportunistic networks [176] where nodes are willing to exchange packets only with those whom they have social relationships. We validated the effectiveness of CCAF through simulations in a wide range real-world traces using SimBetTS as the underlying routing protocol. In all cases, the overall routing performance is significantly improved. At the same time, CCAF exhibits a varying level of fairness that depends on the network traffic. More specifically, CCAF maximizes fairness in cases it is needed the most, i.e, when the offered network load increases.

In our ongoing work, we are investigating other methods to tune the trade-off between fairness and routing performance based on local and network-wide information. Furthermore, we examine possible ways to extend CCAF in order to work in synergy with multi-copy routing algorithms.

6.4 Summary

We examined a series of heterogeneous topics in opportunistic networks closely related to the context of this thesis. From realistic simulation and large scale synthetic traces to the need for congestion control, we discussed the open research problems and provided efficient solutions.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

One of the main challenges in wireless ad hoc networks is designing energy efficient networking protocols. In this thesis, we focused on two well-known classes of wireless ad hoc networks; MANETs and OppNets. In both classes, there are situations where fair communication is only provided through protocols that spread multiple packet instances across the network. This approach is extensively used in efficient broadcasting in MANETs and routing in OppNets. In both cases, we investigated the impact of using multiple packet duplicates on the energy efficiency of networking algorithms. Furthermore, we introduced novel schemes that target at minimizing the energy efficiency through the reduction of the number of transmissions (replications), without impacting the overall performance.

In the field of broadcasting in MANETs, we examined energy-efficient schemes that combine traditional algorithms with network coding. Initially, we focused on XOR-based broadcasting where we: (i) fixed a performance issue due to the incompatibility between network coding and the underlying broadcast algorithm, (ii) addressed some implementation concerns of network coding, and (iii) demonstrated how XOR-based coding can be used towards reducing delay. We demonstrated through extensive experiments that the common approach of applying XOR-based network coding over conventional broadcasting suffers from performance breakdowns. To tackle the problem, we provided a solution proposing a novel termination criterion for the underlying broadcasting algorithm that is fully compatible with XOR-based network coding. Moving a step further, we designed a XOR-based broadcast algorithm that, in addition to the new termination criterion, utilizes a lightweight method for detecting coding opportunities. The proposed algorithm also exploits information originally used only for coding purposes to reduce the number of transmissions. Finally, we introduced the concept of "Coded Redundancy" that, in contrast to existing approaches, uses XOR-based network coding towards reducing the end-to-end delay without compromising the overall delivery and energy efficiency.

Along with XOR-based coding approaches, we investigated energy efficient broadcast schemes that utilize random linear network coding (RLNC). Our key contribution in this field is that, for the first time, we combined the resilience of RLNC with the pruning efficiency of CDS-based broadcasting. We developed an analytical model that highlights RLNC's increased resilience to packet losses compared to XOR-based coding. However, our model also revealed that the common approach in the literature which is to apply random linear coding on top of probabilistic forwarding impairs the performance of coding. To this end, we devised a novel RLNC-based broadcast algorithm that combines RLNC with a deterministic underlying scheme specially designed to prune transmissions. In order to increase reliability in poorly connected nodes, we proposed an extension of the basic algorithm that enhances the topology-awareness. Moreover, we provided a practical distributed scheme for managing packet generations that is key issue in RLNC-based broadcasting when inter-source coding is used. Through extensive experiments, we demonstrated that our proposed algorithms significantly reduce the energy costs of broadcasting while at the same time achieve a better and faster delivery performance compared to state-of-the-art.

In the second part, we examined routing in OppNets focusing on replication-based schemes that use multiple packet instances to increase routing efficiency. Motivated by approaches that perform packet replication dynamically using the utility values of carrier nodes, we proposed a novel routing scheme that significantly enhances the energy efficiency at negligible cost and without compromising the overall routing performance compared to the state-of-the-art algorithms. To accomplish this, our approach allows carrier nodes to coordinate their views regarding the replication process and make forwarding/replication decisions accordingly. Furthermore, we provided an extension that further increases the energy efficiency by allowing non-carrier nodes to play a more active role in the replication process. Each non-carrier node refrains from receiving replicas for which it was previously rejected. For the proposed extension, we presented a lightweight implementation based on bloom filters tailored for opportunistic environments where the storage capacity of network nodes is limited. Finally, we demonstrated the performance gains of our solutions through analysis as well as extensive simulations on real-world opportunistic traces.

We also provided solutions to some additional issues that arise when researching OppNets and are closely related to the context of this thesis. In particular, we implemented an event-driven simulator for evaluating the performance of routing schemes under a diverse range real-world and synthetic opportunistic traces. Furthermore, we developed a new paradigm for constructing large scale synthetic traces that exhibit a configurable degree of separation and similar characteristics with real-world traces. Finally, we highlighted the need for congestion control when routing in OppNets with sparse resources and devised a novel congestion control algorithm that can be easily incorporated into virtually any single-copy utility-based algorithm.

7.2 Future work

In the following, we discuss some short-term and long-term perspectives for improving the energy efficiency in wireless ad hoc networks.

In chapters 3 and 4, we demonstrated the ability of network coding to significantly enhance the energy efficiency of broadcasting in MANETS. However, applying network coding requires additional node resources in terms of processing and storage space. That is true regardless of the coding technique utilized. Although we have already examined the aforementioned impact to some extent, there is still work to be done. This is because such a task strongly depends on the implementation specifics of the network coding techniques that cannot be captured through simulation. Therefore, we plan to examine practical ways to deploy the proposed schemes on real hardware and create a testbed for experimentation. Furthermore, we plan to investigate the viability of coding-based broadcast schemes that utilize XOR and RLNC simultaneously. XOR coding can be used for pruning transmissions while RLNC can provide resilience to packet loss. Towards this direction, we envision a coding scheme where XOR operates over RLNC allowing intermediate nodes to take advantage of local coding opportunities by XORing RLNC packets in a hop-by-hop fashion.

In the field of routing in OppNets we have already emphasized the need for designing new routing strategies. Through extensive simulations on large scale synthetic traces produced with CrossWorld (section 6.2), we revealed that state-of-the-art routing algorithms exhibit a systematic performance degradation when the degree of separation increases. Despite its effectiveness on real-world cases, COORD (chapter 5), also faces the same performance challenges. A promising direction towards coping with such cases is to utilize multiple utility metrics in the routing process. This approach is suitable for large scale traces comprised of nodes that tend to form distinct communities. Destination dependent utilities can be used to deliver packets inside a community, while destination independent metrics can be utilized for inter-community communication. Furthermore, we plan to investigate how other network mechanisms affect the routing efficiency in OppNets. Typical examples include packet scheduling policies, buffer management algorithms and congestion control mechanisms. Towards this direction, we intend to expand the Adyton simulator (section 6.1) that we recently released under GPLv3.

A long-term perspective lies in designing XOR-based routing algorithms for OppNets. Similar to RLNC [177–179], XOR coding can be utilized to enhance the routing performance in OppNets. However, incorporating XOR coding into routing in OppNets is a challenging task. In most cases, nodes come into contact in pairs due to intermittent connectivity. As a result, exploiting packet overhearing that is essential for applying traditional XOR coding is not possible. We plan to explore alternative ways to utilize XOR coding for enhancing the routing performance in OppNets. An interesting idea is to use XOR coding towards economizing on storage space and reducing the packet drops. Assuming that senders temporarily store the packets that generate, intermediate nodes can exploit coding opportunities to XOR packets together in order to save buffer space. Then, by using the same routing principles the encoded packets can be forwarded to the corresponding destinations to be decoded.

Bibliography

[1] A. Vahdat, D. Becker, *et al.*, "Epidemic routing for partially connected ad hoc networks," in *Duke University*, CS-200006,2000.

[2] D. Miorandi, S. Sicari, F. D. Pellegrini, and I. Chlamtac, "Internet of things: Vision, applications and research challenges," *Ad Hoc Networks*, vol. 10, no. 7, pp. 1497 – 1516, 2012.

[3] D. G. Reina, S. L. Toral, F. Barrero, N. Bessis, and E. Asimakopoulou, "The role of ad hoc networks in the internet of things: A case scenario for smart environments," in *Internet of Things and Inter-cooperative Computational Technologies for Collective Intelligence* (N. Bessis, F. Xhafa, D. Varvarigou, R. Hill, and M. Li, eds.), pp. 89–113, Berlin, Heidelberg: Springer Berlin Heidelberg, 2013.

[4] P. Bellavista, G. Cardone, A. Corradi, and L. Foschini, "Convergence of manet and wsn in iot urban scenarios," *IEEE Sensors Journal*, vol. 13, no. 10, pp. 3558–3567, 2013.

[5] G. Fanti, Y. B. David, S. Benthall, E. Brewer, and S. Shenker, "Rangzen: Circumventing government-imposed communication blackouts," *University of California, Berkeley, Tech. Rep. UCB/EECS-2013-128*, 2013.

[6] F. Rebecchi, M. D. de Amorim, V. Conan, A. Passarella, R. Bruno, and M. Conti, "Data offloading techniques in cellular networks: A survey," *IEEE Communications Surveys Tutorials*, vol. 17, no. 2, pp. 580–603, 2015.

[7] M. Conti and S. Giordano, "Mobile ad hoc networking: milestones, challenges, and new research directions," *IEEE Communications Magazine*, vol. 52, no. 1, pp. 85–96, 2014.

[8] V. F. Mota, F. D. Cunha, D. F. Macedo, J. M. Nogueira, and A. A. Loureiro, "Protocols, mobility models and tools in opportunistic networks: A survey," *Computer Communications*, vol. 48, pp. 5 – 19, 2014.

[9] B. Williams and T. Camp, "Comparison of Broadcasting Techniques for Mobile Ad Hoc Networks," in *ACM Proc. Int. Symp. Mobile Ad Hoc Networking & Computing (MobiHoc)*, pp. 194–205, 2002.

[10] P. Ruiz and P. Bouvry, "Survey on Broadcast Algorithms for Mobile Ad Hoc Networks," *ACM Comput. Surv.*, vol. 48, no. 1, pp. 8:1–8:35, 2015.

[11] I. Stojmenovic and J. Wu, "Broadcasting and activity scheduling in ad hoc networks," in *Mobile Ad Hoc Networking* (S. Basagni, M. Conti, S. Giordano, and I. Stojmenovic, eds.), ch. 7, pp. 205–229, Wiley-IEEE Press, 2005.

[12] O. Liang, Y. A. Sekercioglu, and N. Mani, "A survey of multipoint relay based broadcast schemes in wireless ad hoc networks," *IEEE Communications Surveys & Tutorials*, vol. 8, no. 4, pp. 30–46, 2006.

[13] D. Reina, S. Toral, P. Johnson, and F. Barrero, "A survey on probabilistic broadcast schemes for wireless ad hoc networks," *Ad Hoc Networks*, vol. 25, Part A, pp. 263 – 292, 2015.

[14] M. Abolhasan, T. Wysocki, and E. Dutkiewicz, "A review of routing protocols for mobile ad hoc networks," *Elsevier Ad hoc Networks*, vol. 2, no. 1, pp. 1–22, 2004.

[15] A. Boukerche, B. Turgut, N. Aydin, M. Z. Ahmad, L. Bölöni, and D. Turgut, "Routing protocols in ad hoc networks: A survey," *Computer Networks*, vol. 55, no. 13, pp. 3032 – 3080, 2011.

[16] A. N. Mian, R. Baldoni, and R. Beraldi, "A survey of service discovery protocols in multihop mobile ad hoc networks," *IEEE Pervasive Computing*, vol. 8, no. 1, pp. 66–74, 2009.

[17] C. Ververidis and G. Polyzos, "Service discovery for mobile Ad Hoc networks: a survey of issues and techniques," *IEEE Communications Surveys Tutorials*, vol. 10, no. 3, pp. 30–45, 2008.

[18] Y. Luo and O. Wolfson, "Mobile P2P Databases," in *Encyclopedia of GIS*, pp. 671–677, Springer US, 2008.

[19] R. Ahlswede, N. Cai, S.-Y. Li, and R. Yeung, "Network information flow," *IEEE Trans. Information Theory*, vol. 46, no. 4, pp. 1204–1216, 2000.

[20] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft, "Xors in the air: Practical wireless network coding," *IEEE/ACM Trans. on Networking*, vol. 16, no. 3, pp. 497–510, 2008.

[21] L. Li, R. Ramjee, M. Buddhikot, and S. Miller, "Network Coding-Based Broadcast in Mobile Ad-hoc Networks," in *Proc. IEEE Int. Conf. Computer Communications (INFOCOM)*, pp. 1739–1747, 2007.

[22] P. A. Chou, Y. Wu, and K. Jain, "Practical network coding," in *Proc. Allerton Conf. Communication, Control and Computing*, pp. 40–49, 2003.

[23] C. Fragouli, J. Widmer, and J.-Y. Le Boudec, "Efficient Broadcasting Using Network Coding," *IEEE/ACM Trans. Networking*, vol. 16, no. 2, pp. 450–463, 2008.

[24] J. Yu, N. Wang, G. Wang, and D. Yu, "Connected dominating sets in wireless ad hoc and sensor networks – A comprehensive survey," *Elsevier Computer Communications*, vol. 36, no. 2, pp. 121–134, 2013.

[25] A. Qayyum, L. Viennot, and A. Laouiti, "Multipoint relaying for flooding broadcast messages in mobile wireless networks," in *Proc. Annu. Hawaii Int. Conf. System Sciences*, pp. 3866–3875, 2002.

[26] H. Lim and C. Kim, "Flooding in wireless ad hoc networks," *Elsevier Computer Communications*, vol. 24, no. 3–4, pp. 353–363, 2001.

[27] W. Lou and J. Wu, "On reducing broadcast redundancy in ad hoc wireless networks," *IEEE Trans. Mobile Computing*, vol. 1, no. 2, pp. 111–122, 2002.

[28] S.-Y. Li, R. Yeung, and N. Cai, "Linear network coding," *IEEE Trans. Information Theory*, vol. 49, no. 2, pp. 371–381, 2003.

[29] T. Ho, R. Koetter, M. Medard, D. Karger, and M. Effros, "The benefits of coding over routing in a randomized setting," in *IEEE Proc. Int. Symp. Information Theory*, p. 442, 2003.

[30] Z. Zhang, "Routing in intermittently connected mobile ad hoc networks and delay tolerant networks: overview and challenges," *IEEE Communications Surveys Tutorials*, vol. 8, no. 1, pp. 24–37, 2006.

[31] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Efficient routing in intermittently connected mobile networks: the multiple-copy case," *IEEE/ACM Trans. Netw.*, vol. 16, no. 1, pp. 77–90, 2008.

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

[33] E. M. Daly and M. Haahr, "Social network analysis for information flow in disconnected delay-tolerant MANETs," *IEEE Trans. Mobile Comput.*, vol. 8, no. 5, pp. 606–621, 2009.

[34] S. C. Nelson, M. Bakht, and R. Kravets, "Encounter-based routing in DTNs," in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, pp. 846–854, 2009.

[35] T. Spyropoulos, K. Psounis, and C. Raghavendra, "Efficient routing in intermittently connected mobile networks: The multiple-copy case," *IEEE Trans. Netw.*, vol. 16, no. 1, pp. 77–90, 2008.

[36] V. Erramilli, M. Crovella, A. Chaintreau, and C. Diot, "Delegation Forwarding," in *Proc. ACM Int. Conf. Mobile Ad Hoc Netw. and Comp. (MobiHoc)*, pp. 251–260, 2008.

[37] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and J.-P. Sheu, "The broadcast storm problem in a mobile ad hoc network," in *ACM/IEEE Proc. Int. Conf. Mobile Computing and Networking*, pp. 151–162, 1999.

[38] J. Wu and H. Li, "On calculating connected dominating set for efficient routing in ad hoc wireless networks," in *ACM Proc. Int. Workshop Discrete Algorithms and Methods for Mobile Computing and Communications*, pp. 7–14, 1999.

[39] F. Dai and J. Wu, "Distributed dominant pruning in ad hoc networks," in *IEEE Int. Conf. Communications (ICC)*, pp. 353–357, 2003.

[40] J. Wu, F. Dai, and S. Yang, "Iterative local solutions for connected dominating set in ad hoc wireless networks," *IEEE Trans. on Computers*, vol. 57, pp. 702–715, 2008.

[41] C. Adjih, P. Jacquet, and L. Viennot, "Computing connected dominated sets with multipoint relays," *Ad Hoc and Sensor Wireless Networks, Old City Publishing*, vol. 1, no. 1-2, pp. 27–39, 2005.

[42] J. Wu, "An enhanced approach to determine a small forward node set based on multipoint relays," in *IEEE Vehicular Technology Conf. (VTC)*, vol. 4, pp. 2774–2777, 2003.

[43] J. Wu, W. Lou, and F. Dai, "Extended multipoint relays to determine connected dominating sets in manets," *IEEE Trans. on Computers*, vol. 55, no. 3, pp. 334–347, 2006.

[44] X. Chen and J. Shen, "Reducing connected dominating set size with multipoint relays in ad hoc wireless networks," in *IEEE Proc. Int. Symp. Parallel Architectures, Algorithms and Networks*, pp. 539–543, 2004.

[45] W. Peng and X.-C. Lu, "On the reduction of broadcast redundancy in mobile ad hoc networks," in *IEEE Annual Workshop Mobile and Ad Hoc Networking and Computing*, pp. 129–130, 2000.

[46] I. Stojmenovic, M. Seddigh, and J. Zunic, "Dominating sets and neighbor elimination-based broadcasting algorithms in wireless networks," *IEEE Trans. Parallel Distributed Systems*, vol. 13, no. 1, pp. 14–25, 2002.

[47] I. Stojmenovic, "Comments and corrections to "Dominating sets and neighbor elimination-based broadcasting algorithms in wireless networks"," *IEEE Trans. Parallel Distributed Systems*, vol. 15, no. 11, pp. 1054–1055, 2004.

[48] A. A. Khan, I. Stojmenovic, and N. Zaguia, "Parameterless broadcasting in static to highly mobile wireless ad hoc, sensor and actuator networks," in *IEEE Int. Conf. Advanced Information Networking and Applications*, pp. 620–627, 2008.

[49] F. J. Ros, P. M. Ruiz, and I. Stojmenovic, "Acknowledgment-based broadcast protocol for reliable and efficient data dissemination in vehicular ad hoc networks," *IEEE Trans. Mobile Computing*, vol. 11, no. 1, pp. 33–46, 2012.

[50] D. Nguyen and P. Minet, "Analysis of MPR Selection in the OLSR Protocol," in *IEEE Int. Conf. Adv. Information Networking and Applications Workshops (AINAW)*, vol. 2, pp. 887–892, 2007.

[51] F. J. Ros and P. M. Ruiz, "Minimum broadcasting structure for optimal data dissemination in vehicular networks," *IEEE Trans. Vehicular Technology*, vol. 62, no. 8, pp. 3964–3973, 2013.

[52] J. H. Ahn and T. J. Lee, "A multipoint relay selection method for reliable broadcast in ad hoc networks," in *IEEE Int. Conf. ICT Convergence (ICTC)*, pp. 616–617, 2011.

[53] M. Khabbazian, I. F. Blake, and V. K. Bhargava, "Local broadcast algorithms in wireless ad hoc networks: Reducing the number of transmissions," *IEEE Trans. Mobile Computing*, vol. 11, no. 3, pp. 402–413, 2012.

[54] W. Lou and J. Wu, "Double-covered broadcast (DCB): a simple reliable broadcast algorithm in manets," in *Proc. IEEE Int. Conf. Computer Communications (INFOCOM)*, vol. 3, pp. 2084–2095, 2004.

[55] S. Wang, A. Vasilakos, H. Jiang, X. Ma, W. Liu, K. Peng, B. Liu, and Y. Dong, "Energy Efficient Broadcasting Using Network Coding Aware Protocol in Wireless Ad Hoc Network," in *IEEE Int. Conf. Communications (ICC)*, pp. 1–5, 2011.

[56] S. Wang, C. Wang, K. Peng, G. Tan, H. Jiang, and Y. Dong, "Network coding over connected dominating set: energy minimal broadcasting in wireless ad hoc networks," *Springer Wireless Networks*, vol. 20, no. 5, pp. 1023–1036, 2013.

[57] T. Kunz and L. Li, "Broadcasting in multihop mobile tactical networks: To network code or not," in *ACM Proc. Int. Wireless Communications and Mobile Computing Conf.*, pp. 676–680, 2010.

[58] P. Ostovari, J. Wu, and A. Khreishah, "Deadline-aware broadcasting in wireless networks with local network coding," in *Proc. IEEE Int. Conf. Computing, Networking and Communications (ICNC)*, pp. 1123–1127, 2012.

[59] S. Yang and J. Wu, "Efficient Broadcasting Using Network Coding and Directional Antennas in MANETs," *IEEE Trans. Parallel and Distributed Systems*, vol. 21, no. 2, pp. 148–161, 2010.

[60] J. Widmer and J.-Y. Le Boudec, "Network Coding for Efficient Communication in Extreme Networks," in *Proc. ACM SIGCOMM Workshop Delay-tolerant Networking (WDTN)*, pp. 284–291, 2005.

[61] T. Kunz, K. Mahmood, and A. Matrawy, "Broadcasting in multihop wireless networks: The case for multi-source network coding," in *Proc. IEEE Int. Conf. Communications (ICC)*, pp. 6679–6684, 2012.

[62] N. Papanikos and E. Papapetrou, "Deterministic broadcasting and random linear network coding in mobile ad hoc networks," *Computing Research Repository (CoRR)*, vol. arXiv:1411.4465, 2014.

[63] N. Kadi and K. Agha, "Network coding for MPR-based flooding in ad hoc wireless networks under mobility conditions," *Springer-Verlag Annals Telecommunications*, vol. 66, no. 5-6, pp. 319–329, 2011.

[64] S. Wang, G. Tan, Y. Liu, H. Jiang, and T. He, "Coding Opportunity Aware Backbone Metrics for Broadcast in Wireless Networks," *IEEE Trans. Parallel and Distributed Systems*, vol. 25, pp. 1999–2009, Aug 2014.

[65] N. Rahnavard, B. N. Vellambi, and F. Fekri, "CRBcast: a reliable and energy-efficient broadcast scheme for wireless sensor networks using rateless codes," *IEEE Trans. Wireless Communications*, vol. 7, no. 12, pp. 5390–5400, 2008.

[66] B. Vellambi, N. Rahnavard, and F. Fekri, "FTS: A distributed energy-efficient broadcasting scheme using fountain codes for multihop wireless networks," *IEEE Trans. Communications*, vol. 58, no. 12, pp. 3561–3572, 2010.

[67] A. Asterjadhi, E. Fasolo, M. Rossi, J. Widmer, and M. Zorzi, "Toward network coding-based protocols for data broadcasting in wireless Ad Hoc networks," *IEEE Trans. Wireless Communications*, vol. 9, no. 2, pp. 662–673, 2010.

[68] I. Hou, Y. Tsai, T. Abdelzaher, and I. Gupta, "Adapcode: Adaptive network coding for code updates in wireless sensor networks," in *Proc. IEEE Int. Computer Communications (INFOCOM)*, pp. 1517–1525, 2008.

[69] S. Y. Cho and C. Adjih, “Wireless Broadcast with Network Coding: Dynamic Rate Selection,” in *Advances in Ad Hoc Networking: Proc. Annual Mediterranean Ad Hoc Networking Workshop*, pp. 191–202, 2008.

[70] Z. Yang, M. Li, and W. Lou, “R-Code: Network coding-based reliable broadcast in wireless mesh networks,” *Elsevier Ad Hoc Networks*, vol. 9, no. 5, pp. 788–798, 2011.

[71] J. Subramanian, R. Morris, and H. Balakrishnan, “UFlood: High-throughput flooding over wireless mesh networks,” in *Proc. IEEE Int. Conf. Computer Communications (INFOCOM)*, pp. 82–90, 2012.

[72] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, “Trading structure for randomness in wireless opportunistic routing,” *ACM SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 4, pp. 169–180, 2007.

[73] D. Koutsonikolas, Y. C. Hu, and C.-C. Wang, “Pacifier: High-throughput, Reliable Multicast Without “Crying Babies” in Wireless Mesh Networks,” *IEEE/ACM Trans. Networking*, vol. 20, no. 5, pp. 1375–1388, 2012.

[74] P. Ostovari, A. Khreishah, and J. Wu, “Deadline-aware broadcasting in wireless networks with network coding,” in *Proc. IEEE Conf. Global Communications (GLOBECOM)*, pp. 4435–4440, 2012.

[75] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott, “Impact of human mobility on opportunistic forwarding algorithms,” *IEEE Transactions on Mobile Computing*, vol. 6, no. 6, pp. 606–620, 2007.

[76] A. Chaintreau, A. Mtibaa, L. Massoulie, and C. Diot, “The diameter of opportunistic mobile networks,” in *ACM Proc. CoNEXT Conf.*, pp. 12:1–12:12, 2007.

[77] S. Gaito, E. Pagani, and G. P. Rossi, “Strangers help friends to communicate in opportunistic networks,” *Computer Networks*, vol. 55, no. 2, pp. 374 – 385, 2011.

[78] V. Erramilli, A. Chaintreau, M. Crovella, and C. Diot, “Diversity of forwarding paths in pocket switched networks,” in *Proc. ACM SIGCOMM Internet Meas. Conf. (IMC)*, pp. 161–174, 2007.

[79] E. M. Daly and M. Haahr, "Social network analysis for routing in disconnected delay-tolerant manets," in *ACM Proc. Int. Symposium Mobile Ad Hoc Netw. and Comput. (MobiHoc)*, pp. 32–40, ACM, 2007.

[80] H. Dubois-Ferriere, M. Grossglauser, and M. Vetterli, "Age matters: efficient route discovery in mobile ad hoc networks using encounter ages," in *Proc. ACM Mobile Ad Hoc Netw. and Comput. (MobiHoc)*, pp. 257–266, 2003.

[81] A. Lindgren, A. Doria, and O. Schelén, "Probabilistic routing in intermittently connected networks," *ACM SIGMOBILE Mobile Comput. and Commun. Rev.*, vol. 7, no. 3, pp. 19–20, 2003.

[82] S. Grasic, E. Davies, A. Lindgren, and A. Doria, "The evolution of a dtn routing protocol - prophetv2," in *Proc. ACM Workshop Challenged Netw. (CHANTS)*, pp. 27–30, 2011.

[83] E. Bulut and B. K. Szymanski, "Friendship based routing in delay tolerant mobile social networks," in *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*, pp. 1–5, IEEE, 2010.

[84] E. Bulut and B. K. Szymanski, "Exploiting friendship relations for efficient routing in mobile social networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 12, pp. 2254–2265, 2012.

[85] P. V. Marsden, "Egocentric and sociocentric measures of network centrality," *Social networks*, vol. 24, no. 4, pp. 407–422, 2002.

[86] L. C. Freeman, "A set of measures of centrality based on betweenness," *Sociometry*, pp. 35–41, 1977.

[87] M. S. Granovetter, "The strength of weak ties," *American journal of sociology*, pp. 1360–1380, 1973.

[88] R. C. Shah, S. Roy, S. Jain, and W. Brunette, "Data mules: Modeling and analysis of a three-tier architecture for sparse sensor networks," *Ad Hoc Networks*, vol. 1, no. 2, pp. 215–233, 2003.

[89] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Efficient routing in intermittently connected mobile networks: The single-copy case," *IEEE/ACM Trans. Netw.*, vol. 16, no. 1, pp. 63–76, 2008.

[90] W. Woon and K. Yeung, "Enhanced Termination Condition for Deterministic Broadcasting Protocols in Mobile Ad Hoc Networks," in *IEEE Vehicular Technology Conference Fall (VTC)*, pp. 1–5, Sept 2010.

[91] L. Zhu, B.-S. Lee, B.-C. Seet, K.-J. Wong, G. Liu, S.-Y. Huang, and K.-K. Lee, "Performance of New Broadcast Forwarding Criteria in MANET," in *Information Networking. Networking Technologies for Broadband and Mobile Networks*, vol. 3090 of *Lecture Notes in Computer Science*, pp. 34–43, Springer Berlin Heidelberg, 2004.

[92] T. H. Cormen, C. Stein, R. L. Rivest, and C. E. Leiserson, *Introduction to Algorithms*. McGraw-Hill Higher Education, 2nd ed., 2001.

[93] M. Baysan, K. Sarac, R. Chandrasekaran, and S. Bereg, "A polynomial time solution to minimum forwarding set problem in wireless networks under unit disk coverage model," *IEEE Trans. Parallel and Distributed Systems*, vol. 20, no. 7, pp. 913–924, 2009.

[94] M. Baysan, K. Sarac, and R. Chandrasekaran, "Polynomial time solution to minimum forwarding set problem in wireless networks under disk coverage model," *Elsevier Ad Hoc Networks*, vol. 10, no. 7, pp. 1253 – 1266, 2012.

[95] S. Agathos and E. Papapetrou, "On the Set Cover Problem for Broadcasting in Wireless Ad Hoc Networks," *IEEE Communications Letters*, vol. 17, no. 11, pp. 2192–2195, 2013.

[96] S. McCanne and S. Floyd, "ns2 - Network Simulator." `http://www.isi.edu/nsnam/ns/`.

[97] C. Bettstetter, G. Resta, and P. Santi, "The node distribution of the random waypoint mobility model for wireless ad hoc networks," *IEEE Trans. Mobile Computing*, vol. 2, no. 3, pp. 257–269, 2003.

[98] J. G. Jetcheva and D. B. Johnson, "Adaptive demand-driven multicast routing in multi-hop wireless ad hoc networks," in *ACM Proc. Int. Symp. Mobile Ad Hoc Netw. and Comput. (MobiHoc)*, pp. 33–44, 2001.

[99] J.-Y. Le Boudec and M. Vojnovic, "Perfect simulation and stationarity of a class of mobility models," in *Proc. IEEE Int. Conf. Computer Communications (INFOCOM)*, pp. 2743–2754, 2005.

[100] Y. Wu, P. Chou, and K. Jain, "A comparison of network coding and tree packing," in *IEEE Proc. Int. Symp. Information Theory*, p. 143, 2004.

[101] M. Penrose, *Random geometric graphs (Oxford Studies in Probability)*, vol. 5. Oxford University Press Oxford, 2003.

[102] P. Nain, D. Towsley, B. Liu, and Z. Liu, "Properties of random direction models," in *Proc. IEEE INFOCOM*, pp. 1897–1907, 2005.

[103] C. Reis, R. Mahajan, M. Rodrig, D. Wetherall, and J. Zahorjan, "Measurement-based models of delivery and interference in static wireless networks," *ACM SIGCOMM Computer Communication Review*, vol. 36, pp. 51–62, Aug. 2006.

[104] Z. J. Haas, J. Y. Halpern, and L. Li, "Gossip-based ad hoc routing," *IEEE/ACM Trans. Networking*, vol. 14, no. 3, pp. 479–491, 2006.

[105] D. A. Levin, Y. Peres, and E. L. Wilmer, *Markov chains and mixing times*. Amer. Math. Soc., 2006.

[106] M. J. Appel and R. P. Russo, "The connectivity of a graph on uniform points on $[0,1]^d$," *Statist. & Probability Lett.*, vol. 60, no. 4, pp. 351–357, 2002.

[107] L. Xie, P. H. Chong, I. W. Ho, and Y. Guan, "A survey of inter-flow network coding in wireless mesh networks with unicast traffic," *Elsevier Computer Networks*, vol. 91, pp. 738 – 751, 2015.

[108] H. Seferoglu, A. Markopoulou, and K. Ramakrishnan, "I^{2}NC: Intra- and inter-session network coding for unicast flows in wireless networks," in *IEEE Proc. INFOCOM*, pp. 1035–1043, 2011.

[109] C. Qin, Y. Xian, C. Gray, N. Santhapuri, and S. Nelakuditi, "I^{2}MIX: Integration of intra-flow and inter-flow wireless network coding," in *IEEE Proc. Annu. Conf. Sensor, Mesh Ad Hoc Commun. Netw. Workshops (SECON)*, pp. 1–6, 2008.

[110] X. Zhu, H. Yue, Y. Fang, and Y. Wang, “A batched network coding scheme for wireless networks,” *Springer Wireless Networks*, vol. 15, no. 8, pp. 1152–1164, 2009.

[111] A. Khreishah, I. Khalil, and J. Wu, “Low complexity and provably efficient algorithm for joint inter and intrasession network coding in wireless networks,” *IEEE Trans. Parallel and Distrib. Syst.*, vol. 24, no. 10, pp. 2015–2024, 2013.

[112] J. Krigslund, J. Hansen, M. Hundeboll, D. E. Lucani, and F. H. P. Fitzek, “CORE: COPE with MORE in wireless meshed networks,” in *IEEE Veh. Technol. Conf. (VTC)*, pp. 1–6, 2013.

[113] J. Sundararajan, D. Shah, M. Medard, S. Jakubczak, M. Mitzenmacher, and J. Barros, “Network coding meets tcp: Theory and implementation,” *Proc. of the IEEE*, vol. 99, no. 3, pp. 490–512, 2011.

[114] “ns2 simulation code for wireless network coding.” `http://telecom.dei.unipd.it/simulation`.

[115] X. Chen, J. Shen, T. Groves, and J. Wu, “Probability delegation forwarding in delay tolerant networks,” in *Proc. IEEE Int. Conf. Comput. Commun. and Netw. (ICCCN)*, pp. 1–6, 2009.

[116] W. Gao, Q. Li, and G. Cao, “Forwarding Redundancy in Opportunistic Mobile Networks: Investigation and Elimination,” in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, pp. 2301–2309, 2014.

[117] A. Z. Broder, A. Kirsch, R. Kumar, M. Mitzenmacher, E. Upfal, and S. Vassilvitskii, “The Hiring Problem and Lake Wobegon Strategies,” in *Proc. ACM-SIAM Symp. Discrete Algorithms*, pp. 1184–1193, 2008.

[118] N. Papanikos, D.-G. Akestoridis, and E. Papapetrou, “CRAWDAD toolset tools/simulate/uoi/adyton (v. 2016-04-21).” Downloaded from `http://crawdad.org/tools/simulate/uoi/adyton/20160421`, Apr. 2016.

[119] T. H. David Kotz, “CRAWDAD: A Community Resource for Archiving Wireless Data At Dartmouth.” http://crawdad.cs.dartmouth.edu.

[120] P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, and C. Diot, "Pocket switched networks and human mobility in conference environments," in *Proc. ACM SIGCOMM Workshop on Delay-tolerant Networking (WDTN)*, pp. 244–251, 2005.

[121] J. Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, and A. Chaintreau, "CRAWDAD data set cambridge/haggle (v. 2006-01-31)." Downloaded from http://crawdad.org/cambridge/haggle/, Jan. 2006.

[122] A.-K. Pietiläinen and C. Diot, "Dissemination in opportunistic social networks: The role of temporal communities," in *Proc. ACM Int. Symp. Mobile Ad Hoc Netw. Comput. (MobiHoc)*, pp. 165–174, 2012.

[123] A.-K. Pietilainen, "CRAWDAD data set thlab/sigcomm2009 (v. 2012-07-15)." Downloaded from http://crawdad.org/thlab/sigcomm2009/, July 2012.

[124] N. Eagle and A. Pentland, "Reality mining: sensing complex social systems," *Pers. and Ubiq. Comput.*, vol. 10, no. 4, pp. 255–268, 2006.

[125] N. Eagle and A. S. Pentland, "CRAWDAD data set mit/reality (v. 2005-07-01)." Downloaded from http://crawdad.org/mit/reality/, July 2005.

[126] S. Gaito, E. Pagani, and G. P. Rossi, "Strangers help friends to communicate in opportunistic networks," *Comput. Netw.*, vol. 55, no. 2, pp. 374–385, 2011.

[127] P. Meroni, S. Gaito, E. Pagani, and G. P. Rossi, "CRAWDAD data set unimi/pmtr (v. 2008-12-01)." Downloaded from http://crawdad.org/unimi/pmtr/, Dec. 2008.

[128] J. Leguay, A. Lindgren, J. Scott, T. Friedman, and J. Crowcroft, "Opportunistic content distribution in an urban setting," in *Proc. ACM SIGCOMM Workshop Challenged Netw. (CHANTS)*, pp. 205–212, 2006.

[129] J. Leguay, A. Lindgren, and T. Friedman, "CRAWDAD data set upmc/content (v. 2006-11-17)." Downloaded from http://crawdad.org/upmc/content/, Nov. 2006.

[130] S. Tarkoma, C. Rothenberg, and E. Lagerspetz, "Theory and practice of bloom filters for distributed systems," *IEEE Commun. Surveys Tutorials*, vol. 14, no. 1, pp. 131–155, 2012.

[131] A. Lindgren and P. Hui, "The quest for a killer app for opportunistic and delay tolerant networks: (invited paper)," in *ACM Proc. Workshop on Challenged Networks (CHANTS)*, pp. 59–66, 2009.

[132] A. Keränen, J. Ott, and T. Kärkkäinen, "The ONE Simulator for DTN Protocol Evaluation," in *ICST SIMUTools: Proc. Int. Conf. Simulation Tools and Techniques*, pp. 55:1–55:10, 2009.

[133] P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, and C. Diot, "Pocket switched networks and human mobility in conference environments," in *Proc. ACM SIGCOMM workshop on Delay-tolerant networking*, pp. 244–251, ACM, 2005.

[134] J. Stehlé, N. Voirin, A. Barrat, C. Cattuto, L. Isella, J. Pinton, M. aggiotto, W. Van den Broeck, C. Régis, B. Lina, and P. Vanhems, "High-resolution measurements of face-to-face contact patterns in a primary school," *PLOS ONE*, vol. 6, 08 2011.

[135] V. Gemmetto, A. Barrat, and C. Cattuto, "Mitigation of infectious disease at school: targeted class closure vs school closure.," *BMC infectious diseases*, vol. 14, p. 695, Dec. 2014.

[136] S. Gaito, E. Pagani, and G. P. Rossi, "Opportunistic forwarding in workplaces," in *ACM Proc. Workshop Online Social Networks*, pp. 55–60, ACM, 2009.

[137] M. McNett and G. M. Voelker, "Access and mobility of wireless pda users," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 9, no. 2, pp. 40–55, 2005.

[138] T. Henderson, D. Kotz, and I. Abyzov, "The changing usage of a mature campus-wide wireless network," *Computer Networks*, vol. 52, no. 14, pp. 2690 – 2712, 2008.

[139] D. Kotz, T. Henderson, I. Abyzov, and J. Yeo, "CRAWDAD dataset dartmouth/campus (v. 2009-09-09)." Downloaded from `http://crawdad.org/dartmouth/campus/20090909`, Sept. 2009.

[140] S. Bell, A. McDiarmid, and J. Irvine, "Nodobo: Mobile phone as a software sensor for social network research," in *IEEE Vehicular Technology Conference (VTC)*, pp. 1–5, May 2011.

[141] A. McDiarmid, J. Irvine, S. Bell, and J. Banford, “CRAWDAD dataset strath/nodobo (v. 2011-03-23).” Downloaded from http://crawdad.org/strath/nodobo/20110323, Mar 2011.

[142] G. Bigwood, D. Rehunathan, M. Bateman, T. Henderson, and S. Bhatti, “Exploiting self-reported social networks for routing in ubiquitous computing environments,” in *IEEE Int. Conf. Wireless and Mobile Computing, Networking and Communications*, pp. 484–489, Oct 2008.

[143] G. Bigwood, T. Henderson, D. Rehunathan, M. Bateman, and S. Bhatti, “CRAWDAD dataset st_andrews/sassy (v. 2011-06-03).” Downloaded from http://crawdad.org/st_andrews/sassy/20110603, Jun 2011.

[144] J. Leguay, A. Lindgren, J. Scott, T. Friedman, and J. Crowcroft, “Opportunistic content distribution in an urban setting,” in *ACM Proc. SIGCOMM Workshop on Challenged Networks*, pp. 205–212, 2006.

[145] P. U. Tournoux, J. Leguay, F. Benbadis, V. Conan, M. D. de Amorim, and J. Whitbeck, “The accordion phenomenon: Analysis, characterization, and impact on dtn routing,” in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, pp. 1116–1124, April 2009.

[146] F. Benbadis and J. Leguay, “CRAWDAD dataset upmc/rollernet (v. 2009-02-02).” Downloaded from http://crawdad.org/upmc/rollernet/20090202, Feb 2009.

[147] X. Zhang, J. Kurose, B. N. Levine, D. Towsley, and H. Zhang, “Study of a bus-based disruption-tolerant network: Mobility modeling and impact on routing,” in *ACM Proc. Ann. Int. Conf. Mobile Computing and Networking (Mobicom)*, pp. 195–206, 2007.

[148] J. Burgess, J. Zahorjan, R. Mahajan, B. N. Levine, A. Balasubramanian, A. Venkataramani, Y. Zhou, B. Croft, N. Banerjee, M. Corner, and D. Towsley, “CRAWDAD dataset umass/diesel (v. 2008-09-14).” Downloaded from http://crawdad.org/umass/diesel/20080914, Sep 2008.

[149] M. Piorkowski, N. Sarafijanovoc-Djukic, and M. Grossglauser, “A Parsimonious Model of Mobile Partitioned Networks with Clustering,” in *Proc. Int. Conf. Commun. Syst. and Netw.*, pp. 1–10, 2009.

[150] M. Piorkowski, N. Sarafijanovic-Djukic, and M. Grossglauser, "CRAWDAD dataset epfl/mobility (v. 2009-02-24)." Downloaded from `http://crawdad.org/epfl/mobility/20090224`, Feb 2009.

[151] N. Aschenbruck, R. Ernst, E. Gerhards-Padilla, and M. Schwamborn, "Bonnmotion: A mobility scenario generation and analysis tool," in *ICST SIMUTools: Proc. Int. Conf. Simulation Tools and Techniques*, pp. 51:1–51:10, 2010.

[152] A. Lindgren, A. Doria, E. B. Davies, and S. Grasic, "Probabilistic Routing Protocol for Intermittently Connected Networks." RFC 6693, Oct. 2015.

[153] P. Hui, J. Crowcroft, and E. Yoneki, "Bubble rap: Social-based forwarding in delay tolerant networks," in *ACM Proc. Int. Symp. Mobile Ad Hoc Netw. and Comput. (MobiHoc)*, pp. 241–250, 2008.

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

[155] N. Papanikos and E. Papapetrou, "Coordinating Replication Decisions in Multi-copy Routing for Opportunistic Networks," in *Proc. IEEE Int. Conf. Wireless and Mobile Comput., Netw. and Commun. (WiMob)*, pp. 8–13, 2014.

[156] J. A. Davis, A. H. Fagg, and B. N. Levine, "Wearable computers as packet transport mechanisms in highly-partitioned ad-hoc networks," in *IEEE Proc. Int. Symp. Wearable Computers*, pp. 141–148, 2001.

[157] V. Erramilli and M. Crovella, "Forwarding in opportunistic networks with resource constraints," in *ACM Proc. Workshop on Challenged Networks (CHANTS)*, pp. 41–48, 2008.

[158] A. Lindgren and K. S. Phanse, "Evaluation of queueing policies and forwarding strategies for routing in intermittently connected networks," in *IEEE Int. Conf. Communication Systems Software Middleware*, pp. 1–10, 2006.

[159] S. Burleigh, E. Jennings, and J. Schoolcraft, "Autonomous congestion control for an interplanetary internet," in *SpaceOps 2006 Conference*, pp. 59–70, 2006.

[160] J. M. Pujol, A. L. Toledo, and P. Rodriguez, "Fair routing in delay tolerant networks," in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, pp. 837–845, 2009.

[161] J. Lakkakorpi, M. Pitkänen, and J. Ott, "Using buffer space advertisements to avoid congestion in mobile opportunistic dtns," in *IFIP Proc. Int. Conf. Wired/Wireless Internet Communications (WWIC)* (X. Masip-Bruin, D. Verchere, V. Tsaoussidis, and M. Yannuzzi, eds.), pp. 386–397, Springer Berlin Heidelberg, 2011.

[162] K. Wei, S. Guo, and K. Xu, "Cacc: A context-aware congestion control approach in smartphone networks," *IEEE Communications Magazine*, vol. 52, no. 6, pp. 42–48, 2014.

[163] D. G. Akestoridis, N. Papanikos, and E. Papapetrou, "Exploiting social preferences for congestion control in opportunistic networks," in *IEEE Int. Conf. Wireless and Mobile Computing, Networking and Communications (WiMob)*, pp. 413–418, 2014.

[164] T. Small and Z. J. Haas, "Resource and performance tradeoffs in delay-tolerant wireless networks," in *Proc. ACM SIGCOMM Workshop Delay Tolerant Netw. (WDTN)*, pp. 260–267, 2005.

[165] S. Batabyal and P. Bhaumik, "Mobility models, traces and impact of mobility on opportunistic routing algorithms: A survey," *IEEE Commun. Surveys Tutorials*, vol. PP, no. 99, 2015.

[166] W. jen Hsu, T. Spyropoulos, K. Psounis, and A. Helmy, "Modeling spatial and temporal dependencies of user mobility in wireless mobile networks," *IEEE/ACM Trans. Netw.*, vol. 17, no. 5, pp. 1564–1577, 2009.

[167] S. Kosta, A. Mei, and J. Stefa, "Large-scale synthetic social mobile networks with swim," *IEEE Trans. Mobile Comput.*, vol. 13, pp. 116–129, Jan 2014.

[168] T. Karagiannis, J.-Y. Le Boudec, and M. Vojnovic, "Power law and exponential decay of intercontact times between mobile devices," *IEEE Trans. Mobile Comput.*, vol. 9, no. 10, pp. 1377–1390, 2010.

[169] A. Chaintreau, A. Mtibaa, L. Massoulie, and C. Diot, "The diameter of opportunistic mobile networks," in *ACM Proc. CoNEXT Conf.*, pp. 12:1–12:12, 2007.

[170] A.-D. Nguyen, P. Senac, and M. Diaz, "Understanding and modeling the small-world phenomenon in dynamic networks," in *Proc. Int. Conf. Modeling, Analysis and Simulation Wireless and Mobile Systems*, pp. 377–384, ACM, 2012.

[171] S. Bayhan, E. Hyytia, J. Kangasharju, and J. Ott, "Analysis of hop limit in opportunistic networks by static and time-aggregated graphs," in *IEEE Int. Conf. Communications (ICC)*, pp. 3287–3292, June 2015.

[172] M. C. Gonzalez, C. A. Hidalgo, and A.-L. Barabasi, "Understanding individual human mobility patterns," *Nature*, vol. 453, pp. 779–782, Jun 2008.

[173] A. Koufoulis, "Routing in large scale social opportunistic networks," Master's thesis, Dept. of Computer Science and Engineering, University of Ioannina, Greece, Jun 2015.

[174] A. Mtibaa and K. A. Harras, "Fairness-related challenges in mobile opportunistic networking," *Computer Networks*, vol. 57, no. 1, pp. 228–242, 2013.

[175] A. P. Silva, S. Burleigh, C. M. Hirata, and K. Obraczka, "A survey on congestion control for delay and disruption tolerant networks," *Ad Hoc Networks*, vol. 25, Part B, no. 0, pp. 480–494, 2015.

[176] Q. Li, S. Zhu, and G. Cao, "Routing in socially selfish delay tolerant networks," in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, pp. 857–865, 2010.

[177] X. Zhang, G. Neglia, J. Kurose, D. Towsley, and H. Wang, "Benefits of network coding for unicast application in disruption-tolerant networks," *IEEE/ACM Trans. Netw.*, vol. 21, pp. 1407–1420, Oct. 2013.

[178] S. Ahmed and S. S. Kanhere, "Hubcode: hub-based forwarding using network coding in delay tolerant networks," *Wireless Communications and Mobile Computing*, vol. 13, no. 9, pp. 828–846.

[179] N. Shrestha and L. Sassatelli, "Inter-session network coding-based policies for delay tolerant mobile social networks," *IEEE Trans. Wireless Communications*, vol. 15, no. 11, pp. 7329–7342, 2016.

Author's Publications

Journals

1. N. Papanikos and E. Papapetrou, "Deterministic broadcasting and random linear network coding in mobile ad hoc networks," *IEEE/ACM Transactions on Networking (ToN)*, vol.PP, no.99, pp.1-15, 2017.

2. N. Papanikos and E. Papapetrou, "Revisiting XOR-based network coding for energy efficient broadcasting in mobile ad hoc networks," *Computer Communications (COMCOM)*, vol. 96, pp. 1-16, 2016.

Conferences

1. N. Papanikos and E. Papapetrou, "Coordinating Replication Decisions in Multicopy Routing for Opportunistic Networks," *in Proc. IEEE Int. Conf. Wireless and Mobile Comput., Netw. and Commun. (WiMob)*, 2014, pp. 8-13.

2. D. -G. Akestoridis, N. Papanikos, and E. Papapetrou, "Exploiting Social Preferences for Congestion Control in Opportunistic Networks," *in Proc. IEEE Int. Conf. Wireless and Mobile Comput., Netw. and Commun. (WiMob)*, 2014, pp. 413-418.

3. I. Papadopoulos, N. Papanikos, E. Papapetrou, and L. Kondi, "Network-wide MD and Network Coding for Heterogeneous Video Multicast," *in Proc. IEEE Int. Symp. Pers. Indoor and Mobile Radio Commun. (PIMRC)*, 2013, pp. 3578-3582.

In preparation

1. N. Papanikos, and E. Papapetrou, "Energy-efficient Replication-based Routing in Opportunistic Networks".

2. A. Koufoulis, N. Papanikos, and E. Papapetrou, "Pushing the limits of separation in Social Mobile Networks: Is Routing still feasible?".

Short Biography

Nikolaos Papanikos was born in Kozani, Greece. He received his B.Sc. in Computer Science and M.Sc. in Computer Systems, both from the Computer Science Department, University of Ioannina, Greece in 2007 and 2009 respectively. In the past, he has worked as a laboratory associate in the Technological Institute of Western Macedonia, Greece (2007-2013). Currently, he is a Ph.D. candidate in the Computer Science and Engineering Department, University of Ioannina, Greece. His research interests include mobile ad hoc and opportunistic networks with emphasis in the design of efficient networking algorithms.