Towards network coding enhanced routing protocols for group mobility

Xie, Lingfu

2014

Xie, L. (2014). Towards network coding enhanced routing protocols for group mobility.
Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/61756>

<https://doi.org/10.32657/10356/61756>

**TOWARDS NETWORK CODING ENHANCED ROUTING
PROTOCOLS FOR GROUP MOBILITY**

XIE LINGFU

School of Electrical & Electronic Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirement for the degree of
Doctor of Philosophy

2014

Acknowledgements

First of all, I would like to express my sincere thanks and great gratitude to my supervisor, Associate Professor Chong Han Joo, Peter. Without his intellectual guidance, constant attention and strict demands, this work would not have been completed. His extensive knowledge, enlightened direction and continuous encouragement make my Thesis work much easier. In particular, I wish to thank him for teaching me some of the philosophical aspects of research and some intangible skills, which I consider are the most invaluable knowledge I have acquired in this research program.

Next, I would like to give special thanks to Professor Liew Soung Chang in the Chinese University of Hong Kong (CUHK) for his enlightening guidance during my attachment in CUHK in 2012.

I would like to thank the Director of Infocomm Centre of Excellence, Associate Professor Guan Yong Liang, for giving me the chance of joining in the Advanced Communication Research Program.

I would also like to express my gratitude to all my family members in China for their consistent support and encouragement during all periods of my study.

Last, but not the least, I would like to thank Nanyang Technological University for all the study facilities it provided.

Table of Contents

Acknowledgements	i
Table of Contents	ii
Summary	viii
List of Figures	xi
List of Tables	xiii
Acronyms	xiv
Chapter 1 Introduction	1
1.1 Background	1
1.2 Motivations	3
1.3 Objectives	5
1.4 Major Contributions of the Thesis	6
1.5 Organization of the Thesis	10
Chapter 2 Review of MANET Routing and Network Coding	12
2.1 Mobility Models in MANETs	12
2.1.1 Entity Mobility Models	12
2.1.2 Group Mobility Models	13
2.2 Traditional MANET Routing	16
2.2.1 Routing under Entity Mobility	16
2.2.2 Routing under Group Mobility	19
2.3 DTN Routing Strategies	21
2.3.1 The Deterministic Routing	22

2.3.2	The Flooding based Routing	23
2.3.3	The Flooding Controlled Routing	25
2.3.4	The Mobility Control based Routing	26
2.3.5	Other Strategies in DTN Routing	27
2.4	Introduction of XNC	28
2.5	XNC under Reliable Links	33
2.5.1	The Determinant Factor	34
2.5.2	The Scheduling Factor	39
2.5.3	The Flow Route Selection Factor	42
2.5.4	The Counteractive Factors	46
2.5.5	Summary of Proposed XNC Schemes	47
2.5.6	Discussions	48
2.6	XNC under Lossy Links	50
2.6.1	XNC with Fixed Transmission Rate	51
2.6.1.1	XNC in the First Coding Structure Type	51
2.6.1.2	XNC in the Second Coding Structure Type	52
2.6.2	XNC with Rate Adaption	53
2.7	Drawbacks of Applying XNC	56
2.8	Conclusion	59
Chapter 3	Packet Replication based Routing Protocols under Group Mobility	61
3.1	Introduction	61
3.2	Network Model	63
3.3	G-ER for Group Mobility	64
3.3.1	The Intra-Group Routing Strategy	65

3.3.2	The Inter-Group Routing Strategy	66
3.3.2.1	Group Summary Vector Update Broadcast within a Group	66
3.3.2.2	GSV Exchange and Group based Packet Exchange	68
3.3.3	Node Buffer Management.....	70
3.3.4	Buffer Sharing Inside the Group.....	71
3.3.5	G-ER under Group Dynamics.....	72
3.4	LBGR for Group Mobility	74
3.4.1	The Intra-Group Routing	75
3.4.2	The Leader-Dominating Inter-Group Routing.....	75
3.4.3	Node Buffer Management in LBGR.....	78
3.4.4	LBGR under Group Dynamics	79
3.4.5	Complexity of LBGR.....	79
3.5	Simulation Results	80
3.5.1	Performance of G-ER without Buffer Sharing	82
3.5.1.1	G-ER at Different b and R_g without Group Dynamics.....	82
3.5.1.2	G-ER under Group Dynamics.....	86
3.5.2	Performance of G-ER with Buffer Sharing	87
3.5.3	Performance of LBGR without Group Dynamics.....	89
3.5.3.1	LBGR under The Default Traffic Model	90
3.5.3.2	LBGR under A Demanding Traffic Model.....	92
3.5.4	Performance of LBGR under Group Dynamics.....	94
3.6	Conclusion	94
Chapter 4	Mobility Control in MANETs under Isolated Groups.....	96
4.1	Introduction.....	96

4.2	Analysis of the Two Mobility Control Techniques	99
4.2.1	Network Model	99
4.2.2	Cumulative Density Function of the Packet Delivery Delay in G-ER	101
4.2.3	The Group Formation Enforcement.....	105
4.2.4	The Group Purposeful Movement	107
4.3	Purposeful Movement Assisted Routing	112
4.4	Simulation Study and Results	117
4.4.1	Simulation Model	117
4.4.2	PMAR at Different Group Buffer Sizes and Searching Ranges	118
4.4.3	PMAR at Different Group Speeds	120
4.5	Conclusion	122
Chapter 5	Studies on Network Coding Conditions in Wireless Networks.....	123
5.1	Description of Problem 1	124
5.2	Performance Analysis of DCAR and DCAR-CEO	127
5.3	A Distributed Way for Adherence of CEO.....	131
5.3.1	Overheard Nodes Recording.....	131
5.3.2	Packet Encoding/Decoding Recording	132
5.3.3	Re-Encoding Decision at the Encoding Node	133
5.4	Simulation Results for DCAR-CEO	133
5.5	Description of Problem 2	136
5.6	Virtual Overhearing and Its Applications	140
5.6.1	Conditions of Virtual Overhearing	140
5.6.2	Relaxed Network Coding Conditions	142
5.6.3	Application of Virtual Overhearing	145

5.7	Performance of XNC with Virtual Overhearing	146
5.7.1	Comparison of Rate Region	147
5.7.2	Impact of the Unawareness of $R_{nc} \leq R_{voh}$	154
5.7.3	Discussion	156
5.8	Simulation Results for NC-VOH	157
5.8.1	Performance Study of NC-VOH in Figure 5.8	157
5.8.2	Performance Study of NC-VOH in Figure 5.10(b).....	159
5.9	Conclusion	160
Chapter 6	Network Coding-Aware Routing with Virtual Overhearing	162
6.1	Introduction.....	162
6.2	Virtual Overhearing and Its Discovery	164
6.2.1	New Definitions in Virtual Overhearing.....	164
6.2.2	Virtual Neighbor Discovery	166
6.3	DCAR-VOH	168
6.3.1	Route Discovery.....	168
6.3.2	Route Selection	170
6.3.3	Virtual Overhearing Initiation.....	171
6.4	DCAR-VOH+	172
6.5	Complexity of DCAR-VOH+	176
6.6	Simulation Results and Discussion	177
6.6.1	Performance Evaluation of DCAR-VOH	178
6.6.1.1	DCAR-VOH in the Illustrative Topology.....	178
6.6.1.2	DCAR-VOH under Inter-cluster Traffic in a Random Network	182
6.6.2	Performance Evaluation of DCAR-VOH+	184

6.6.2.1	DCAR-VOH+ in an Illustrative Topology	184
6.6.2.2	DCAR-VOH+ under Random Traffic in a Random Network	186
6.6.2.3	DCAR-VOH+ under Inter-cluster Traffic in a Random Network	187
6.7	Conclusion	187
Chapter 7	Conclusions and Future Work	189
7.1	Conclusions.....	189
7.2	Recommendations for Further Work	192
7.2.1	Problem of Low Virtual Overhearing Rate.....	192
7.2.2	Extension of XNC to MANETs.....	194
7.2.3	Further Improvement of G-ER.....	195
Author's Publications		197
Bibliography		199

Summary

The routing strategies in mobile ad hoc networks (MANETs) have been mainly studied for entity mobility, where nodes move independently. In fact, many cooperative scenarios, especially those in military field, are better to be modeled as group mobility, in which nodes form different groups to move forward. Clearly, those strategies for entity mobility cannot perform well under group mobility. In this Thesis, we consider routing strategies for group mobility especially when any two groups are isolated most of time. Specifically, we first employ the delay/disruptive-tolerant network (DTN) routing for the packet delivery in group mobility; and then we consider a new technique called network coding to further improve the routing performance in group mobility.

The Thesis begins with a brief review of existing DTN routing strategies proposed for entity mobility, particularly pointing out the pros and cons of each strategy. In fact, we find that when considering group mobility, the cons of each strategy can be effectively overcome while the pros can be preserved.

First, we propose group-epidemic routing (G-ER) and leader based group routing (LBGR) to solve the resource-consuming problem in the packet replication based DTN routing. G-ER takes advantages of the cooperative nature of group mobility and incorporates two mechanisms to largely reduce the required resource such as the node buffer space and the bandwidth for the packet delivery. First, it treats each group as a single node and disseminates only one copy of one packet to each group. Second, it allows each node to share its buffer resource with other nodes in the same group. As compared to G-ER, LBGR further copes with group dynamics in group mobility by

introducing the buffer resource allocation in each group.

After that, we consider the mobility control based DTN routing in group mobility. We propose two mobility control techniques, the group formation enforcement and the group purposeful movement, both of which can be integrated into G-ER or LBGR to reduce the packet delivery delay. We analytically show the packet delivery delay in G-ER can be reduced by each technique. Then, we design a new DTN routing scheme called purposeful movement assisted routing (PMAR) based on the group purposeful movement, and show by simulation that PMAR improves G-ER under many practical network conditions.

The second part of this Thesis is to study the new technique of network coding in wireless ad hoc networks. Our work here targets *a general static network*, but it can be *extended* to group mobility. To better benefit from network coding, we address the following two problems.

We first address the re-encoding problem in network coding. We find that most proposed network coding solutions in wireless ad hoc networks may misidentify the coding opportunities when a packet is about to be re-encoded, which is detrimental to network performance. To solve this problem, we propose a principle called consistency of encoding and overhearing (CEO). The analytical and simulation results show that CEO is important to network coding to avoid performance degradation.

Then, to find more coding opportunities, we address the limitation of overhearing in network coding. Overhearing is widely used to acquire the information for the packet decoding, but it only takes effect in the one-hop neighborhood of a node, which in turn restricts the identification of coding opportunities. We propose a new technique called virtual overhearing (VOH) to allow a node to obtain a packet sent by another node which

can be two or more hops away. Thus, with VOH, more coding opportunities can be found. To make use of VOH, two network coding-aware routing schemes, distributed coding-aware routing with VOH (DCAR-VOH) and its enhanced version DCAR-VOH+, are proposed and studied. Both our analytical and simulation results indicate that VOH provides an effective way to discover more coding opportunities, resulting in improved network performance.

Lastly, the discussion on how our studies on network coding can be applied to G-ER or MANETs is presented and elaborated.

List of Figures

Figure 1.1: Overview of our work in the Thesis.....	9
Figure 2.1: Node location updating in RPGM.....	15
Figure 2.2: The end-to-end path in connected MANETs.	16
Figure 2.3: The hierarchy in group mobility.....	20
Figure 2.4: An example in deterministic routing.....	23
Figure 2.5: The packet exchange between two meeting nodes in ER.	24
Figure 2.6: An example of XNC and its rationale.	29
Figure 2.7: Rationale behind XNC.	30
Figure 2.8: Factors affecting coding opportunities in XNC.	33
Figure 2.9: Two-hop coding structures for different N	35
Figure 2.10: Butterfly coding structure and the generalized coding structure.....	35
Figure 2.11: A complicated coding structure.....	37
Figure 2.12: Examples of k-tuple coding structure.....	38
Figure 2.13: Scheduling in two-way traffic scenario.....	41
Figure 2.14: Local route switching for coding opportunities.	44
Figure 2.15: Congestion awareness in coding-aware routing.....	47
Figure 2.16: An example of XNC with relaxed coding conditions.	49
Figure 2.17: Examples of XNC with rate adaption.	55
Figure 2.18: The negative impact of the higher transmission power required in XNC....	57
Figure 2.19: The negative impact of the lower link rate selection in XNC.	58
Figure 3.1: GSV update broadcast in a group.....	67
Figure 3.2: Main procedures in the inter-group routing in G-ER.....	69
Figure 3.3: GSV and data packet exchange in LBGR when two groups meet.....	76
Figure 3.4: Major operations for the group member in LBGR.....	77
Figure 3.5: Major operations for the group leader in LBGR.	77
Figure 3.6: Intra-group packet delivery ratio.....	82
Figure 3.7: Inter-group packet delivery ratio.....	83
Figure 3.8: Overall packet delivery ratio.	84
Figure 3.9: Intra-group packet delivery delay.....	85
Figure 3.10: Inter-group packet delivery delay.....	86
Figure 3.11: Overall packet delivery ratio of G-ER under group dynamics.....	87
Figure 3.12: Performance of G-ER with and without buffer sharing.	88
Figure 3.13: Average node buffer utilization in G-ER at $R_g = 50$	88
Figure 3.14: Overall packet delivery ratio of LBGR under the demanding traffic.....	93
Figure 3.15: Overall packet deliver ratio of LBGR under group dynamics.	94
Figure 4.1: The assumed group structure.....	100
Figure 4.2: Maximal distance in a connection between two leaders.	103
Figure 4.3: The extended group reachability and the connection between two leaders under the formation enforcement.....	106
Figure 4.4: Comparison between simulation results and analytical results for F_{TGE}	107
Figure 4.5: Connection creation in the purposeful movement of a group.	108
Figure 4.6: The delay reduction of G-ER+ over G-ER at different N	111

Figure 4.7: The delay reduction of G-ER+ over G-ER at different λ_{leader}^{IM}	111
Figure 4.8: Connection notification in PMAR.....	115
Figure 4.9: The average packet delivery delay, t_d	119
Figure 5.1: Illustrative topology for the failure of network coding.	125
Figure 5.2: A more complicated topology for the failure of network coding.....	126
Figure 5.3: The rate region of (r_1, r_2) and the comparison between DCAR and DCAR-CEO.....	131
Figure 5.4: Performance comparison between DCAR and DCAR-CEO in the illustrative topology.	134
Figure 5.5: Performance comparison between DCAR and DCAR-CEO in the grid topology.	135
Figure 5.6: Performance comparison between DCAR and DCAR-CEO in random networks.	135
Figure 5.7: Some typical coding structures in XNC.	137
Figure 5.8: Network coding with virtual overhearing.....	138
Figure 5.9: Conditions of VOH.	140
Figure 5.10: Relaxed coding conditions with virtual overhearing.....	143
Figure 5.11: r_3^{max} at different (r_1, r_2) in TS and NC-VOH.	151
Figure 5.12: Verification of the correctness of the analysis.	152
Figure 5.13: Throughput gain of f3 in NC-VOH.	153
Figure 5.14: Impact of the unawareness of the critical rule.....	155
Figure 5.15: Flow throughput comparison in network of Figure 5.8.....	158
Figure 5.16: Flow throughput comparison in network of Figure 5.10(b).....	160
Figure 6.1: A more complicated example of network coding with VOH.....	164
Figure 6.2: Four definitions in a general wireless network.	165
Figure 6.3: Virtual neighbor discovery.	168
Figure 6.4: Virtual overhearing initiation in DCAR-VOH.	172
Figure 6.5: Coding opportunity creation in a more general case.	173
Figure 6.6: DCAR-VOH+ architecture.....	175
Figure 6.7: Flow throughput with a fixed rate of f2.	179
Figure 6.8: Overall throughput with a fixed rate of f2.....	179
Figure 6.9: Average packet delivery delay.	180
Figure 6.10: Flow throughput at different flow rates of f2.	180
Figure 6.11: Decoded packet ratio of f3 at different flow rates of f2.	181
Figure 6.12: Simulation topology in the random network with clusters.....	183
Figure 6.13: Overall throughput in the random network with clusters.....	184
Figure 6.14: Overall throughput of the three flows in the topology of Figure 6.12.	184
Figure 6.15: The average transmission times for each delivered packet.	185
Figure 6.16: Overall throughput in the random network.	186

List of Tables

Table 2-1: Summary of proposed XNC schemes.	47
Table 3-1: Simulation parameters.	82
Table 3-2: The packet transmission number in LBGR and ER at group radius 50	91
Table 3-3: The packet transmission number in LBGR and ER at group radius 150	91
Table 4-1: Simulation parameters in the group formation enforcement.....	106
Table 4-2: M_{min} under network setting shown in Table 4-1 but with $d \approx 0$	110
Table 4-3: Simulation parameters.	118
Table 4-4: Performance of PMAR.	120
Table 4-5: Performance of G-ER with PMAR at different group speeds.....	120
Table 4-6: Connection parameters in PMAR at different node speed when $R = 300$.	121
Table 5-1: Virtual overhearing process.....	141
Table 5-2: Performance parameters in NC-VOH and TS	159

Acronyms

MANET	Mobile Ad Hoc Network
DTN	Delay/Disruption-Tolerant Network
G-ER	Group-Epidemic Routing
ER	Epidemic Routing
LBGR	Leader Based Group Routing
PM	Purposeful Movement
PMAR	Purposeful Movement Assisted Routing
CEO	Consistency of Encoding and Overhearing
DCAR	Distributed Coding-Aware Routing
DCAR-CEO	Distributed Coding-Aware Routing with CEO
VOH	Virtual Overhearing
TS	Traditional Scheme
NC-VOH	Network Coding with VOH
DCAR-VOH	Distributed Coding-Aware Routing with VOH
DCAR-VOH+	Enhanced Version of DCAR-VOH
RWMM	Random Waypoint Mobility Model
RPGM	Reference Point Group Mobility
VTGM	Virtual Track based Group Mobility
DSDV	Destination-Sequenced Distance Vector Routing
DSR	Dynamic Source Routing
ECBRP	Enhanced Cluster based Routing Protocol
GPS	Global Positioning System
LANMAR	Landmark Ad Hoc Routing
SV	Summary Vector
GSV	Group Summary Vector
XNC	Inter-flow Network Coding
INC	Intra-flow Network Coding
noCoCo	Near-Optimal Coordinated Coding
CSMA	Carrier Sense Multiple Access
GID	Group ID
IEEE	Institute of Electrical and Electronics Engineers
MAC	Media Access Control
RTS	Request to Send
CTS	Clear to Send
CDF	Cumulative Density Function
ODEs	Ordinary Differential Equations
UDP	User Datagram Protocol
TCP	Transmission Control Protocol
MCN	Multi-hop Cellular Network
4G	Fourth Generation
MS	Mobile Station
BS	Base Station

OHR	Overhearing Record
ROHR	Reported Overhearing Record
VNR	Virtual Neighbor Record
RREQ	Route Request
RREP	Route Reply
CCS	Channel Congestion State
MIQ	Modified Interface Queue

Chapter 1

Introduction

1.1 Background

A mobile ad hoc network (MANET) comprises a number of nodes which move within the network area. Each node is generally equipped with a radio transmission device, with which the node can communicate with other nodes nearby directly. However, when any two nodes in MANETs are out of the direct communication range, the communication is accomplished by employing some other nodes as the relay nodes to forward the traffic between them. This communication mode in MANETs is distinct from that in traditional cellular networks where the traffic between two mobile nodes is first directed to the fixed infrastructures, i.e., base stations. Thus, MANETs are infrastructureless networks where every node can function as the router or the base station for others. In particular, with no infrastructure required, MANETs can be deployed easily and bear many special applications in reality, e.g., the disaster and emergency relief, the wild animal tracking, and the military operation.

In general, the node movement in MANETs can be classified into entity mobility and group mobility [1]. In entity mobility, nodes are assumed to move independently within the network; whereas in group mobility, nodes form different groups and those in the same group generally move forward as a whole. Various real examples can be found for these two mobility types. Particularly, the cooperative scenarios are better to be

modeled as group mobility. Here, we show three examples only for group mobility. First, in the disaster relief, there normally exists a searching team dedicated to cooperatively looking for trapped/missing people and a medical team responsible for cooperatively curing those injured [2]. Due to the same mission or the required cooperation, members in one group or team generally stay close to each other and move together. Second, in wild animal tracking [3, 4], most species, e.g., the zebra and the whale, are socialized and thus, they move in different groups to search for food. Third, importantly, in military operations, soldiers or tanks are generally organized in groups, with each being assigned a certain mission.

In group mobility, it is highly possible that any two groups are isolated or disconnected for some time duration. By two *isolated or disconnected* groups, we mean two groups that are far away such that the communication between any two nodes from them cannot be established. We present in the following the major causes for the isolation of two groups.

- Limited power supply [3, 5, 6]: in many MANET applications such as wireless sensor networks, the battery of each node has limited power and it is not rechargeable. In wild animal tracking, for example, sensors attached to the animals are generally compact such that the power for each sensor is very limited [3, 6]. As a result, the following two features are present in MANETs: (1) the radio transmission range at each node is short; and (2) a node is prone to be down due to the limited power.
- Large or even uncertain network area [3]: in some MANET applications such as the wild animal tracking, nodes like animals may move within a large-size or even uncertain area.

- Hostile network environment: in some MANET applications, e.g., in the military field, one node like the soldier or the tank may be down due to the attack from the enemy [5].

As we know, group mobility represents a variety of cooperative scenarios, and it often appears in our real-life examples. However, under the isolated or disconnected groups caused by the three reasons above, the communications for nodes under group mobility will be often disconnected and interrupted. Hence, in this Thesis, we concern about the communications in MANETs under isolated or disconnected groups and aim to propose some effective routing strategies to achieve high network performance.

1.2 Motivations

It has been shown that traditional MANET routing schemes fail to work in a disconnected network [7, 8]. Here, traditional MANET routing schemes refer to either the table-driven routing schemes or the on-demand routing schemes [9]. In the table-driven routing, each node always maintains a routing table indicating the next hop to reach any destination node; and in the on-demanding routing, only when a source node generates some packets to a destination node, the source will attempt to find a complete end-to-end path to the destination. Particularly, for a packet destined to a certain destination, if no next hop is indicated in the table-driven routing, or if no complete path is found in the on-demand routing, the packet will be dropped eventually. Hence, when a destination is disconnected to a source or when a network is disconnected, the traditional MANET routing will fail to deliver packets to the destination nodes.

Fortunately, the routing in the delay/disruption-tolerant network (DTN) or the DTN routing provides a paradigm for the packet delivery between two disconnected

nodes [5]. A DTN is actually a sparse or disconnected network where any two nodes in the network are disconnected most of time. The DTN routing employs a new approach called the *store-carry-forward* approach to deliver packets. In this approach, when a packet is received by a node, it is generally *stored* in the buffer of the node and *carried* by the node during its movement; later, as new contact opportunities arise among nodes, this packet can be *forwarded* to other nodes or even its destination node. Obviously, we see that the DTN routing makes use of the node mobility to deliver packets, and that the store-carry-forward approach is also applicable to MANETs under isolated groups. Thus, how to design effective routing schemes for isolated groups based on the store-carry-forward approach is our main focus in this Thesis.

In addition, we also notice that a new technique called network coding [10] is effective in boosting network performance such as the network throughput. Traditionally, an intermediate node transmits packets from different flows separately to their destinations. In contrast, if the intermediate node above could transmit only one packet and this *single* packet can let the destination of each flow obtain its desired packet, then, it can be expected that the network throughput will be improved. This is actually what network coding really does. Generally, the transmitted packet in network coding is a mixture of those packets from different flows. In fact, MANETs under isolated groups can also benefit from network coding. For example, network coding may be used by any node in one group when that node is responsible for forwarding packets from different flows in that group. Thus, how to make effective use of network coding in MANETs under isolated groups is another focus in this Thesis.

1.3 Objectives

We here first introduce some key issues or problems in the DTN routing and network coding, respectively. Then we will present the objectives of our research work.

The first issue in the DTN routing is how to guarantee the packet delivery with limited resources such as the node buffer space. In the DTN routing, before being delivered to the destination, a packet generally needs to be carried in the buffer by different nodes for relatively long time. Thus, we see that the node buffer resource can critically affect network performance, because packets will be dropped in case of insufficient buffer space at each node. Moreover, in order to increase the packet delivery probability and decrease the end-to-end delay, many proposed DTN routing schemes choose to make copies for a packet and disseminate them to different nodes [7, 11]. This packet replication based routing strategy actually introduces a tradeoff between the network resource and the network performance. Hence, in MANETs under isolated groups, how to achieve satisfactory network performance but with limited node buffer resource is an important concern in our study in this Thesis.

The second issue in the DTN routing is about the node mobility control. As mentioned earlier, the packet delivery in a DTN largely relies on the node mobility. Thus, if the node mobility could be controlled properly in some scenario, it will benefit network performance. For example, upon detecting that the destination node of a packet is moving nearby, a node carrying that packet can deliver it faster by moving towards the destination. Despite the advantage of the mobility control, previous studies targeting entity mobility have shown the difficulties in the deployment of this technique [12, 13] because of the need of the centralized coordinator or some other strict assumptions.

Hence, how to apply the node mobility control to group mobility is the second concern in our study.

For network coding, we here consider one fundamental issue; the network coding conditions. In fact, network coding only works under some certain network conditions. Intuitively, the more the network coding opportunities are identified, the more the network performance will benefit. Thus, how to relax the network coding conditions to find more coding opportunities is the third concern in our study in this Thesis.

Based on the three issues above, the objectives of our study in MANETs under isolated groups are listed as follows.

- (1) To propose a DTN routing scheme that requires less node buffer space to achieve the same network performance (e.g., the packet delivery ratio or the packet delivery delay) as most existing DTN routing schemes.
- (2) To propose some mobility control technique(s) that can be well supported by group mobility.
- (3) To relax the network coding conditions in most existing network coding solutions and identify additional network coding opportunities in a general network.

1.4 Major Contributions of the Thesis

First, this Thesis makes the following contributions to the DTN routing under group mobility.

1) We propose a packet replication based (or flooding based) routing scheme called group-epidemic routing (G-ER) [14] to achieve the first objective above. Thus far, although various routing strategies have been proposed to deliver packets in a DTN [15,

16], we find that the simple strategy, i.e., the flooding based strategy, is a good match for group mobility. This is because the group structure in group mobility can be exploited to largely mitigate the resource-consuming problem in the flooding based routing strategy while preserving its advantage in speeding up the packet delivery. The proposed G-ER is designed based on epidemic routing (ER) [7], which is a popular flooding based routing, and it incorporates two mechanisms supported by the group structure in group mobility to lower the resource requirement. First, each group is treated as a single unit and only one copy of one packet is disseminated to each group instead of each node. Second, the buffer space of a node is shared by all nodes within its group. With these two mechanisms, the simulation results show that G-ER requires much lesser buffer space than ER to guarantee the packet delivery. However, it is also shown that the performance of G-ER is degraded under a scenario specific to group mobility, i.e., under group dynamics.

2) We propose another scheme called leader based group routing (LBGR) [17] to cope with group dynamics in group mobility. LBGR is designed based on G-ER, and it considers strategically storing the packet received by nodes in one group to alleviate the impact from group dynamics. That is, in LBGR, most packets are required to be stored at the group leader, i.e., a node who organizes and manages the group. Meanwhile, LBGR considers the buffer resource allocation in each group to work with this new strategy. Our simulation results show LBGR is able to cope with group dynamics.

3) We propose two mobility control techniques for group mobility; the group formation enforcement and the group purposeful movement. Both can be easily integrated into some existing DTN routing scheme such as G-ER to expedite the packet delivery. We integrate each technique into G-ER and then analytically derive the

cumulative density function of the packet delivery delay to show the reduced delay. Moreover, we design a new DTN routing scheme called purposeful movement assisted routing (PMAR) based on the second technique, and conduct simulations to show the effectiveness of PMAR in improving G-ER under various network conditions.

Second, this Thesis makes the following contributions to the body of network coding in wireless ad hoc networks.

1) We first solve a problem called the re-encoding problem in most proposed network coding solutions. We find that most proposed network coding solutions may misidentify the coding opportunities when a packet is about to be re-encoded, which is detrimental to network performance. To solve this problem, we propose a principle called consistency of encoding and overhearing (CEO) [18]. By adhering to CEO, the encoding node is able to avoid misidentifying coding opportunities and ensure the decoding of all encoded packets. Both the analytical and simulation results show that CEO is important to network coding to avoid performance degradation.

2) We then propose a new technique called virtual overhearing (VOH) to find additional coding opportunities. In network coding, overhearing is widely used to acquire the information for the packet decoding, but it only takes effect in the one-hop neighborhood of a node, which in turn restricts the identification of coding opportunities. By contrast, the proposed VOH allows a node to obtain a packet sent by another node which can be two or more hops away. Thus, with VOH, more coding opportunities can be found. Our analysis and simulation results in some typical topologies show that the additional coding opportunities brought by VOH can be translated into performance gain.

3) To take advantage of VOH, we design two network coding-aware routing schemes; distributed coding-aware routing with virtual overhearing (DCAR-VOH) and

its enhanced version DCAR-VOH+. Actually, VOH can increase coding opportunities in two ways. First, it increases the chance of mixing a new incoming flow with existing flows. Second, and more importantly, it allows a new incoming flow to create coding opportunities between two existing flows that were previously unmixable. DCAR-VOH is designed based only on the first feature whereas DCAR-VOH+ incorporates both features. Our extensive simulations indicate that VOH provides an effective way to discover more coding opportunities, resulting in improved network performance.

We emphasize that our work on network coding above targets connected and static wireless ad hoc networks, and it can be extended to G-ER and enhance G-ER. We will also discuss the extension of network coding to the routing under group mobility in this Thesis.

Consequently, the overview of our work in this Thesis is shown in Figure 1.1. As seen from the figure, G-ER (or LBGR) is the base of our work, and other proposed schemes can be added onto it. In particular, even though our work on network coding targets a general wireless ad hoc network, it can be extended to G-ER with some additional mechanism(s).

Figure 1.1: Overview of our work in the Thesis.

1.5 Organization of the Thesis

The rest of this Thesis is organized as follows:

Chapter 2 gives a review of both the basics of MANET and network coding. For the former, we review the traditional MANET routing and the DTN routing. For the latter, we conduct a survey on inter-flow network coding in wireless ad hoc networks. In inter-flow network coding, only packets from different flows can be mixed or encoded. In this survey, we first present various situations or conditions in which inter-flow network coding can be applied. Then, some factors that could be considered to increase the positive effect of inter-flow network coding are introduced.

Chapter 3 proposes G-ER and LBGR for group mobility. In G-ER, the two mechanisms proposed to largely lower the resource requirement are illustrated in detail. In addition, how G-ER reacts to group dynamics is also considered in this Chapter. In LBGR, how to cope with group dynamics is the focus. How packets are stored in one group and why LBGR is resilient to group dynamics are illustrated in detail. Extensive simulations are conducted to examine the effectiveness of the two proposed mechanisms in G-ER and the effectiveness of LBGR in improving G-ER under group dynamics.

Chapter 4 introduces two node mobility control techniques in group mobility; the group formation enforcement and the group purposeful movement. We integrate each technique into G-ER, and then analytically show the packet delivery delay of G-ER can be reduced by the proposed technique. In addition, the new scheme PMAR is also presented in this Chapter, and how it improves G-ER under different network conditions is studied by simulation.

Chapter 5 studies network coding conditions and addresses two problems in the

conditions in previous studies. First, we presents the principle of CEO for network coding to solve the re-encoding problem in network coding, and we show the importance of CEO to network coding by both analysis and simulation in a simple network topology. Second, we propose the new technique of VOH for network coding relaxation, and we show by both analysis and simulation in some typical network topologies that the additional coding opportunities brought by VOH can be translated into network performance gain.

Chapter 6 introduces DCAR-VOH and DCAR-VOH+. How VOH is detected by a node and how a new incoming flow discovers a path with coding opportunities are described in DCAR-VOH. Then, how a new incoming flow creates coding opportunities between two existing flows that were previously unmixable is illustrated in DCAR-VOH+. Extensive simulations are also conducted to evaluate the two proposed schemes under various network conditions.

Finally, Chapter 7 concludes this Thesis and points out some future work. Particularly, how to extend our work of network coding to G-ER will be presented in detail as part of the future work.

Chapter 2

Review of MANET Routing and Network Coding

In this Chapter, we review the basics of MANETs and network coding in wireless ad hoc networks. The review of both are critical to the design of the routing under group mobility in this Thesis. For the former, we focus on mobility models, the traditional MANET routing, and the DTN routing. For the latter, we restrict our attention to one important form of network coding, namely inter-flow network coding (XNC), because it can be used to effectively boost wireless network performance such as the network throughput. In what situations or conditions XNC can be used and how to make better use of XNC are the two main concerns in this review. In addition, we also present some drawbacks of applying XNC. This Chapter is organized as follows. From Section 2.1 to Section 2.3, the basics of MANETs are introduced, and from Section 2.4 to Section 2.7, XNC is reviewed. Section 2.8 concludes this Chapter.

2.1 Mobility Models in MANETs

2.1.1 Entity Mobility Models

Many entity mobility models have been proposed to mimic how a person or an entity moves in reality [1]. Among them, some models such as the Random Walk Mobility

Model and the Random Waypoint Mobility Model (RWMM) have simple rules to generate the node movement; whereas some others like the Gauss-Markov Mobility Model [1] consider more practical factors in node movement and thus have more complicated rules. However, all these models own one common feature that nodes move within the network area independently or how a node moves is not affected by any other nodes in the network. Here, we present the basic rules on how each node moves in RWMM because RWMM is widely adopted for the performance evaluation of MANET routing.

1. At the beginning, all nodes in RWMM are randomly placed within the network area.
2. Each node then randomly chooses a point within the network area as the destination point to move to. Meanwhile, to move to the destination point, this node chooses a velocity uniformly distributed within $[V_{min}, V_{max}]$.
3. After each node arrives at the destination point at step 2, it pauses for time t_p .
Then, it repeats Step 2.

2.1.2 Group Mobility Models

As mentioned in Chapter 1, group mobility is useful in modeling those cooperative or socialized activities. The reason is that nodes in cooperation usually share the same interest or the same assigned mission and they are willing to form a group to accomplish some task together [1, 19]. Thus far, many group mobility models have been proposed in the literature [2, 20-24]. Among them, some models like the Reference Point Group Mobility (RPGM) [20] are generic and have simple rules to generate the node movement; whereas some others like the Group Force Mobility model [23] and the Virtual Track

based Group Mobility (VTGM) [21] are more specific and consider many practical factors to generate the node movement. Here, we illustrate how nodes move in different groups in the generic model of RPGM because RPGM is widely adopted for the performance evaluation of MANET routing.

First, the following presents the key rules used in RPGM.

1. Each group consists of a leader and a number of members.
2. Each group has a logical center, and all nodes in one group are distributed within some certain scope around the logical center. This scope (hereafter called the group coverage in this Thesis) can be a circle with the radius of R_g for simplicity.
3. The motion of the logical center defines the entire group's movement. The trajectory of the logical center can be a predefined path or conform to some aforementioned entity mobility model like RWMM.
4. Each node in one group is associated with a reference point, which is fixed and randomly located inside the group coverage. Each node is only allowed to move around its reference point within a distance of R_r . Moreover, the reference point follows the group movement or the logical center movement, and it keeps a constant position relative to the logical center.

Then, based on the four rules above, we illustrate how the location of each node in one group is updated in RPGM. In fact, RPGM defines two motion vectors, the group motion vector, $\overrightarrow{GM}$, and the random motion vector, $\overrightarrow{RM}$. $\overrightarrow{GM}$ represents the current velocity of the logical center of a group, and it is shared by all reference points in that group. $\overrightarrow{RM}$ represents the random motion of a node with reference to its reference point. The magnitude and the direction of $\overrightarrow{RM}$ are uniformly distributed within $[0, R_r]$ and

within $[0, 2\pi]$ around its reference point, respectively. Thus, based on $\overrightarrow{GM}$, the new position of the reference point of a node can be calculated, and based on $\overrightarrow{RM}$, how the node is positioned with reference to its updated reference point can also be known. Figure 2.1 gives an example of a group in RPGM moving from time t to time $t + 1$. In the figure, when the logical center moves from time t to time $t + 1$ based on $\overrightarrow{GM}$, all reference points also move to new locations based on $\overrightarrow{GM}$; then, each node selects a new location in the neighborhood of its reference point based on $\overrightarrow{RM}$.

Figure 2.1: Node location updating in RPGM.

Other important features of RPGM are presented as follows. First, RPGM can be used to model various scenario types by properly choosing its parameters. For example, by properly arranging the positions of the reference points in a group, a certain formation can appear in that group, which is useful in modeling the activities in the military field. Second, and more importantly, any two nodes in one group could be connected most often, since all nodes in one group, as shown in Figure 2.1, stay close to each other and move around the logical center. In fact, this second feature is also present in other group

mobility models we mentioned above.

2.2 Traditional MANET Routing

Due to the multi-hop nature of MANETs, routing plays a critical role in delivering the packet from a source node to a destination node, since it is used to find a next hop through which the destination node is reachable or becomes closer. In a dense or connected MANET, any two nodes in the network are generally connected by at least one end-to-end path, on which any two neighboring nodes are within the radio transmission range of each other, as shown in Figure 2.2. Hence, how to find such an end-to-end path between any two nodes is essential in connected MANETs. The traditional MANET routing [9] was proposed for this task. From the literature [8, 9, 25], we find that the design of the traditional MANET routing scheme largely depends on the mobility model used in the network. Thus, the following two sub-sections present some representative routing schemes under entity mobility and under group mobility, respectively.

Figure 2.2: The end-to-end path in connected MANETs.

2.2.1 Routing under Entity Mobility

We here introduce two routing strategies which are used to discover a shortest end-to-end path between two nodes. The first routing strategy is the table-driven routing [26-28], in which each node always maintains a routing table indicating the next-hop node and the

distance (or the number of hops) to any destination node in the network. As an example in Figure 2.2, the next hop in node S's routing table to the destination node D is node 1; similarly, the next hop in node 1's routing table to the destination node D is node 2, and so on. The second routing strategy is the on-demand routing [29, 30], in which an end-to-end path will be found by a source node only when the source node has a session to be established to some other node. For example, when node S in Figure 2.2 attempts to set up a session to node D, the complete path, $S \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow D$, will be found by node S. We now present one representative scheme for each aforementioned routing strategy.

Destination-Sequenced Distance Vector Routing (DSDV) [26]: This routing protocol falls into the table-driven routing. To maintain a fresh routing table for each node in MANETs or in a dynamic environment, each node in DSDV is required to periodically broadcast its routing table (called routing table update) to its neighbors. Initially, when a node joins the network, its routing table only contains itself as a destination with the hop distance being 0. When a node receives a routing table update from its neighbor, it will update its table as follows. For each destination node shown in the received table, if the destination node is seen for the first time, the node adds this destination node to its own table, sets the neighbor as the next hop, and increases the hop distance by 1. Otherwise, only if there is a shorter path to the destination node via this neighbor, the node will use this shorter path to update its routing table for this destination node. In addition to this periodic routing table update broadcast, DSDV also implements the trigger update broadcast to keep the routing table at each node fresh or more current. This trigger broadcast takes place only when the change of the network topology is detected by a node, e.g., a broken link is detected by a node.

In order to avoid the propagation of the stale routing information and the

formation of the route loop, DSDV requires each node to issue a sequence number in its routing table update. This sequence number increases each time when the node is about to broadcast the routing information. The sequence number of one node is later carried in other nodes' updates and further propagated to others. Thus, upon receiving an update for one destination node, a node first checks whether the sequence number of this destination node in the update is larger than the one in its own table. If it is higher, it means the route update is fresh and can be used to update the old one for this destination node.

Pros and cons of DSDV: as a table-driven routing protocol, DSDV performs well only in scenarios where the network scale is small and the network topology changes slowly. This is mainly because in a large-scale network or in a network where nodes move fast, the periodic routing table updates in DSDV generate a considerable amount of overhead and consume much bandwidth. To avoid this overhead problem, the on-demand routing is an option.

Dynamic Source Routing (DSR) [30]: This routing scheme falls into the on-demand routing, and it attempts to find an end-to-end route only when a source node intends to set up a data flow to a destination node. DSR incorporates two key mechanisms; the route discovery mechanism and the route maintenance mechanism. Here, we only present the route discovery mechanism because it is essential in DSR and related to our work in this Thesis.

To discover a route to a destination node, two steps are involved in DSR. First, a source node broadcasts a route request (RREQ) which contains the IDs of the source node and the destination node and a forwarder list. Then, upon receiving this request, each node first appends its ID to the forwarder list, and then rebroadcasts it. Finally, this request will reach the destination node. Second, the destination node sends to the source

node a route reply (RREP), which contains all forwarders recorded in RREQ. With these forwarders, the source node can eventually acquire a source route to the destination and then send its packets along this source route. In addition, DSR implements a route cache mechanism to let each node learn more available routes from other nodes and store them in its route cache. These routes are overheard by a node from the packet transmission of its neighbors, because packets such as RREP and the data packet always carry the source route information. With some routes stored in the route cache, a source node can consult its route cache about a route to a destination node before initiating the route discovery process.

Pros and cons of DSR: In DSR, no periodic route update is needed and thus, the routing overhead problem in the table-driven routing can be avoided. However, as compared to DSDV, DSR may suffer from a longer packet delivery delay due to the need of the route discovery process.

Besides the table-driven routing and the on-demand routing, some other routing strategies are also proposed in MANETs under entity mobility [8, 31-33]. For example, the cluster-based routing [8, 33] is effective in controlling the routing overhead in a moderate scale MANET; the geographic position assisted routing [31, 32] makes use of the node location from Global Positioning System (GPS) to deliver packets, and it performs effectively in a moderate or even a large scale MANET. However, when considering a small scale network with only several hops, both the table-driven routing and the on-demand routing generally suffice.

2.2.2 Routing under Group Mobility

We have previously pointed out a unique feature in group mobility that nodes in a group

stay with each other most often. This topology feature in group mobility can be considered to facilitate the route setup [25, 34, 35]: in group mobility, each group is normally treated as a logical sub-network, and it is represented by some certain node such as the leader in the group. Then, the network becomes hierarchical in that the top layer comprises the representative nodes of all groups and the bottom layer comprises the rest of the nodes. Figure 2.3 shows this hierarchy in group mobility. Thus, accordingly, the routing also becomes hierarchical, because packets destined to a node in other group need to be forwarded via both the two layers. To illustrate how packets are routed in this hierarchical network, we here briefly introduce one earlier proposed scheme called Landmark Ad Hoc Routing protocol (LANMAR) [25] for group mobility.

Figure 2.3: The hierarchy in group mobility.

LANMAR: based on the hierarchical topology shown in Figure 2.3, LANMAR provides a framework for the design of the routing in both the top layer and the bottom layer. For the top layer in LANMAR, a table-driven routing like DSDV can be applied. However, the routing table only shows the route information regarding the representative node of each group (which is called the landmark of the group in LANMAR). This table

is called the global table. For the bottom layer in LANMAR, the table-driven routing can still be adopted. However, this routing table only needs to provide the route information to those within several hops of a node, which is able to cover all nodes in one group. We call this table the local table. Thus, if the destination node of a packet appears in the local table, it can be delivered using the local table. Otherwise, the packet is first routed towards the landmark of the destination group, where the destination node belongs to, using the global table, and then delivered to the destination node using the local table.

The main advantage of LANMAR or other hierarchical routing schemes for group mobility is that the routing overhead can be effectively controlled [9]. However, similar to the cluster-based routing [8], this advantage only results from a stable group structure where nodes in one group seldom leave for another group.

2.3 DTN Routing Strategies

As mentioned earlier, a DTN is a sparse or disconnected MANET where an end-to-end path between any two nodes rarely exists. For example, the situation in Figure 2.2 that nodes S and D are connected by several other nodes is unlikely to exist in a DTN. As a result, the abovementioned MANET routing strategies under entity mobility or group mobility fail to work in a DTN due to the lack of a complete route between the source and destination nodes. However, previous studies [5, 36-38] have come up with some approaches to achieve the communications in a DTN. The first approach is to increase the network connectivity by introducing some special nodes such as robots to the network. The main task of these special nodes is to move purposely to fill the gaps in the network (or avoid the disconnection between any two nodes) such that traditional MANET routing schemes can then be applied [36-38]. The second approach is the

aforementioned store-carry-forward approach. With this approach, the network disconnection is tolerable, and the packet delivery can still be achieved. This advantage of the store-carry-forward approach, as compared to the first approach above, makes it widely studied in the literature. As mentioned earlier, each packet in this second approach needs to be carried by some node(s) in the buffer for long time, so how to select the node(s) as the carrier(s) for a packet is the main concern in the DTN routing. Here, we introduce several previously proposed DTN routing strategies based on the store-carry-forward approach in the literature.

2.3.1 The Deterministic Routing

In the deterministic routing [39, 40], once a packet is generated in the network, how it will be forwarded among mobile nodes is already determined. In fact, the deterministic routing makes use of the knowledge of how all nodes will move in the future to select the packet carrier. This routing strategy works as follows. First, each node is assumed to know its own movement plan and the plans of all other nodes so that it can construct the network topology at any time. Then, by knowing how the topology evolves in the future, a source node can compute a set of intermediate nodes which can carry its packets continuously to deliver them to the destination node. The work in [39] proposes an algorithm called the tree based deterministic routing for each source to compute this set of carriers, details of which are omitted here.

As a simple example in [39], Figure 2.4 shows how the source node S delivers a packet to the destination node D. At some time between t_0 and t_1 , node S generates a packet, say P1, destined to node D, and it knows there is no end-to-end route for this packet. However, with the known movement plans of nodes E and D that node E will

move toward node D from time t_1 to t_2 and then meet node D from time t_2 to t_3 , node S selects node E as the carrier by forwarding the packet P1 to it. Then, after carrying the packet P1 for some time, node E can deliver it to node D between time t_2 and t_3 , finally.

The most challenging issue in this routing strategy is how to select those carriers for each packet such that the shortest average packet delivery delay is achieved. This problem can be much more complicated when considering limited node buffer space and limited bandwidth [40]. Details of this issue are omitted here.

Figure 2.4: An example in deterministic routing.

Pros and cons: one advantage of the deterministic routing is that with the known node's movement plan, nodes can make intelligent packet forwarding decisions and deliver the packet faster. However, this routing strategy is only applicable to some scheduled systems such as the satellite communication system and the bus system.

2.3.2 The Flooding based Routing

The flooding based routing, namely epidemic routing (ER) in [7], assumes all nodes do not know the movement plan of any other node in the network. However, it can be expected that the destination node of one packet will surely meet some other nodes in the future. Thus, if the copies of this packet are carried by all nodes but the destination, the packet can surely be delivered to the destination node in the future. This is the main idea of the flooding based routing. As shown in Figure 2.5, four steps are involved in ER to

exchange packets between two meeting nodes.

Step 1: when two nodes, say nodes A and B, meet each other, they will identify the *different* packets between them. Here, the *different* packets refer to the packets that are stored at one node's buffer but have never been received by the other node. Each node maintains a table, called summary vector (SV), to record all received packets. Note that a packet in SV is uniquely represented by its source node ID and its packet ID assigned by the source. Whenever a node receives a new packet, it will update its SV by recording the new packet. Thus, when node A meets node B, one node, say node A, will send its SV to the other, node B. Then, node B can determine the *different* packets between them.

Step 2: once node B identifies those *different* packets, it sends a request to node A for the *different* packets ever received by node A.

Step 3: node B sends to node A the *different* packets which are stored in node B's buffer but have never been received by node A.

Step 4: upon receiving the request from node B, node A sends to node B those *different* packets that node B needs.

Figure 2.5: The packet exchange between two meeting nodes in ER.

To avoid the frequent SV exchange between the same two nodes, each node in ER keeps the record of the contacts with other nodes. With this record, only the node, which has not been met for some certain duration, is considered a new neighbor, with which the SV exchange will be activated.

Pros and cons: ER is a simple routing scheme, and it can achieve the optimal performance given unlimited node buffer space and bandwidth. However, due to the packet flooding, ER is a resource-consuming protocol, especially when the number of the node becomes large in the network.

2.3.3 The Flooding Controlled Routing

This flooding controlled routing is proposed to solve the resource-consuming problem in ER by restricting the number of the packet copies in the network. Thus far, a number of flooding controlled routing schemes have been proposed in the literature [11, 41-54]. In this flooding controlled routing, how many packet copies should be generated and which nodes should be the carriers for those packet copies are the two major questions. With the knowledge of how nodes meet with each other in the network, authors in [11, 51] propose to relate the number the packet copies to the packet delivery delay requirement in a DTN. In general, more copies of a packet are needed when the low packet delivery delay is desired. Many other studies, however, consider avoiding unnecessary packet replication to restrict the number of the copies of each packet. As a result, the only question to be answered in these studies is how to select the node(s) to be the carriers for a packet. We now present how this question is generally answered in these studies.

In many real scenarios, one node usually has different probabilities to meet other nodes. Thus, a packet can be delivered faster to its destination node if the nodes with higher probability to meet the packet's destination node are selected as the carriers. More specifically, a packet can be forwarded to new carriers only if these new carriers have higher probability to meet the destination node. This is the approach that those studies adopt, and hence we call it the contact probability based routing. From the literature, we

find that this contact probability based routing either attempts to exploit some known pattern repeatedly presenting in nodes' movement [45, 46, 48, 53, 54] or concerns about how to make good use of the contact history of two nodes so that the future contact between them can be better predicted [41-44, 49, 50, 52].

Pros and cons: obviously, the contact probability based routing can effectively overcome the resource-consuming problem in ER, but it only performs well in scenarios where the movement of the nodes shows some certain predictable moving pattern.

2.3.4 The Mobility Control based Routing

In the aforementioned three DTN routing strategies, packets wait for new contact opportunities passively to be forwarded to other nodes. This means the node mobility is never affected or controlled by the traffic demand. Obviously, one common problem of these three strategies is that the packet delivery delay is unpredictable in case that the destination node is out of contact from most of the nodes for long time. To overcome this unpredictability of the packet delivery delay, or to reduce the packet delivery delay, the mobility control based routing is proposed [12, 13, 55-58].

In the mobility control based routing, the mobility of some nodes is controlled such that they can move purposely over the network area to assist in establishing a connection to the destination node(s). Thus, with mobility control in a DTN, packets can be delivered faster. In [13, 55-58], a special kind of node, the ferry node or the robot, is introduced to the network, and its mobility is controlled to assist other nodes in the packet delivery. For example, in [13] if a source node has packets to send to a destination node, the ferry node will first move towards the source node and carry its packets. Then, the ferry node helps deliver the packets to the destination node. In addition, authors in

[12] design a different solution in which no special node is introduced but the mobility of the nodes in the network is controlled by their peers. In [12], some intermediate nodes are employed to move to certain locations to help establish an end-to-end connection to the destination node.

Pros and cons: Clearly, with node mobility control, the packet delivery delay can be shortened to some extent. However, we find that the drawback of introducing a special kind of node (i.e., the ferry node) lies in the centralized deployment of these special nodes, and that the work in [12] suffers from its strict assumption that how each node moves in the network should be known to each other in advance.

2.3.5 Other Strategies in DTN Routing

From the literature, we find that some other DTN routing strategies are proposed, but most of them are based on the aforementioned DTN routing strategies. Here, we illustrate them as follows. First, some previous studies [59, 60] consider a special DTN scenario where a portion of nodes meet each other more frequently, and apply clustering to it. The formed cluster can be used to benefit the routing performance. For example, in [59] a table-driven routing scheme is adopted within each cluster to route packets quickly because of the frequent meeting between two nodes in a cluster. Second, similarly, authors in [61-63] consider the hybrid routing scheme that integrates the traditional MANET routing and DTN routing for the packet delivery. In this hybrid routing, a packet can be delivered by either of the two routing modes. Obviously, the two strategies above only suit some certain scenarios. Third, authors in [64-66] further investigate the flooding based routing or the flooding controlled routing under limited node buffer space or limited bandwidth. For example, authors in [65] consider which packet(s) stored at a

node should be given the priority for transmission when that node meets some other node in a bandwidth-limited DTN.

2.4 Introduction of XNC

The shared nature of the network resources and the fading nature of the wireless channel pose two big challenges for the design of an efficient scheme for the packet delivery in wireless networks [67]. First, similar to wired networks, resources such as the bandwidth in wireless networks are shared among different flows; these flows at one common intermediate node need to compete for the use of the common channel whenever an transmission opportunity arises for this node. This competition among flows imposes a main constraint on network performance. Second, different from wired networks where the channel is stable and reliable, the wireless channel generally experiences drastic variation due to channel fading, causing each transmission unreliable or prone to be a failure. To overcome this problem, retransmission generally serves as a necessary way to guarantee the packet delivery in this situation. Obviously, the more retransmission times it requires, the less efficiently the bandwidth is used. Hence, by mitigating either the competition among flows for the shared resource or the retransmission problem will undoubtedly benefit wireless network performance.

Network coding [10] is a promising technique for increasing wireless network performance. Proposed in [10] for achieving multicast capacity in wired networks, network coding introduces a new paradigm of packet processing and forwarding, i.e., it allows an intermediate node to encode or mix different received packets into one single *coded* packet and send the coded packet along an outgoing channel. Clearly, this new paradigm is different from the traditional packet routing, where packets are simply

replicated and then forwarded at intermediate nodes. Importantly, this new paradigm of network coding can be successfully applied to wireless environments to improve network performance even under unicast traffic [68-70]. Regarding the packets to be encoded together, network coding can be classified into inter-flow network coding (XNC) and intra-flow network coding (INC): XNC (INC) considers encoding packets from different flows (the same flow). It can be found in the literature that both XNC and INC [71-73] can be used for increasing wireless network performance, and that some work even considers combining the two forms of network coding to better benefit network performance [74-76]. This is because both XNC and INC can effectively address the two big challenges mentioned above. However, they have distinctly different design issues [104]. In this Thesis we emphasize on illustrating how the two challenges can be resolved by XNC under unicast traffic, i.e., the traffic model that we concern about in group mobility.

XNC under reliable wireless links: Consider a wireless network shown in Figure 2.6 where flow 1 (f1) and flow 2 (f2) intersect at nodes A, B and C. Suppose each node is required to store all data packets it has already sent. We now focus on how the intermediate node, node B, handles the packet forwarding when it receives P1 from f1 and P2 from f2 in the following two schemes.

Figure 2.6: An example of XNC and its rationale.

Traditional routing: as mentioned earlier, node B is only allowed to replicate the

received packet and forward it. Thus, P1 and P2 will be sent individually by node B to node C and node A, respectively. This means f1 and f2 compete for the channel use at node B and two transmission times are needed for node B to forward P1 and P2.

XNC: here, we consider a simple form of packet encoding, i.e., XOR. Node B XORs P1 and P2 into an encoded packet, $P1 \oplus P2$, and then broadcasts the encoded packet to both node A and node C. Upon receiving $P1 \oplus P2$, node A (C) XORs it with the stored packet, P1 (P2), to yield P2 (P1), which is what node A (C) expects to receive. Consequently, with network coding or XNC, one transmission from node B serves f1 and f2 simultaneously, which alleviates or even eliminates the competition between f1 and f2 for the channel use at node B, and thus improves network performance as compared to traditional routing.

Figure 2.7: Rationale behind XNC.

We expose the rationale behind the example of XNC in Figure 2.6. Let us consider a scenario where one node, say node 1, demands a packet, P1, from another node, say node 2. Traditionally, node 2 has only one option to satisfy the demand of node 1, i.e., node 2 only considers sending P1 to node 1. However, the option at node 2 can actually be diversified by considering the packets previously received at node 1. For example, if node 1 has received a bundle of packets before, then, node 2 can encode any

packet from that bundle with P1 to form an optional encoded packet and send it to node 1. Thus, the packet options that node 2 can choose to satisfy node 1's demand are expanded. Furthermore, with the expanded packet options for satisfying each demand, node 2 may find a common option to simultaneously satisfy multiple different demands, as shown in Figure 2.7. If a common option is found, it means there is a network coding opportunity and thus, as shown in Figure 2.6, the network throughput will be improved. Particularly, the improved performance is attained even at the expense of less energy consumption due to the reduced transmission times at the intermediate node. In fact, XNC is such a scheme that exploits this option commonness among different flows. Unfortunately, this common option only exists in some certain situations called *coding structures*, which will be explained and surveyed in Section 2.5.

XNC under lossy wireless links: Consider the scenario in Figure 2.7 where node 1 (3) has stored packet P2 (P1) and now demands packet P1 (P2) from node 2. We assume in Figure 2.7 that the wireless link is unreliable or the packet delivery probability from node i to node j is given by $p_{i \rightarrow j}$. Now, let us compare in this scenario the average total transmission times at node 2 between the traditional routing and XNC.

Traditional routing: similar to the case in Figure 2.6, node 2 needs to separately transmit P1 and P2 to node 1 and node 3, respectively, so the average total transmission times will be the sum of the expected transmission times from node 2 to node 1 and that from node 2 to node 3. Since the probability for P1 to be correctly received at node 1 is $p_{2 \rightarrow 1}$, and the probability for the acknowledgement of node 1 to be correctly received at node 2 is $p_{1 \rightarrow 2}$, the expected transmission times from node 2 to node 1 is $1/(p_{1 \rightarrow 2} p_{2 \rightarrow 1})$ [77]. Similarly, the expected transmission times from node 2 to node 3 is $1/(p_{3 \rightarrow 2} p_{2 \rightarrow 3})$. Thus, the average total transmission times at node 2, $T_{routing}$, is given by

$$T_{routing} = 1/(p_{1 \rightarrow 2} p_{2 \rightarrow 1}) + 1/(p_{3 \rightarrow 2} p_{2 \rightarrow 3})$$

XNC: instead of sending P1 and P2 separately, node 2 keeps broadcasting the XORed packet, $P1 \oplus P2$, to both nodes 1 and 3 until it receives the acknowledgements from both. Then, we can derive the following expression for T_{nc} , the average total transmission times at node 2. Details of the derivation of T_{nc} can be found in [77].

$$T_{nc} = T_{routing} - \frac{1}{p_{1 \rightarrow 2} p_{2 \rightarrow 1} + p_{3 \rightarrow 2} p_{2 \rightarrow 3} - (p_{1 \rightarrow 2} p_{2 \rightarrow 1} p_{3 \rightarrow 2} p_{2 \rightarrow 3})}$$

Hence, compared to traditional routing, XNC here can save some retransmission times for node 2 to complete its tasks, and thus improve the efficiency of the bandwidth usage at node 2.

The rationale behind this example is twofold. First, the existence of the common option (i.e., $P1 \oplus P2$) as mentioned above at node 2 can be used to serve two different demands simultaneously. Second, during the retransmissions of the common option, any of its demanders (i.e., node 1 or 3) hearing it can extract its needed packet; whereas during a packet retransmission in traditional routing, it is useless for a demander (e.g. node 3) to overhear the retransmission destined to others (e.g. P1 to node 1). Hence, with XNC, an intersection node of multiple flows can reduce the total retransmission times to let the next hop of each flow correctly receive its desired packet.

Therefore, we see from the two examples in Figure 2.6 and Figure 2.7 that XNC can effectively address the two aforementioned big challenges in wireless networks. In particular, combining the two examples, we also see that as long as a common packet option or a coding opportunity is found at an intermediate node, XNC will benefit network performance regardless of the quality of its links. More importantly, around the exposed rationale behind each of the two examples, we find that a variety of XNC

schemes have been proposed in the literature. Although there is some survey work on network coding [78, 79], we find that no work particularly focuses on XNC. Hence, in this Chapter we attempt to give an in-depth study of XNC in wireless networks under unicast traffic. Based on the two examples above, we will provide a survey on XNC in wireless networks under reliable links and under lossy links, respectively. Specifically, we will present in the survey various factors that can be considered to make network performance benefit more from XNC.

2.5 XNC under Reliable Links

Figure 2.8: Factors affecting coding opportunities in XNC.

In this Section, we present the survey of XNC in wireless networks with reliable links. As observed from the example in Figure 2.6, to benefit network performance from XNC, we need to find the coding opportunity (i.e., the common packet option) to serve different flows. Intuitively, the more coding opportunities we can find, the more improvement we can make to network performance. Hence, our survey here focuses on how to find coding opportunities as many as possible. Figure 2.8 shows various factors, including the determinant factor, the positive factors, and the counteractive factors, that

will affect the coding opportunity to be found for two or more intersecting flows. In particular, the determinant factor and the positive factors can be considered for increasing coding opportunities whereas the counteractive factors are used to be against the greediness of the network for coding opportunities. In what follows, we first explain how each factor in Figure 2.8 affects the coding opportunity to be found. Then, a table is used to summarize how each proposed representative XNC scheme considers these factors. Finally, we end this Section with some discussions.

2.5.1 The Determinant Factor

As seen in Figure 2.8, the determinant factor refers to the coding structure type, which actually reflects the conditions of the coding opportunity in XNC. Hence, this factor is essential to XNC because it answers how different flows (e.g. f_1 and f_2 in Figure 2.6) need to intersect with each other such that the intersection node (e.g. node B in Figure 2.6) can encode packets from them. In the literature, a number of coding structure types have been proposed, and they are enumerated as follows.

Two-hop coding structure [80]: this coding structure type considers encoding packets from N flows with $N \geq 2$. Suppose node C is the intersection node of these flows, and by $prev(i)$ and $next(i)$ we denote the previous hop and the next hop of flow i ($1 \leq i \leq N$) at node C, respectively. Then, two features exist in this type: (1) for any i and j ($i \neq j$), we have either $next(i) = prev(j)$ or $next(i) \in Ngb(prev(j))$, where $Ngb(prev(j))$ denotes all neighbors of the node, $prev(j)$; and (2) each encoded packet is decoded at all next hops of the encoding node, i.e., node C. Using the two features, all possible coding structures for $N = 2$ are constructed in Figure 2.9(a), (b), and (c). Particularly, the structure in Figure 2.9(a) is called reverse carpooling [81]. Let us focus

on Figure 2.9(c) to see why the intersection node C is an encoding node. Due to overhearing, node E (node D) can obtain P1 (P2) when the packet is sent to node C by node A (node B). Thus, similar to the example in Figure 2.6, P1 and P2 can be encoded at node C. Similarly, using these two features, we can construct all two-hop coding structures for other N . Figure 2.9(d) and (e) show one possible coding structure for $N = 3$ and $N = 4$, respectively.

Figure 2.9: Two-hop coding structures for different N .

Figure 2.10: Butterfly coding structure and the generalized coding structure.

Butterfly coding structure [82]: this is a specific structure and it only relaxes the requirement in the two-hop coding structure in Figure 2.9(c) by allowing the packet decoding at the node two hops away rather than one hop away from the encoding node. Figure 2.10(a) shows the butterfly structure where nodes UM and LM are the two

intersection nodes of f_1 and f_2 and node LL (LR) on f_1 (f_2) can overhear node UL (UR) on f_2 (f_1). Within this structure, node UM will encode packets P_1 and P_2 sent from nodes UR and UL, respectively, and send the mixed packet $P_1 \oplus P_2$ to node LM. Then, node LM will broadcast this packet to nodes LL and LR, where the packet decoding will take place. Here, we can see that the decoding nodes, LL and LR, are two hops away from the encoding node, UM. Actually, this butterfly structure can be easily extended to the generalized coding structure as shown below.

Generalized coding structure [83]: the authors in [83] propose the sufficient conditions for the intersection node of two flows to become the encoding node. The coding structure formed under these conditions is called the generalized coding structure [83]. Suppose the intersection node is node c , and we denote $Ngb(a)$, $U(a, f)$, and $D(a, f)$ as node a 's neighbors, all node a 's upstream nodes on flow f , and all node a 's downstream nodes on flow f , respectively. Then, node c will become an encoding node when the following two conditions are satisfied: (1) there exists a node $d_1 \in D(c, f_1)$, such that $d_1 \in U(c, f_2)$ or $d_1 \in Ng(b(s_1))$, where $s_1 \in U(c, f_2)$, and (2) there exists a node $d_2 \in D(c, f_2)$, such that $d_2 \in U(c, f_1)$ or $d_2 \in Ng(b(s_2))$, where $s_2 \in U(c, f_1)$. As an example, one coding structure formed under these two conditions is shown in Figure 2.10(b). In Figure 2.10(b), node 7 on f_2 and node 4 on f_1 can overhear node 1 on f_1 and node 5 on f_2 , respectively; node 3 can encode packets from f_1 and f_2 ; and the encoded packet will be decoded at both node 7 and node 4.

Although the two conditions above are used for the coding opportunity identification between only two flows, it can be easily extended to N flows with $N \geq 3$ [83]. When an intersection node of the N flows is an encoding node for any two flows among the N flows, it is allowed to encode N packets from these N flows into one

all the three packets, it broadcasts the following two encoded packets to all the three neighbors: $P1 \oplus P2$ and $P2 \oplus P3$. Then, each receiver can get its desired packet by XORing the following three packets; the two received encoded packets and the packet it sent to node R. For example, node A can get packet P3 by doing $P1 \oplus (P1 \oplus P2) \oplus (P2 \oplus P3)$. The similar happens at both nodes B and C.

Figure 2.12: Examples of k -tuple coding structure.

The k -tuple coding structure actually considers XNC in a scenario where k nodes demand traffic from each other via a common relay node R. However, the traffic cannot be random; it should be in a certain pattern that any node among the k demanders can be demanded by only one other node [85]. Figure 2.12(b) shows an example of the 4-tuple coding structure, which presents the abovementioned traffic pattern.

In k -tuple coding structure, the relay R performs network coding as follows [85]. Suppose the demanded packets at the k nodes are $P1, \dots$, and Pk , then, node R will generate the $k - 1$ encoded packets as $P1 \oplus P2, P2 \oplus P3, \dots$, and $Pk - 1 \oplus Pk$. Finally, each demander can get the packet it desires by using the $k - 1$ received encoded packets and the packet it sent to the relay.

Remark on coding structure type in XNC: all above surveyed coding structure types present different network coding conditions under which packets from two or more

flows can be encoded at some intersection node. It is obvious that the more the coding structures are considered, the more the coding opportunities can be found. However, we find in the literature that the most widely used coding structures are those simple structures for only two flows, such as the structures shown in Figure 2.9(a), (b), and (c). One main reason why those complicated structures for three or more flows are rarely used can be found in [86]: due to the geometrical constraints on the coding structure, the probability for any coding structure of N flows with $N \geq 4$ to appear in a realistic wireless network is very low.

2.5.2 The Scheduling Factor

The coding structures introduced above define the situations where packets from two or more flows can be encoded. However, it does not mean the encoding node can encode multiple packets from different flows in each of its transmissions, especially when *opportunistic scheduling* such as the IEEE 802.11 standard is used in the network. For example, in Figure 2.9(a), when node C is about to send a packet from f1, it is highly possible that there happens to be no queued packet from f2. Then, no network coding can be performed at node C, and thus network performance cannot be maximally enhanced. Hence, this begets the question of how to maximally utilize the coding opportunity in XNC. From the literature, we find that a major factor that affects the coding opportunities to be utilized is the scheduling. In what follows, we present different scheduling schemes in XNC.

Centralized scheduling: suppose in Figure 2.9(a) f1 and f2 are sending traffic to each other at rates of r_1 and r_2 , respectively, then, the maximal coding opportunities that can be utilized in one unit time at node C will be $\min(r_1, r_2)$. This upper bound of the

amount of the coding opportunity actually can be achieved by the centralized scheduling. It can be found that this scheduling generally appears in the Linear Program formulation of some XNC scheme in wireless networks [81, 87-89]. Nonetheless, the big challenge here is how to achieve the system behavior defined by the centralized scheduling in a distributed manner. One way to address or ease this challenge is to introduce a central coordinator to the network [90], which is responsible for computing routes for flows and properly allocating bandwidth resource to each flow.

noCoCo scheduling [91]: the authors in [91] propose a solution called Near-Optimal Coordinated Coding (noCoCo) to maximally utilize coding opportunities for a special kind of traffic pattern, namely, the two-way traffic, where two nodes send packets to each other on the same path but in opposite directions. Figure 2.13(a) shows one example of this traffic pattern, where node 0 and node 3 send packets to each other. The goal in [91] is to propose a proper distributed scheduling to maximally increase the chance for each intermediate node or relay node to send encoded packets. One important feature in noCoCo scheduling is the use of a backpressure rule, with which each intermediate node is not allowed to store in its buffer more than one packet of each flow. To follow this rule, each intermediate node only needs to know the packets stored in its neighbors' queues to determine which flow's packet to send. With this backpressure rule, it is stated in [91] that after injecting sufficient packets from the two flows, the network will eventually reach the desired state that some intermediate node, say node X, has stored two packets from two flows, and its left (right) neighbor stores one packet to the right (left) destination, as shown in Figure 2.13(b). Then, node X can encode the two packets and broadcast the encoded packet to its two neighbors. After that, node X will refrain from transmission until each of its two neighbors performs a transmission, which

will ensure node X to perform network coding again in its next transmission. The similar happens at other intermediate nodes.

Although the simulation in [91] shows the benefit of network coding under noCoCo scheduling, this scheduling has the following limitations. First, it may not work well in the scenario where one flow's packet can be encoded with any other flow's packet at some intermediate node. This is because the backpressure rule allows each node to store only one packet from one flow, which will cause some coding opportunities missed at that intermediate node [92]. Second, it may not be able to work effectively under random traffic pattern.

Figure 2.13: Scheduling in two-way traffic scenario.

Random access with different priorities [86, 93-95]: the scheduling presented here is based on random access among nodes. In this scheduling, one key factor that affects coding opportunities to be found is the node priorities in accessing the channel. The work [86, 93-95] we survey here only considers a simple coding structure, i.e., the two-hop coding structure, and the Markov chain is generally employed for analysis.

The authors in [86] have studied the performance of the two-hop coding structure with N flows under two types of scheduling, i.e., equal access among nodes and higher priority for the encoding node. They show that the equal access leads to a higher network throughput in case of saturated sources. This is because each time when the encoding

node occupies the channel, it generally can encode more packets into one coded packet than that under the higher-priority access for the encoding node, thus resulting in higher network throughput. Similar findings are presented in [95] for the two-hop coding structure.

In addition, the authors in [93] and [94] analyze how the transmission probability of the encoding node, p_r , affects the coding opportunities and the network throughput in the coding structure of Figure 2.9(a). Although their studies are based on Slotted ALOHA and Carrier Sense Multiple Access (CSMA), respectively, they draw a similar conclusion that a moderate p_r will result in the highest network throughput: a too high p_r will result in unsaturated queues at the encoding node and thus few coding opportunities at it; and a too low p_r actually makes the encoding node the bottleneck of the network.

Remark on scheduling: we can see from the surveyed scheduling schemes above that all proposed scheduling schemes for XNC have a common drawback, i.e., the difficulty of applying them to a general and distributed network. For example, although it is stated in [86, 93-95] that we can adjust the transmission probability of the encoding node to achieve high network performance, it is still unknown whether it can be applied to the scenario with multi-hop flows instead of only two-hop flows. Therefore, we find in the literature that most distributed XNC schemes only consider the opportunistic scheduling (as shown in Table 2-1).

2.5.3 The Flow Route Selection Factor

As shown in the coding structures above, flows, to be mixable with each other at one intersection node, need to properly intersect with each other. However, in wireless ad hoc networks there normally exist multiple available paths for each incoming flow. Thus, this

begets the question of how to select the path for a flow so as to form some coding structures and benefit from XNC. In what follows, we give the survey on different approaches that deal with the flow route establishment for increasing coding opportunities.

Network backbone construction [96, 97]: a multi-hop wireless ad hoc network is generally considered as a flat or infrastructureless network, where each node plays the same role in delivering packets. Hence, there normally exist multiple paths for each incoming flow in this flat network. By contrast, an infrastructure based network always requires the traffic to be directed to the infrastructure nodes first. Due to the limited number of the infrastructure node in the network, they can form a common path segment for different flows, with the traffic running on them in opposite directions, and thus a number of coding structures shown in Figure 2.9(a) will appear on this segment. Motivated by this, the work in [96, 97] aims to maximally utilize network coding by selecting some nodes in the flat ad hoc network to play as the infrastructure nodes or the dominators. By connecting these dominators, a network backbone can be formed, on which all traffic will be routed. The simulation in [96] shows the benefit of backbone based XNC over regular routing based XNC; the simulation in [97] shows that backbone based XNC also improves network performance even under broadcast traffic.

Local route switching [82, 98]: this approach is proposed under the traditional routing such as DSR [30] to increase coding opportunities. Since the traditional routing schemes do not consider coding opportunities during the path selection to establish new flows, the abovementioned coding structures may not appear among flows. However, it is possible that some coding structure could appear by properly adjusting part of the established flow paths, thus making coding opportunities arise. We use the example in

[98] for illustration. Two flows are initially established as shown in Figure 2.14, and there is no coding opportunity at any node of these two flows. However, node B, with its neighbor information limited to two hops, can determine that by letting it forward the packet from node D to node E, the coding structure in Figure 2.9(c) will appear. Thus, node B can ask node D to forward its packet to it instead of node H, and then it can send the packet using network coding.

Figure 2.14: Local route switching for coding opportunities.

Network coding-aware routing: compared to the approach of local route switching, it is better to identify the coding opportunity during route establishment of a new incoming flow. This new routing is called network coding-aware routing [87]. By contrast, the routing which is oblivious to coding opportunities in the flow route setup and only identifies coding opportunities after the flow route setup is called coding-oblivious routing [87]. The coding-aware routing aims to find a route or routes for a new incoming flow on which it can be mixed with existing flows. Generally, due to the benefit of network coding, a non-shortest route with coding opportunities on it may become an ideal option for a new flow. Now, we present some representative network coding-aware routing schemes as follows.

The work in [81, 87-89] formulates the joint scheduling and network coding-aware routing as a linear program. The formulation considers any traffic pattern of

concurrent unicast flows. More specifically, the work in [87, 89] aims to find the maximal network throughput for those flows, whereas the work in [81, 88] aims to find the minimal energy consumption for delivering some certain traffic amount of each flow. By solving the linear program, the upper bound of the throughput in [87, 89] and the lower bound of the energy consumption in [81, 88] can be obtained. For example, the work in [87] proves that network coding-aware routing has higher network throughput than coding-oblivious routing in many different network scenarios. Moreover, the work in [81] proposes a distributed way for achieving the lower bound of the energy consumption, of which we omit the details here.

Besides, various distributed network coding-aware routing schemes have been proposed, most aiming to find the best path for a new flow [83, 99-102]. Basically, we find that all these proposed schemes differ from each other in the coding structure type used for coding opportunity identification and the metrics used for path evaluation. As mentioned earlier, here the widely used coding structure types include the two-hop coding structure for two flows and the generalized coding structure for two flows. The path metrics used in these schemes mainly include the hop number of the path, the coding opportunity amount on the path, and the congestion state of the path. As we know, more coding opportunities could be found on a longer route or a more congested route, in general. Thus, we find in these schemes that the quality of a path is not only measured by the coding opportunity amount: the work in [99, 102] considers the first two path metrics and the work in [83, 101] considers the last two path metrics. In particular, the authors in [100] additionally address the issue of flow rate control in coding-aware routing. To make effective use of XNC, they consider using all possible paths rather than one best path for a new flow and injecting traffic on each path at a rate jointly determined by the

coding opportunity amount on the path and the congestion state of the path.

2.5.4 The Counteractive Factors

Here, we present two counteractive factors, the congestion awareness and the node or network lifetime awareness, which are considered to be against the greediness of the network for coding opportunities. In other words, by considering either one of the two factors the coding opportunities will be decreased to some extent.

Even though coding opportunities can be effectively increased by applying the backbone based routing or network coding-aware routing, it does not necessarily mean network performance will always benefit from them. Rather, the greediness for coding opportunities sometimes might harm network performance: as just mentioned above, it is possible that a network with more coding opportunities could become more congested. Figure 2.15 shows such an example where f_3 , from node D to node F, is searching for a route with coding opportunities after f_1 and f_2 are established. It can be seen that f_3 will find coding opportunities at node B if node B is selected as the next hop. However, as can be observed in the figure, due to the traffic from f_1 and f_2 via node B, node B becomes much more congested than node E where no coding opportunity exists for f_3 . Hence, a better choice for f_3 in this situation is to choose node E instead of node B as the next hop for higher throughput. From this example, we can see why the aforementioned coding-aware routing schemes [83, 100, 101] consider both the coding opportunity amount and the path congestion state in search for a route for a new flow.

Besides, the authors in [103] point out another disadvantage of the greediness for coding opportunities. That is, although routing traffic for more coding opportunities can save the total energy consumption of the network, it causes the encoding node to deplete

its energy much faster than other non-encoding nodes. For example, in backbone based routing, the nodes in the backbone will drain their energy faster than others because most traffic needs to be forwarded by them. In fact, the faster energy depletion of some nodes will shorten the network lifetime. Hence, to extend the network lifetime, it is necessary to be aware of the node lifetime for routing traffic: the authors in [103] propose to adopt multi-path routing together with flow traffic splitting to distribute the traffic more evenly in the network and thus ensure the network lifetime.

Figure 2.15: Congestion awareness in coding-aware routing.

2.5.5 Summary of Proposed XNC Schemes

In this Sub-section, we summarize in Table 2-1 how each representative XNC scheme proposed under reliable links addresses the factors illustrated above. It can be seen from the table that (1) opportunistic scheduling is widely adopted in many XNC schemes due to its simplicity; and that (2) in most centralized scheduling based XNC schemes (e.g., [85, 87, 103]), the routing is generally aware of the coding opportunities and the network congestion state.

Table 2-1: Summary of proposed XNC schemes.

XNC scheme	Coding structure type	Scheduling	Flow route selection	Congestion awareness	Node lifetime awareness
------------	-----------------------	------------	----------------------	----------------------	-------------------------

COPE [80]	Two-hop coding	Opportunistic	Coding-oblivious routing	No	No
BFLY [82]	Butterfly coding	Opportunistic	Local route switching	No	No
DCAR [83]	Generalized coding	Opportunistic	Coding-aware routing	Yes	No
Jones [85]	k -tuple coding	Centralized	Coding-aware routing	Yes	No
Le [86]	Two-hop coding	Random access with different priorities	Coding-oblivious routing	No	No
Umehara [93, 94]					
Sengupta [87]	Two-hop coding ($N = 2$)	Centralized	Coding-aware routing	Yes	No
Gheibi [81]	Reverse carpooling			No	
Shabdanov [89]	Reverse carpooling			Yes	
Huang [88]	Two-hop coding ($N = 2$)			No	
noCoCo [91]	Reverse carpooling	noCoCo	Coding-oblivious routing	No	No
BRONC [96]	Two-hop coding ($N = 2$)	Opportunistic	Backbone Construction	No	No
NCDS [97]	Two-hop coding	Opportunistic	Backbone Construction	No	No
MMSR [99]	Reverse carpooling	Opportunistic	Coding-aware routing	No	No
FORM [102]	Generalized coding ($N = 2$)	Opportunistic	Coding-aware routing	No	No
C ² AR [101]	Two-hop coding ($N = 2$)	Opportunistic	Coding-aware routing	Yes	No
RCR [100]	Reverse carpooling	Opportunistic	Coding-aware routing	Yes	No
Gaddam [103]	Reverse carpooling	Centralized	Coding-aware routing	Yes	Yes

2.5.6 Discussions

We have surveyed a number of factors in XNC which can affect the coding opportunities to be found under reliable links. Particularly, it can be seen that the coding opportunity identification highly depends on the identification of the coding structure type. However, it is quite possible that in some scenarios two or more flows may not form any of the

aforementioned coding structures. This begets the question of whether an intermediate node can still perform XNC when none of those coding structures is formed. This actually aims for the relaxation of the XNC conditions. An attempt for this relaxation should be worthwhile, because XNC can be more exploited in that manner. Here, we show one possible attempt using an example borrowed from [104, 105] where wired networks are considered.

Figure 2.16: An example of XNC with relaxed coding conditions.

Figure 2.16 shows that network coding is still enabled at an intersection node, node C, of two flows (i.e., f1 and f2) even when no aforementioned coding structure is formed by the two flows. As compared to the generalized coding structure in Figure 2.10(b), here it does not require any downstream node on f1 (f2) of node C to overhear some upstream node on f2 (f1) of node C. However, it guarantees the decoding of the packet encoded at node C by requiring some upstream node on f1 (f2) to deliberately send packets to some downstream node on f2 (f1) along some new route. This example shows one way to relax network coding conditions, but a number of issues need to be addressed first for this relaxation. For example, we may need to address 1) which upstream node on f1 (f2) should be selected to send packets, 2) which downstream node on f2 (f1) should be selected to receive the packets, and 3) which route should be

selected between the two nodes.

2.6 XNC under Lossy Links

In this Section, we present the survey of the XNC schemes under lossy links. In particular, we mainly survey how XNC can be combined with rate adaption to increase network performance. Due to relatively limited amount of work found in the literature on this topic, we directly introduce each representative scheme in this survey.

Before proceeding to the survey below, we first classify the aforementioned coding structures into the following two types.

- The coding structure without overhearing links: this type requires no overhearing link in the coding structure. Obviously, the reverse carpooling in Figure 2.9(a) and the k -tuple structure in Figure 2.12 fall into this type.
- The coding structure with overhearing links: this type makes use of the overhearing link for packet decoding purpose. It can be seen that the two-hop coding structures shown from Figure 2.9(b) to (e) and the butterfly coding structure shown in Figure 2.10 fall into this type.

The reason behind this classification is that under lossy links, the encoding node in the first type can always encode multiple packets from flows whereas it may be not allowed in the second type due to the unreliable overhearing links. In Figure 2.9(c), for example, if node F fails to overhear P1 from node G due to the unreliable link between them, then, node C cannot encode P1 and P2. Due to this uncertainty of the coding opportunity under lossy links, the design of the XNC scheme of the second type will be different from that of the first type, as presented below.

2.6.1 XNC with Fixed Transmission Rate

First, we consider the XNC schemes with fixed transmission rate under lossy links. The XNC schemes of the first and second coding structure types will be surveyed here.

2.6.1.1 XNC in the First Coding Structure Type

We find that most factors surveyed above are still applicable to the XNC schemes of the first coding structure type, due to the certainty of the coding opportunity in this type and the benefit of XNC shown in the example in Figure 2.7. Particularly, due to the simplicity of the reverse carpooling structure, we find that it has been studied more often than the k -tuple structure: the following three XNC schemes [77, 106, 107], which are surveyed to show how the scheduling factor and flow route selection factor can be considered to increase coding opportunities, are based on the reverse carpooling structure.

The authors in [77] consider the minimal cost to deliver a certain amount of traffic of each flow. This minimization problem is formulated as a linear program in [77], where the routing, scheduling, and the reverse carpooling based XNC are jointly considered. The basic idea behind the formulation is that the traffic of each flow is allowed to be split and distributed onto multiple paths, which then determines the amount of coding opportunities at each intermediate node. Given a packet delivery probability for each link, the total required number of packet transmission (i.e., the cost) can then be computed by summing up the required number of transmission at each node. Note that an encoding node in the reverse carpooling structure only needs T_{nc} rather than $T_{routing}$ transmission times to deliver two packets from two flows. Through performance evaluation in [77], it shows that the shortest path routing with network coding can save the cost as compared to that without network coding, and that network coding-aware

routing can further reduce the cost.

The authors in [107] continue with the work in [77] by addressing the problem of the greediness for coding opportunities in [77]. In maximizing coding opportunities, the work in [77] ignores the constraints of link capacity and the interference around each link. Hence, the work in [107] adds these constraints into the linear program in [77] for a more reasonable modeling and formulation. We here omit the details of the formulation in [107].

Other distributed reverse carpooling based XNC schemes, e.g., [106], can also be found in the literature, but we omit details of them because most of them deal with the same design factors as those schemes we surveyed under reliable links.

2.6.1.2 XNC in the Second Coding Structure Type

We find that the uncertainty of the coding opportunity under lossy links in the second coding structure type makes those aforementioned factors difficult to be applied. For example, the work in [86] states that using the equal channel access in the two-hop coding structure shown in Figure 2.9(c), plenty of coding opportunities will arise at node C and the highest network throughput could be attained under reliable links. However, this may not be true when the overhearing link from node A to node E (or from node B to node D) has low delivery probability, because many packets queued in the buffer of node C cannot be encoded in this situation. From the literature, we find that XNC is normally used in combination with INC under unreliable links, and that XNC is mainly used in an opportunistic manner, with no scheduling or flow route selection being considered. For example, in [74, 75], an encoding node under the two-hop coding structures will encode two packets from two flows only when it knows the next hop of each flow has already

overheard or obtained the packet for decoding. Obviously, under unreliable links, the coding opportunities opportunistically arise at the encoding node in [74, 75].

2.6.2 XNC with Rate Adaption

Now, we present a new factor, i.e., the transmission rate, which can affect the performance of the XNC schemes in both coding structure types classified above. The reason to consider this factor is that the packet delivery probability of each link can be improved by reducing the transmission rate [108]. We can see that T_{nc} in the first coding structure type in Figure 2.7 varies with different packet delivery probabilities, and that the amount of the coding opportunity in the second structure type varies with different reliabilities of the overhearing links. Since both T_{nc} and the amount of the coding opportunity are closely related to network performance, the transmission rate can affect the performance of the XNC scheme in each of the two coding structure types.

The authors in [109] consider the optimal transmission rate for the encoding node under the reverse carpooling structure in Figure 2.7. For node 2 in Figure 2.7 to deliver P1 and P2 to nodes 1 and 3, respectively, node 2 adjusts its transmission rate as follows [109]. First, node 2 repeatedly broadcasts $P1 \oplus P2$ at a rate of R_{nc} until it is received by at least one receiver (i.e. node 2 or node 3). Next, if node 1 (node 3) does not received its desired packet, the desired packet will be transmitted by node 2 at an optimal rate R_1 (R_3). Note that the rate R_1 (R_3) is the rate at which node 2 can transmit a unit of data to node 1 (node 3) in the shortest time. To determine an optimal R_{nc} , the authors point out by simulation that in most cases the two packets, P1 and P2, are delivered to their destinations in the shortest time by setting $R_{nc} = \min(R_1, R_2)$. However, they fail to propose a scheme to determine an optimal R_{nc} .

The authors in [110] consider the rate adaption for the encoding node in the two-hop coding structure of N ($N \geq 2$) flows. Two important assumptions are made in [110]: (1) all overhearing links, if any, are reliable in the coding structure; and (2) the encoding node is allowed to transmit an encoded or non-encoded packet only *once* rather than *multiple times* as shown in Figure 2.7 to ensure the packet reception at each intended receiver. Particularly, to ensure the packet reception at a set of M ($M \leq N$) nodes, the encoding node needs to choose a transmission rate R such that $R = \min(R_1, \dots, R_M)$, where R_i ($1 \leq i \leq M$) is the rate at which the encoding node can reliably deliver a packet to the i th receiver. With a chosen receiver number M and the corresponding transmission rate R , the capacity of the link from the encoding node to any of the M nodes is equal to $M \times R$. Thus, each time when the encoding node is about to send a packet, it encodes packets from those flows which will yield the largest link capacity.

The authors in [111] employ the rate adaption to address the unreliability of the overhearing link in the second coding structure type. One assumption in [111] is that for one node to transmit a packet to another node, if the transmission rate is higher (no larger) than the link capacity, the receiver will erroneously (correctly) receive the packet. For simplicity, we here only present two examples in [111], as shown in Figure 2.17(a) and (b), to illustrate how the rate adaption can be combined with XNC to affect network performance. In both figures, the number associated with each link denotes the maximal capacity of that link, and nodes A and B intend to send 1 bit to nodes C and A, respectively. In Figure 2.17(a), if node B chooses the rate of 1, then, node C cannot overhear node B and there is no coding opportunity at node R. As a result, the throughput is $2/4 = 0.5$. If node B chooses the rate of 0.8, then, node R can encode the two bits from nodes A and B, which results in a larger throughput of $\frac{2}{2+1.25} = 8/13$. However, the

situation will turn around in the example of Figure 2.17(b). Obviously, without network coding at node R in Figure 2.17(b), the throughput is still 0.5. In contrast, when network coding is enabled at node R, we get a throughput of $4/9$, which is less than 0.5. Hence, from these two examples, we can see that by lowering the transmission rate to increase the reliability of the overhearing links and the coding opportunities, only in some situations it helps improve network performance. The rate adaption can also be applied to a more complicated two-hop relay-based scenario to increase coding opportunities, of which details can be found in [111].

Figure 2.17: Examples of XNC with rate adaption.

For the two-hop coding structures (e.g., those in Figure 2.9(b), (c), and (d)) where overhearing links are involved, the authors in [108] propose a framework for *each node* in the coding structures to choose an optimal rate among a set of available rates. It is assumed in [108] that the packet delivery probability of each link, including the overhearing link, continuously varies with the transmission rate. Nodes in this framework choose the transmission rate as follows. First, *each previous-hop node* of the encoding node is only allowed to choose the rate such that the packet delivery probability of each overhearing link is higher than a certain threshold. Note that this threshold is used to ensure the reliability of the overhearing link or ensure XNC to be performed at the

encoding node. Second, the *encoding node* will choose a rate such that the expected network throughput is maximized. For more details, readers are referred to [108]. The simulation in [108] shows XNC with the rate selection scheme for each node usually leads to higher network performance than that with fixed rate for each node.

Remark on XNC with rate adaption: although the work surveyed here targets those simple coding structures, mainly two-hop coding structures, we can also apply the rate adaption to a more general multi-hop network by viewing the multi-hop network as a combination of many two-hop coding structures and applying the technique to each formed two-hop coding structure. The work in [108] has shown the applicability of the rate adaption to a multi-hop network.

2.7 Drawbacks of Applying XNC

Although XNC benefits network performance in most situations, there exist some scenarios where applying XNC is detrimental to network performance. We present in this Section some drawbacks of applying XNC in wireless networks.

Higher transmission power in XNC: as we know, due to signal attenuation, a higher power threshold is generally required for a sender to transmit a packet to a receiver which is further away. Thus, when an encoding node in XNC broadcasts a packet to a set of receivers, it needs to select a higher power level to ensure that the furthest receiver can correctly receive the broadcast packet [112]. As a result, the use of a larger transmission power will bring stronger interference to other transmissions, which may harm network performance. One illustrative example used in [112] is shown in Figure 2.18(a), where node B is the encoding node for f1 and f2. It is assumed in [112] that the transmission power threshold from node B to node A is larger than that from

node B to node C, i.e., $P_{BA} > P_{BC}$, and that the use of P_{BA} at node B will impose strong interference on the transmission from node G to node F. Thus, when node B is using P_{BA} to broadcast an encoded packet, node G needs to refrain from transmitting. Other conflicting transmission pairs are connected in the figure by the dashed lines. Suppose one packet transmission takes one time slot, then, the optimal schedules under XNC and the non-encoding scheme are constructed and shown in Figure 2.18(b) and (c), respectively. Obviously, we see that the non-encoding scheme here can deliver the same amount of traffic in a shorter period.

Figure 2.18: The negative impact of the higher transmission power required in XNC.

Although the authors in [112] point out the disadvantage of applying XNC shown above, they fail to specify other general situations in which performing XNC at an encoding node is indeed detrimental to network performance.

Lower link rate selection in XNC: suppose that an encoding node needs to broadcast one encoded packet to two or more receivers, and that the links to the receivers have different maximal link rates. Then, for correct reception at all receivers, the

encoding node needs to transmit at the lowest link rate among these rates [113-115]. This requirement actually has been mentioned when we introduce the work in [110] above. Using a lower rate to broadcast the packet is very likely to harm network performance: intuitively, if a link rate is much lower than others, performing XNC at the encoding node will yield a lower network performance. We here employ the work in [114] for illustration. In Figure 2.19(a), each link is labeled with a maximal capacity, and we are interested in the regions of (R_0, R_2) , where R_0 (R_2) is the bit rate for the traffic from node 0 (node 2) to node 2 (node 0), under XNC and the non-encoding scheme, respectively. This rate region problem can be formulated as a simple linear program [114], of which the details are omitted here. Figure 2.19(b) and (c) show the regions of (R_0, R_2) under the two schemes when $C_{12} \neq C_{10}$. From the two figures, we can find that when $C_{12} > C_{10}$, it is not possible for XNC to support a (R_0, R_2) with $R_0 > 1/(1/C_{01} + 1/C_{21})$, and that when $C_{12} < C_{10}$, it is not possible for XNC to support a (R_0, R_2) with $R_2 > 1/(1/C_{12} + 1/C_{21})$. Therefore, blindly applying XNC to a network where links differ from each other in the capacity is risky.

Figure 2.19: The negative impact of the lower link rate selection in XNC.

2.8 Conclusion

As the first part in this Chapter, we give a brief review of some basics of MANETs, including mobility models, the traditional MANET routing, and the DTN routing, which will serve as the basis for the design of the routing schemes under group mobility in the next two Sections.

Regarding the mobility models, we present RWMM and RPGM to illustrate how nodes move in entity mobility and in group mobility, respectively. Both RWMM and RPGM are the mobility models widely adopted to study the performance of the MANET routing schemes.

Regarding the traditional MANET routing, we revisit various routing strategies in connected MANETs. In MANETs under entity mobility, we present the table-driven routing, DSDV, and the on-demand routing, DSR, to illustrate how a shortest end-to-end route is discovered and then used by a new incoming flow. In MANETs under group mobility, we use LANMAR as an example to show how a hierarchy can be formed in group mobility and how the hierarchy can be used to facilitate the route setup for a flow. Then, we particularly point out both DSDV and DSR suffice in a small scale MANET.

Regarding the DTN routing, we review several DTN routing strategies based on the store-carry-forward approach in a disconnected MANET or in a DTN. These strategies mainly include the deterministic routing, the flooding based routing, the flooding controlled routing, and the mobility control based routing. In particular, we point out the pros and cons of each routing strategy and find that each strategy only suits for some certain scenarios. However, we here emphasize that all these DTN routing strategies are proposed under entity mobility.

As the second part in this Chapter, we revisit XNC in wireless ad hoc networks, which will finally serve as the performance booster for the routing schemes to be designed in the next two Sections.

Our review focuses on surveying various XNC conditions and other important factors that can be considered to make better use of XNC under reliable links and under lossy links, respectively. For XNC under reliable links, we first enumerate various situations or coding structures where XNC is applicable. Then, based on these coding structures, we present how to increase coding opportunities by considering the scheduling and the flow route selection. Meanwhile, we also emphasize that the optimal network performance is generally achieved by trading off some coding opportunities for a more load balanced network. For XNC under lossy links, we mainly focus on how the technique of the rate adaption can be properly used to maximize the benefit of XNC. In fact, this rate adaption can also be combined with XNC under reliable links to improve network performance.

Besides, we also present some drawbacks of applying XNC, i.e., the higher transmission power and the lower link rate in the broadcast of the encoded packet. This reminds us that blindly applying XNC poses a risk in a scenario where the transmission power is adaptable or the capacities of the links are different from each other.

Lastly, we here emphasize that all the works surveyed above on XNC target connected and static wireless networks.

Chapter 3

Packet Replication based Routing

Protocols under Group Mobility

In this Chapter, we propose the first two routing schemes, group-epidemic routing (G-ER) and leader based group routing (LBGR), to achieve the communications in MANETs under isolated groups. G-ER focuses on how to guarantee the packet delivery but with limited network resources such as node buffer space, whereas LBGR focuses on how to preserve the advantages of G-ER and further overcome the drawback of G-ER under group dynamics. The rest of this Chapter is organized as follows. Section 3.1 discusses the suitability of each aforementioned DTN routing strategy for group mobility. Section 3.2 presents the network model in our study for the two routing schemes to be proposed. How G-ER and LBGR work is given in Section 3.3 and Section 3.4, respectively. Section 3.5 gives the performance studies of G-ER and LBGR by simulation in various network conditions. Finally, Section 3.6 concludes this Chapter.

3.1 Introduction

In Chapter 1 and Chapter 2, we have pointed out that group mobility could be extracted in many practical scenarios, especially in those cooperative scenarios, and that any two groups are likely to be isolated or disconnected for long time in the network. In fact,

when any two mobile groups are isolated, the network becomes a DTN, and thus the store-carry-forward approach can be adopted to deliver packets in the network.

Obviously, the DTN routing strategies in Chapter 2 targeting entity mobility can be applied in MANETs under group mobility. However, we find that with some proper mechanisms the flooding based routing or the strategy of ER well suits group mobility. The reason is twofold. On one hand, although ER, as mentioned in Chapter 2, is a resource-consuming protocol, especially in the network with large node number, it can achieve optimal performance given the adequate resources, such as the node buffer space and bandwidth. On the other hand, a MANET under group mobility presents a unique feature that the network is formed by a number of sub-networks or groups. Particularly, each group is connected most of time as nodes in a group normally stay close to each other. We can make use of this feature to treat each group as a *single* unit and then disseminate *one copy of one packet to each group* instead of each node. As a result, the sensitivity of ER to the resource in terms of node buffer could be largely alleviated in group mobility, because the group number is generally much less than the node number in the network and much fewer packet copies will be generated. In fact, with the above mechanism in group mobility, we not only can solve the main problem of ER, but also can preserve the advantage of ER, as will be seen from group-epidemic routing (G-ER) to be proposed in this Chapter.

We propose two important mechanisms in G-ER to achieve high network performance but with limited node buffer space. The first is the mechanism we mentioned above. Similar to [59, 60], by treating each group as a single union, G-ER is a two-layer routing scheme, including the inter-group routing and the intra-group routing. The strategy of ER is adopted for the inter-group routing to exchange packets between

groups, and a table-driven routing is employed for the intra-group routing to handle the packet delivery inside the group. Moreover, we employ the group based packet exchange to expedite the packet delivery in this two-layer routing: when two nodes from two groups meet each other, other members in these two groups will also take this chance to deliver packets to the neighboring group. In contrast, the packet exchange in all DTN schemes mentioned above takes place only between two contacting nodes, which we refer to as the node-based packet exchange. The second mechanism is the buffer sharing inside a group, which states that each node's buffer can be shared by all other nodes in its group. This is actually a form of cooperation inside the group.

However, from the simulation study we will find that the performance of G-ER is degraded under group dynamics where group members leave their current groups and join others. Hence, leader based group routing (LBGR), an enhanced version of G-ER, is further proposed, aiming to preserve the advantages of G-ER and overcome its drawback under group dynamics.

3.2 Network Model

Before elaborating G-ER, we describe the network model in this Section.

1) Network Components: We assume there are N groups of nodes moving within an $L \times L$ (m^2) network area independently. The group mobility considered here is built based on a general model, RPGM. Each group comprises M nodes, including one leader and $M - 1$ group members, and has a circular group coverage with the radius, R_g (m). Each node is equipped with a radio transceiver with the transmission range r (m).

2) Group movement: We allow all nodes in one group to move within the whole group coverage, that is, we assign the reference point of each node at the logical center of

the group and set $R_g = R_r$ (see Section 2.1.2 for details). Further, we assume each leader is located at the logical center, and the logical center of each group moves conforming to RWMM, with the speed ranging from $[V_{min}, V_{max}]$ and the pause time to be t_p . By continuously choosing a random destination point and a random speed for the logical center of each group, all nodes in a group are led by the logical center to move as a whole.

3) Group dynamics: Since group dynamics is not included in RPGM, we here propose and integrate a simple group dynamics mechanism into it. In this mechanism, we assume each group leader will not be changed, and the group member in one group may join other groups. When a member is in contact with a new group, it will generate a probability, P_r , uniformly distributed between 0 and 1. If P_r is larger than a predefined group split threshold, Th_{gs} , and its current group has at least G_{min} nodes, then it will leave its current group and follow the new group to move. This group dynamics model is general and applicable to many real scenarios, e.g., those shown in VTGM in the military field [21] and those in the wild animal tracking system [3, 4].

One may note that the group formation in our network model assumes that nodes are grouped in a predefined way. That is, which nodes are grouped together and in each group which node is selected as the leader are predetermined. This is true and reasonable in many real scenarios. For example, in military operations, the leader or the commander and its soldier members should be first designated before the troop goes to the battlefield.

3.3 G-ER for Group Mobility

In this Section, we present G-ER in detail, including the intra-group routing, the inter-group routing, the node buffer management, and the buffer sharing mechanism.

3.3.1 The Intra-Group Routing Strategy

The intra-group routing is used to route a packet to a destination or an intermediate node within a group. That packet can be generated from a particular node and destined to another node within the same group, which is referred to as the *intra-group packet*. Or that packet can be generated from a particular node and destined to another node in a different group, which is referred to as the *inter-group packet*. The intra-group packet will be delivered to the destination node by the intra-group routing. The inter-group packet is first routed to its destination group by the inter-group routing and then, it will be routed to the destination node by the intra-group routing. One important issue here is how a node could determine whether a received packet is destined to a node in its group or to a node in other groups. This can be achieved by checking the destination of that packet from the node's member list, $List_{member}$, which is used to record all nodes in the same group. Thus, if the destination of a packet appears in $List_{member}$, it will be routed by the intra-group routing. Otherwise, the packet will be buffered.

Due to the full connectivity within a group, any routing protocol aimed for connected networks, such as the table-driven routing or on-demand routing, could be used for the intra-group routing. However, it is more suitable to use the table-driven routing because it is reasonable to assume the size of each group in reality is within few hops for the ease of cooperation. Hence, in G-ER, DSDV is adopted for the intra-group routing. Note that each node is required to include its group membership, i.e., group ID (GID), in its DSDV routing update, and only the node with the same GID receives this routing update. Moreover, by receiving this routing update, each node can update its $List_{member}$ in case that some new member joins.

3.3.2 The Inter-Group Routing Strategy

The inter-group routing is used to forward an inter-group packet to its destination group which contains the destination node of that packet. Different from ER, which floods each packet to each node, G-ER disseminates a packet to each group rather than each node. G-ER handles all nodes belonging to a group as an *individual node* and thus, each group will store only *one copy* of each inter-group packet. Consequently, the resource-consuming problem in ER could largely be mitigated. The inter-group routing consists of three procedures as explained in the following sub-sections.

3.3.2.1 Group Summary Vector Update Broadcast within a Group

Similar to SV in ER as shown in Chapter 2, group summary vector (GSV) is used to determine different received packets between two groups when they meet each other. However, different from SV, which only records the received packets for one node, GSV records all packets received by all nodes within a group, not only one node. Hence, GSV at each node is actually the concatenation of SV from all nodes in its group. Furthermore, only inter-group packets need to be recorded in GSV because intra-group packets will not be exchanged between groups.

(1) Each group leader initiates GSV update broadcast, instead of letting each member simply broadcast the update within the group. The reasons are explained as follows. First, we consider IEEE 802.11 Media Access Control (MAC) standard to be used in G-ER, in which the hidden terminal problem exists or no RTS/CTS exchange is used for broadcast. Thus, letting each member simply broadcast GSV update will increase the possibility of collisions between DSDV routing update and GSV update, which will impact the intra-group routing performance. Hence, to reduce the collision

possibility in G-ER, only the leader is required to initiate this broadcast. Second, it is most likely that the leaders in reality are able to keep in touch with their members most of time. Thus, the GSV update broadcast initiated by the leaders can reach their members quickly. Hence, when one member updates its GSV after receiving some new packets, it will send a packet to its leader by unicast to inform these newly received packets. Then, upon receiving this packet from the member, the leader will update its GSV and broadcast this updated GSV to others in its group. This GSV update broadcast includes the updated GSV and the IDs of all 1-hop neighboring members of the leader. This neighbor information is used to help members determine whether they should rebroadcast the received GSV update, which we will see below.

(2) Once a member receives the GSV update broadcast from the leader, it will rebroadcast it only if it has some new 1-hop neighboring members which are not recorded in the received GSV update packet. If the member decides to rebroadcast the GSV update, it will also append its 1-hop neighboring members to the GSV update packet. Hence, we can see that this 1-hop neighbor information is used to avoid the unnecessary rebroadcasting so as to save the overhead incurred in the GSV update broadcast. Figure 3.1 illustrates this GSV update broadcast inside a group in G-ER.

Figure 3.1: GSV update broadcast in a group.

3.3.2.2 GSV Exchange and Group based Packet Exchange

When two nodes from two groups meet each other, the packet exchange will take place. G-ER involves with other group members for packet exchange between two contacting groups to expedite the packet delivery. This is so-called the group based packet exchange, and it is described as follows.

(1) GSV is exchanged between two contacting nodes to determine which packets should be exchanged between them. Here, each node in G-ER is required to send its local GSV to the other.

(2) Once GSV from another group is received by a node, e.g., node **A**, it will be broadcast within the same group to nodes **F** and **G** as shown in Figure 3.2(a) for the preparation of the data packet exchange later. GSV broadcast from **A** is similar to the GSV update broadcast in previous sub-section that the 1-hop neighbor information is included to avoid the unnecessary rebroadcast. Note that G-ER does not require each member in the two contacting groups to successfully receive the GSV from the neighboring group.

(3) By comparing the received GSV to the local GSV, the two member nodes in contact, e.g., nodes **A** and **B** in Figure 3.2, determine which packets should be sent to each other. Since some packets which are supposed to send to another group may not be stored in these two member nodes, they will be sent by other group members in the two contacting groups where these packets are stored. For example, in Figure 3.2(b), node **A** may not store some packets which are supposed to send to node **B**, but nodes **F** and **G** store them. Then, upon receiving the broadcast GSV from node **A** in the second step above, nodes **F** and **G** will forward those needed packets to node **A**. The packet forwarding between nodes **A** and **F** or **G** is based on DSDV. After node **A** receives these

packets, it can forward them to node **B** or any other member in contact in group **Y**. By contrast, in most previous DTN routing schemes including those state-of-the-art schemes [59, 60], packets are exchanged only between two nodes that are in direct contact. Hence, with our group based packet exchange, packets can be delivered in an expedited way.

(4) The GSV update procedure is different for the intermediate group and the destination group. When a new packet arrives at a node of an intermediate group, that node will first update the GSV by adding the new received packet into its GSV and then inform the new received packet to its leader later. However, when a new packet arrives at a node of the destination group, the node's GSV will not be updated until that packet is received by the destination node using the intra-group routing. Since the newly received packet could be dropped during the intra-group routing, it is safer to allow only the destination node to update its GSV. That is, if the newly received packet is dropped during intra-group routing, the destination group would have a chance to receive the same packet again from other groups later.

Figure 3.2: Main procedures in the inter-group routing in G-ER.

In fact, in some situation the principle of one copy of an inter-group packet for each group could be violated, which means the same packet could be sent to the same

group for more than once. For example, if several member pairs from groups **X** and **Y** are in contact simultaneously, the members in these two groups will be requested for packet transmission for multiple times, which will cause one packet to be transmitted repeatedly. To deal with this problem, we propose a method to guarantee the dissemination principle in G-ER by recording the traversed groups in the packet header. Whenever a packet from group **X** is ready to be sent to another group **Y** in contact, the GID of group **Y** will be recorded in the packet header. Thus, the same packet will not be sent to group **Y** again by group **X** because members from group **X** can find out that the packet has been sent to group **Y** already by checking the packet header. Moreover, the information of traversed groups will be used for the buffer management later.

Similar to ER, G-ER implements a time duration to avoid frequent GSV exchange between two groups. It is simply achieved by letting each node record the time of last GSV exchange with any other group. If the time difference between the current time and the last exchange time is less than this duration, the GSV exchange will not be initiated between these two groups. For fairness, we also set this duration 5 s for both G-ER and ER.

3.3.3 Node Buffer Management

Here, the node buffer management is introduced to G-ER to further guarantee the packet delivery. How to selectively drop packets in G-ER is the main concern of our buffer management. We divide the packets into two types for management. The first type refers to the packet to be routed by the intra-group routing, including the intra-group packet and the inter-group packet which has arrived at the destination group. The second refers to the inter-group packet which has not arrived at the destination group. Two strategies for

the buffer management are described as follows.

The first strategy gives higher priority to the first packet type to be kept in the buffer than to the second packet type. The reasons are described as follows: (1) The first packet type could have a better chance for the final delivery because most of time there would be available routes for them using the intra-group routing. (2) Once this type of packet is dropped, especially the intra-group packet, it will permanently be lost, as opposed to the second packet type.

The second strategy is to first drop the packet in the second type which has the highest probability to be delivered to the destination. We sort the packets of this second type with the number of the traversed groups in a node's buffer. The packet with a highest number of traversed groups will be dropped first, because the higher number of traversed groups for a packet means that more copies of that packet are stored in the network and thus, the higher probability that it could have been received by the destination node. Hence, dropping this kind of packet first will not detriment network performance. Since in G-ER the information of traversed groups is recorded in the packet whenever the packet is forwarded to other groups, this information can be used in the buffer management.

3.3.4 Buffer Sharing Inside the Group

The buffer sharing mechanism inside a group is proposed based on another feature of the group. That is, the members inside a group share the same interests or tasks and they are willing to cooperate with each other. Thus, nodes in one group can share their resources like the buffer to improve network performance. The buffer sharing inside a group works effectively in the scenario where group members have different buffer utilizations, which

we will see in the simulation. It is implemented as follows:

(1) Each group member or leader will periodically broadcast its buffer availability information to all other nodes in its group. This information will be simply included in the DSDV routing updates. The buffer statuses of other nodes learnt by a member or leader are also included in its routing updates. In order to prevent the stale buffer state information, each broadcast contains a sequence number, similar to that used in DSDV routing update.

(2) Once a node receives the buffer update from a group member or the leader, it will record the latest buffer availability of these nodes included in the received update.

(3) When a node detects its buffer is full and a new packet is received, it will send the packet at the tail of its buffer to another member or the leader which has the largest available buffer space by DSDV. If such a member or leader cannot be found, the packet will be dropped eventually.

3.3.5 G-ER under Group Dynamics

Now we describe how G-ER reacts to group dynamics. Suppose a member, say node A, intends to leave its current group, say group X, and join a new group, say group Y. Then, the main aim here is to let the leave of node A be known to both group X and group Y.

Step 1: once node A generates the probability, P_r , larger than the threshold, Th_{gs} , it sends a short message, M_{leave} , by DSDV to its leader to ask for the leave permission.

Step 2: once the leader receives the packet in step 1, it will check the current number of nodes in its group by referring to its $List_{member}$. If the number of nodes is larger than G_{min} , the leader will allow that member to leave by sending a short “yes” packet using DSDV. Otherwise, a “no” packet will be replied. If the leave of the member is

allowed, then, the leader will delete this member in its *List_{member}* and trigger the DSDV routing update to broadcast this updated *List_{member}* to its members. Moreover, due to the leave of node A, the leader and its members will check their buffers whether there are any intra-group packets destined to node A. If yes, all these intra-group packets will be treated as the newly generated inter-group packets and stored in their buffers.

Step 3: if the “yes” packet is received by node A, it will first unload all carried inter-group packets to its neighbors in group X. Thus, the copies of the packets can be still stored in group X. Then, it will clear its current GSV and *List_{member}*, and send a short message to inform the leader of group Y of its joining. Meanwhile, node A changes all previously stored intra-group packets to be the newly generated inter-group packet.

Step 4: if the leader of group Y receives the joining message from node A, it will trigger the routing update to inform its members of this new joining member. Once node A receives the DSDV routing update from group Y, it will add all group Y’s members in its *List_{member}*. Also, the leader Y will trigger the broadcast of GSV update, so that the new *joining* node A could initialize its GSV and start to function as a member of group Y. Similarly, the leader of group Y and its members change the stored inter-group packets destined to node A to become intra-group packets.

From the four steps above, we can see that except the packet unloading in step 3, the overhead incurred in the group dynamics is negligible because only the short messages such as M_{leave} and M_{join} are required to be sent. Moreover, G-ER requires only two groups, like groups X and Y, to know the group membership change of any leaving node such as node A. However, in [25, 59, 60], this group membership change needs to be known to all nodes in the network. Thus, G-ER can reduce the overhead incurred in group dynamics.

3.4 LBGR for Group Mobility

As mentioned above, the packet unloading from the leaving group members is needed in G-ER in case of group dynamics, which actually will impact the performance of G-ER. The reason is as follows. Recall that this packet unloading is required to guarantee the principle of one inter-group packet copy for each intermediate group in G-ER, but it is highly possible that it will incur the packet drop. Specifically, the unloaded packets might be dropped due to buffer overflow, and especially, this packet drop happens more frequently when node buffer space is small or when members leave the groups frequently. Obviously, this packet drop will violate the aforementioned principle in G-ER and impact G-ER.

To overcome this drawback of G-ER, LBGR is proposed. Imagine that if there is a policy that the group members with smaller buffer size or fewer stored packets are given a higher priority to leave, then it will not only require fewer packets to be unloaded, but also decrease the drop probability of these unloaded packets. Thus, this policy will benefit network performance eventually, and it motivates us to propose LBGR.

LBGR inherits most strategies in G-ER, including the intra-group routing, the inter-group routing, and the node buffer management. However, LBGR presents the following two own features: (1) it proposes the leader-dominating inter-group routing, which requires that each inter-group packet received by a node to be stored at the buffer of the group leader of that node, and that packets are only exchanged between two group leaders when any two nodes in the two groups meet; and (2) to support this inter-group routing, LBGR adopts the buffer resource allocation in each group.

3.4.1 The Intra-Group Routing

The intra-group routing in LBGR is the same as that in G-ER. We omit the details here.

3.4.2 The Leader-Dominating Inter-Group Routing

We start introducing the inter-group routing in LBGR with differentiating responsibilities between the leader and the member in LBGR. For each group leader, it is required to (1) store all inter-group packets in its buffer, (2) determine the *different* inter-group packets between its group and other group in contact, and (3) exchange those different packets to that other group. However, the group member is mainly required to send the inter-group packets received from other groups to its group leader. In addition, each newly generated inter-group packet at each member also needs to be sent to the group leader for storing. Then, the following three steps, triggered by the contact of two members from two groups, describe how packets are processed in the leader-dominating inter-group routing.

Step 1: GSV exchange between two contacting groups.

1. Similar to G-ER, the group summary vector (GSV) is used in LBGR to record all inter-group packets received by each group. In LBGR the leader will store received inter-group packets in its group, so GSV is managed and updated only by the leader.

2. When two members from two groups meet, each one will send a short message called GSV request to its leader and ask its leader to send GSV to it. Here, we refer to each of these two members as the requester. This request, with quite limited size, will be routed by DSDV to reach the leader. Once the leader has received this request, it will send its GSV or a GSV packet to the requester by DSDV.

3. After each requester receives the GSV packet from its leader, it will forward it to the member in contact from another group. Lastly, these two requesters will forward

the received GSV from each other to their leaders by DSDV. Figure 3.3(a) shows the scenario when node A in group X meets node B in group Y . Node A first requests its leader, node X , to send GSV to it. Then, node A will forward GSV to node B , and node B will further send GSV to its leader, node Y . Also, the leader X will receive the GSV packet from the leader Y in the similar way.

Figure 3.3: GSV and data packet exchange in LBGR when two groups meet.

Step 2: Data packet exchange between two groups.

After each leader receives the GSV packet from the other group, it will compare the received GSV with its own GSV. If the leader finds some different packets, i.e., the packets that have not been received by the other group and are still stored in its buffer, it will send them to this other group. Firstly, those packets will be delivered to the requester, like node A or B . Then, node A or B will forward them to any member in contact from this other group. In Figure 3.3(b), once Y completes the GSV comparison to X 's, it will forward the different packets to node A in group X via node B .

Step 3: The transfer of the received packets to the group leader.

When an inter-group packet is received by a group member, like node *A* or *B*, it will be further delivered by DSDV to the group leader, where it will be stored. After receiving the new inter-group packet, the group leader will update its GSV. As shown in Figure 3.3(b), after node *A* receives the packets from node *B*, it will forward them to its leader *X*. Then, *X* will store them in its buffer and update its GSV.

Figure 3.4: Major operations for the group member in LBGR.

Figure 3.5: Major operations for the group leader in LBGR.

Figure 3.4 and Figure 3.5 summarize the major operations for the member and

leader in LBGR, respectively. Basically, the leader and the member react only when they detect a new contact with a new group or receive a ‘new’ packet. Here, we omit the procedures involved in the DSDV routing table broadcast and update within a group in the flowchart. Thus, the “new” packet refers to the intra-group packet, the inter-group packet, the GSV packet, or the GSV request.

Hence, we see from the above steps that the leader plays a critical role in LBGR in storing and forwarding packets. Thus, to support the leader to function effectively in LBGR, we employ the resource allocation in each group such that most of the resources such as the buffer space and the power are allocated to the leader. In addition, we also have the following two procedures to have LBGR better handle the packet forwarding.

First, we consider how an inter-group packet is handled when it arrives at the destination group. This inter-group packet will be directly forwarded to the destination node by DSDV. The destination node then will inform its leader of this newly received packet for the GSV update at the leader. In order to avoid too frequent informing, the destination node will inform its leader about its newly received packets after T_{inform} s.

Second, similar to G-ER, LBGR also implements a time duration to avoid frequent GSV exchange between two groups. However, this is only applied to the group leader in LBGR, because only the leader manages GSV in LBGR. Here, this time duration is also set at 5 s the same as that in G-ER. For example, if leader X in Figure 3.3(a) is requested to send multiple GSVs to group Y by receiving multiple GSV requests within 5 s, it will only respond to the first GSV request.

3.4.3 Node Buffer Management in LBGR

The node buffer management in LBGR is basically the same as that in G-ER.

Specifically, the two management strategies in G-ER are still applicable to the group leader in LBGR. However, in LBGR each group member only needs to store the intra-group packet and the newly generated inter-group packet. The former needs to be stored when the route to the destination node is not available, and the latter needs to be stored when the route to the group leader is not available. These two kinds of packets are treated equally at each group member, because either will permanently be dropped in case of buffer overflow, which is detrimental to network performance.

3.4.4 LBGR under Group Dynamics

The four steps involved in G-ER to deal with group dynamics are basically still applicable to LBGR. However, since in LBGR no inter-group packet is stored in the group member's buffer, so there is no packet unloading in LBGR when a member is about to leave for a new group. Also, since the member does not manage GSV in LBGR, all procedures related to GSV in G-ER are not required in LBGR.

Hence, we can find from the steps under group dynamics that (1) the overhead incurred in the leave of the member is very limited in LBGR, and (2) the leave of the member neither causes packet drop due to no packet unloading in LBGR nor impacts the “carry” capacity of its group due to its very small buffer space as compared to the leader. Hence, LBGR will be resilient to group dynamics, as will be shown in the simulation.

3.4.5 Complexity of LBGR

We now analyze the communication complexity of LBGR, the enhanced version of G-ER. The communication complexity refers to the number of control messages needed for the operation of a protocol [128]. Here, we focus on the analysis of the intra-group

routing and the inter-group routing in LBGR, respectively. First, LBGR employs DSDV as the intra-group routing, and hence we analyze the number of the affected nodes, denoted by x , that need to trigger a broadcast of the routing table in case of a link failure or addition [128]. Since DSDV is only performed within each group in LBGR, a link failure or addition within a group will trigger M nodes, the number of node per group, to broadcast their routing tables at most, resulting in a communication complexity of $O(x = M)$. Second, the inter-group routing in LBGR mainly involves the dissemination of the data packet and the exchange of GSV packet as shown in Section 3.4.2. Considering an inter-group packet originated from a group member, the maximal transmission times in LBGR for this packet to reach its destination node are $\lceil R_g/r \rceil + (2\lceil R_g/r \rceil + 1)(N - 2) + (\lceil 2R_g/r \rceil + 1)$, where the three components denote the transfer to the leader in the source group, the flooding to leaders of other groups but the destination group, and the delivery to the destination group and to the destination node, respectively. Similarly, the number of exchanged GSV packets can be calculated as $2 \times (2\lceil R_g/r \rceil + 1)(N - 1)$. Thus, the communication complexity for the inter-group routing in LBGR is given by $O(6dN + 3N - 5d)$, where $d = \lceil R_g/r \rceil$, the group radius in terms of the hop number.

3.5 Simulation Results

Now we study the performances of G-ER and LBGR in different scenarios by simulation in NS-2. G-ER and LBGR are mainly evaluated under the following three parameters:

- *Node buffer size, b* : since G-ER and LBGR, similar to ER, are two packet replication or flooding based DTN routing schemes, we are interested to see how sensitive they are to b . Here, b is in terms of the number of the data

packet. Therefore, to show how much they can lower the buffer requirement for the delivery of all packets, ER is used for comparison [11, 41, 48].

- *Group coverage radius*: since the performance of the intra-group routing in G-ER and LBGR, i.e., DSDV, is affected by the group scale, we are interested in how DSDV performs in different group scales. We will adjust the group radius, R_g , to form two different group scales in terms of the maximal hops in each group, i.e., the one-hop group and the multi-hop group. Similarly, to show how DSDV performs, DSR, which is the benchmark in the performance evaluation of traditional MANET routing protocols, is used for comparison.
- *Group structure stability*: given a uniform distribution of P_r from 0 to 1, we will set different values of Th_{gs} to form different group structure stabilities. Thus, we can study how G-ER reacts to different group structure stabilities and how LBGR improves G-ER under group dynamics.

We design the simulation according to the network model described in Section 3.2. First, Table 3-1 shows the default values for the parameters in our network model. In addition, IEEE 802.11 standard is chosen as the MAC for G-ER and LBGR, and the radio transmission range for each node is 100m, which is small compared to the network size, 1000m. The data rate is fixed at 2Mbps. The radio propagation model used here is a determinant model, i.e., the two-ray ground model, which does not take channel fading into account. Second, the traffic model used here is similar to that used in [7, 59]. However, we adapt it to group mobility to generate both the intra- and inter-group traffic to evaluate the performances of the intra-group routing and the inter-group routing. In this model, 30 nodes from three groups take turn to send packets, with each sending two packets to the other 29 nodes. The packet generation rate for all source nodes is 4 packets

per second. Third, for each point of the results below, the simulation is run for a number of times to make it within 90% confidence interval $\pm 10\%$.

Table 3-1: Simulation parameters.

<i>Parameters of Network Components</i>			
network area ($L \times L$)	1000m $\times$ 1000m	group number, N	5
group radius, R_g	50/150m	node number per group, M	10
<i>Parameters of Group Movement</i>			
pause time, t_p	20s	group speed, $[V_{min}, V_{max}]$	5m/s-15m/s
<i>Parameters of Group Dynamics</i>			
G_{min}	5	Th_{gs}	1.0
<i>MAC Layer Parameters</i>			
MAC	IEEE 802.11	data transmission rate	2 Mb/s
carrier sensing range	300m	radio transmission range, r	100m
<i>Other Parameters</i>			
radio propagation model	two-ray ground model		

3.5.1 Performance of G-ER without Buffer Sharing

3.5.1.1 G-ER at Different b and R_g without Group Dynamics

Figure 3.6: Intra-group packet delivery ratio.

The simulation results including the 90% confidence interval for G-ER under the setting shown in Table 3-1 are shown from Figure 3.6, to Figure 3.10. Figure 3.6 shows the

intra-group packet delivery ratio, r_{intra} . First, it can be seen that r_{intra} of all these three routing schemes decreases with the increasing group radius, R_g , because the number of hops for the intra-group packet increases with R_g . Most importantly, G-ER basically performs similar to DSR at both $R_g = 50$ and 150 m. This shows that in spite of the incurred overhead in G-ER, like the GSV update broadcast, the intra-group routing of G-ER could still perform effectively. Second, we can observe that compared to ER, both G-ER and DSR are much less sensitive to the buffer size, because a complete route for the intra-group packet in G-ER and DSR could be found most of time in a connected group.

Figure 3.7: Inter-group packet delivery ratio.

Figure 3.7 shows the inter-group packet delivery ratio, r_{inter} . First, as expected, it can be seen that G-ER, as compared to ER, provides higher r_{inter} even with a small buffer size. Second, it can be observed that r_{inter} of G-ER and DSR is more or less constant when the buffer size is greater than 50, whereas ER is much more sensitive to the buffer size. This is because G-ER forwards only *one copy of one packet to one group*, and DSR does not need to store the copies of the packets. Therefore, G-ER and DSR require lower buffer size to support the packet delivery. Third, the performances of ER and DSR

increase with larger R_g , but the performance of G-ER is independent of R_g . As R_g increases, the network becomes more connected, and thus more routes are available for DSR in a more connected network to deliver packets to destination. For ER, as R_g increases, there would be more contact opportunities between two nodes in two different groups, and hence packets can be more easily delivered to the destinations. For G-ER, due to the low buffer requirement and the efficient intra-group routing even in a multi-hop group, it could deliver most inter-group packets to keep high r_{inter} in most cases.

Figure 3.8: Overall packet delivery ratio.

Figure 3.8 shows the overall packet delivery ratio, r_o , of these three routing protocols, which is the combination of r_{intra} and r_{inter} . For DSR and ER, both perform better at a larger R_g , whereas G-ER always keeps high r_o except in the case of $b = 10$. Overall speaking, G-ER is able to delivery all packets for both radii with very limited buffer size, but the performance of ER severely depends on both buffer size and group radius. For example, at $R_g = 150$, only when $b = 400$ ER can achieve the optimal r_o .

Figure 3.9 shows the average intra-group packet delay, d_{intra} , for these three routing protocols. Obviously, both G-ER and DSR basically have much less d_{intra} than that of ER at each R_g . This is because DSDV in G-ER and DSR could provide the

available route for intra-group packet most of time. The reason for long d_{intra} in ER is that the period (5 s) introduced to ER to avoid the frequent SV exchange actually delays its packet delivery to some extent. Hence, from the similar performance between DSDV and DSR in Figure 3.6 and Figure 3.9, we find that DSDV is appropriate for the intra-group routing of G-ER when the group size is small or within several hops.

Figure 3.9: Intra-group packet delivery delay.

Figure 3.10 shows the inter-group packet delay, d_{inter} . First, it can be seen that d_{inter} for these three routing schemes decreases when R_g increases because, as explained in Figure 3.7, a larger R_g provides more opportunities for packet forwarding between nodes in different groups, and thus the inter-group packet can reach the destination group and be delivered at the destination node faster. Second, similar to Figure 3.7, d_{inter} of G-ER and DSR is basically independent of buffer size when b is larger than 50. Interestingly, we can observe that regardless of R_g , when $b < 200$, G-ER has longer delay but higher packet delivery ratio (as shown in Figure 3.7) as compared to ER. However, when $b > 200$, G-ER not only outperforms ER in the packet delivery ratio, but also outperforms ER in d_{inter} . This indicates that the group based packet exchange in G-ER can deliver packets to the destinations faster, because packets stored in those members

who are not in contact with the neighboring group could also have the chance to be exchanged. Hence, the group based packet exchange in G-ER can expedite the packet delivery as compared with the node based packet exchange in ER.

Figure 3.10: Inter-group packet delivery delay.

3.5.1.2 G-ER under Group Dynamics

Here, we evaluate the performance of G-ER at different group structure stabilities, corresponding to different Th_{gs} . The optional value of Th_{gs} will be 1.0, 0.95, 0.9, and 0.85, and P_r , as mentioned above, is uniformly distributed from 0 to 1. We fix the group radius R_g at 50 m, and run the simulation at different b and Th_{gs} with other parameters remaining the same as used in above.

Figure 3.11 shows the performance of G-ER at different b and Th_{gs} in terms of r_o . First, r_o of ER and DSR are independent of Th_{gs} because they are only sensitive to the network connectivity, which basically has nothing to do with Th_{gs} . Second, at smaller Th_{gs} , which implies a more dynamic group structure, the performance of G-ER is degraded. With a more unstable group structure, more group members are likely to leave their groups at the same time when two groups meet, so more packets, which are stored

in these leaving group members, need to be unloaded to the rest nodes in the same group. Thus, these transferred packets will be dropped easily due to buffer overflow. This is the main reason for the degraded performance of G-ER at lower Th_{gs} . Hence, the improvement of G-ER over ER gradually diminishes as the group structure becomes less and less stable. Third, it can be seen that for a small b , the performance degradation of G-ER with Th_{gs} becomes faster. This is because with a small b at each node, the packets unloaded by those leaving nodes will be dropped more possibly due to buffer overflow. Therefore, from Figure 3.11 we can find that in our solution the larger buffer size could better counteract the negative influence of the group dynamics to G-ER.

Figure 3.11: Overall packet delivery ratio of G-ER under group dynamics.

3.5.2 Performance of G-ER with Buffer Sharing

In order to study the effect of the buffer sharing mechanism, most of the simulation parameters remain the same except the traffic model. Here, we assume the groups will generate unbalanced traffic as follows:

- Three groups are selected to generate traffic, but only half in each group will generate packets. This is to mimic the real scenario where group members

may have different tasks or different traffic amounts within a group.

- No intra-group packet will be generated, because these packets will not be buffered most of time. Each source node will send packets to all 40 nodes from other four groups; each with 4 packets. In total, there will be 2,400 inter-group packets generated. The time interval for these packets is also 0.25 s.

Figure 3.12: Performance of G-ER with and without buffer sharing.

Figure 3.13: Average node buffer utilization in G-ER at $R_g = 50$.

Figure 3.12 shows the packet delivery ratio of G-ER with buffer sharing and without buffer sharing under this new traffic model. From Figure 3.12, it can be seen that

the buffer sharing improves the performance of G-ER. The improvement is greater for $R_g = 50$ m. That means the effect of the buffer sharing stands out in a more disconnected network, because without buffer sharing, more packets permanently are dropped due to the buffer overflow in a more disconnected network. Besides, Figure 3.13 shows the average buffer utilization of one group which generates traffic for $R_g = 50$ m. Obviously, in G-ER without buffer sharing, the source nodes have higher buffer utilization compared to the non-source nodes. However, with buffer sharing, the buffer utilization for non-source nodes increases significantly. Interestingly, for large b , i.e., $b = 200$, the buffer utilization for non-source nodes is still low. This is because $b = 200$ is large enough for the source nodes and thus, the buffer space from the non-source members is not needed.

We have also studied the effect of the buffer sharing under the default traffic, finding that it does not provide much improvement to G-ER. This is because all nodes in one group basically have the same buffer utilization. Hence, we see that our buffer sharing is beneficial in case that nodes in one group have different buffer utilizations.

3.5.3 Performance of LBGR without Group Dynamics

Now we proceed to study the performance of LBGR under different network conditions with no group dynamics. Since LBGR additionally employs the resource allocation in each group, we first describe it as follows. For simplicity, we only consider the buffer resource allocation in each group. In LBGR each leader is simply assigned 80% of the total buffer size of the group, and other group members equally share the rest. For example, suppose in our simulation the average node's buffer size, b , is set at 200, then the group leader is given the buffer size of 1600, and each of the 9 members is given the buffer size of 44. In fact, the percentage of the buffer size for the leader can be varied to

optimize the performance. However, we leave this as the future work. Note that this configuration is used throughout the studies for LBGR.

3.5.3.1 LBGR under The Default Traffic Model

The following studies how LBGR reacts to different b and R_g under the default traffic model shown in Table 3-1.

First, we evaluate the packet delivery ratio and the packet delivery delay in LBGR. Because LBGR inherits most strategies in G-ER, we find that LBGR performs very close to G-ER under the same b and R_g regarding the intra-group packet delivery ratio, the inter-group packet delivery ratio, the intra-group packet delivery delay, and the inter-group packet delivery delay. For example, compared to ER, LBGR guarantees the packet delivery with much less buffer space, and it delivers inter-group packets faster due to the group based packet exchange mechanism; and similar to DSR, LBGR delivers the intra-group packet quickly due to the use of DSDV in the intra-group routing of LBGR. Hence, LBGR preserves the advantages of G-ER shown in Figure 3.6, Figure 3.7, Figure 3.9, and Figure 3.10. We here do not show the results of LBGR in figures.

Second, we investigate the energy consumption in LBGR by comparison to that in ER. We mainly investigate the energy consumption at the leader in LBGR because of its critical role in the packet exchange. The consumed energy is in terms of the number of data packet transmission. We ignore the energy consumption caused by Hello message transmission in ER and the periodical routing update broadcast in LBGR, because the periods for them are set similarly in our simulation. The average number of data packet transmission in LBGR and ER at $R_g = 50$ and $R_g = 150$ are shown in Table 3-2 and Table 3-3, respectively, from which we have two observations as follows.

Table 3-2: The packet transmission number in LBGR and ER at group radius 50

Buffer size, b	LBGR			ER	
	At each leader	At each member	Total	At each node	Total
$b = 50$	1217	258	17695	1108	55400
$b = 100$	1302	265	18435	1307	65350
$b = 200$	1370	280	19450	1693	84650
$b = 400$	1320	283	19335	2100	105000

Table 3-3: The packet transmission number in LBGR and ER at group radius 150

Buffer size, b	LBGR			ER	
	At each leader	At each member	Total	At each node	Total
$b = 50$	1494	404	25650	1118	55900
$b = 100$	1511	426	26725	1348	67400
$b = 200$	1739	455	29170	1708	85400
$b = 400$	1846	459	29885	2043	102150

(1) With respect to the total energy consumption in terms of the total number of data packet transmission, LBGR consumes lesser energy than ER, but it can still achieve higher performance such as the shorter packet delivery delay as discussed above. For example, when $R_g = 50$ and $b = 400$, the consumed energy in LBGR is only about 1/5 of that in ER. In particular, only the energy consumption of each leader in LBGR is comparable to that of each member in ER, and the number of leaders is quite small, 5 in our simulation, so LBGR could save a large amount of energy as compared to ER. This shows that by making use of the group structure in group mobility, LBGR greatly alleviates the energy-consuming problem of ER. Note that we also find from the simulation that the total number of GSV packet transmission in LBGR is much lesser than that in ER, which further differentiates the total energy consumption in LBGR from

that in ER. This is because the GSV exchange is performed between groups in LBGR whereas it is performed between nodes in ER.

(2) In LBGR each leader consumes more energy than each member. However, as compared to each member in ER, the leader in LBGR still consumes less energy, especially at large buffer size, b . As expected, due to the significant role of the leaders in LBGR they consume more energy than members. Therefore, in order to make the leaders not become the bottleneck of the network, it is necessary to allocate more energy to the leaders. Although the performance of ER is approaching LBGR at large b (which can be seen in Figure 3.6 and Figure 3.7), referring to both Table 3-2 and Table 3-3, each leader in LBGR consumes lesser energy than each member in ER at $b = 400$; especially when $R_g = 50$. This is because the energy consumed by each leader in LBGR is mainly determined by the number of the inter-group packets. However, the energy consumed by each member in ER is determined by the number of all data packets, including both the intra-group and inter-group packets.

3.5.3.2 LBGR under A Demanding Traffic Model

Besides the study under the default light traffic, we here further study LBGR under a more demanding traffic model. In this model, a source node from one group continuously generates traffic to its destination node from some other group to form an inter-group flow. We assume each inter-group flow lasts for 200 seconds and the flow rate is 2 packets per second. We randomly choose the source node and the destination node of an inter-group flow and generate different numbers of flows to study LBGR at $b = 200$. In particular, in this study we replace DSR with a more recent routing protocol called enhanced cluster based routing protocol (ECBRP) [8] for comparison. In [8], it shows

that ECBRP employs clustering to effectively reduce the impact of the node mobility on routing performance and thus outperforms DSR in many different network conditions, e.g., under either group mobility or entity mobility.

Figure 3.14 shows how LBGR reacts to the number of flows at $R_g = 150$. First, we see that ECBRP achieves low packet delivery ratio at $R_g = 150$. Similar to DSR, ECBRP also attempts to look for a complete end-to-end path for a new incoming flow, which is generally not available in a disconnected network at $R_g = 150$. This further shows that any attempt to search for an end-to-end path for a flow in a DTN environment will turn into a failure. Second, as the number of flows increases to 15, LBGR is also not capable of handling the constantly generated heavy traffic. However, we can see that under this demanding traffic model, LBGR is less impacted compared to ER. The reasons are as follows. On one hand, based on G-ER, LBGR largely reduces the number of packet copies compared to ER. On the other hand, LBGR aggregates most of the buffer resource of each group at the group leader, thus utilizing the buffer resource maximally.

Figure 3.14: Overall packet delivery ratio of LBGR under the demanding traffic.

3.5.4 Performance of LBGR under Group Dynamics

Here, how LBGR reacts to group dynamics is studied. We are particularly interested in examining whether LBGR is much more resilient or robust to group dynamics than G-ER, so here the performance of LBGR is only compared to that of G-ER. In this study, we will vary the average node buffer size, b , and the threshold, Th_{gs} , to see how the two factors interactively affect LBGR. Here, we still use the same traffic model as used in the study of G-ER under group dynamics.

Figure 3.15 shows the overall packet delivery ratios of LBGR and G-ER at $R_g = 50$. Obviously, we see from this figure that the performance of LBGR is basically not affected by group dynamics at both $b = 50$ and 200, and it outperforms G-ER at any b or Th_{gs} . Thus, these results show that LBGR is much more resilient to group dynamics than G-ER, and that the leader-dominating inter-group routing and the resource allocation in LBGR can effectively overcome the weakness of G-ER under group dynamics.

Figure 3.15: Overall packet deliver ratio of LBGR under group dynamics.

3.6 Conclusion

In this Chapter, we have proposed two new DTN routing schemes called G-ER and

LBGR under group mobility. By making full use of the topology feature of the group mobility, two significant mechanisms (i.e., the principle of one inter-group packet copy for one intermediate group and the buffer sharing within a group) are proposed in G-ER to greatly lower the resource requirement for the packet delivery. In G-ER, the table-driven routing is employed for the intra-group routing, and the strategy of ER or the principle of one inter-group packet copy for one intermediate group is adopted for the inter-group routing. Moreover, the group based packet exchange is integrated into this two-layer routing to expedite the packet delivery, which finally makes G-ER outperform ER and DSR in different scenarios. The simulation results in NS-2 show that the first objective listed in Chapter 1 is attained by G-ER.

However, the performance of G-ER will be degraded in case of group dynamics, especially when node buffer space is small. Hence, we propose LBGR to specifically overcome this weakness of G-ER. LBGR inherits most strategies in G-ER, e.g., it employs the table-driven routing in the intra-group routing, so it preserves most advantages of G-ER under no group dynamics. Moreover, LBGR features the leader-dominating inter-group routing and the resource allocation in each group, which can effectively cancel the impact from group dynamics. Our simulation shows that LBGR can effectively improve G-ER under group dynamics. The simulation also shows the leader consumes more energy than the member in LBGR, but it still consumes lesser energy to ensure the packet delivery as compared to each node in ER.

In particular, we emphasize that both G-ER and LBGR are applicable to various group mobility models, not limited to RPGM. This is because they make no assumption on the movement of the groups (or they can work no matter how each group moves).

Chapter 4

Mobility Control in MANETs under Isolated Groups

4.1 Introduction

In the previous Chapter, we propose G-ER and LBGR in MANETs under isolated groups, and the simulation studies show that both G-ER and LBGR can achieve high network performance such as the packet delivery ratio with limited node buffer space or limited node energy consumption. In fact, these benefits of G-ER and LBGR come from the strategies related to group mobility. That is, the common strategy in G-ER and LBGR is to treat each group as a single unit and apply the strategy of ER to the groups instead of the nodes. However, like those flooding based routing schemes or flooding controlled routing schemes mentioned in Chapter 2, the common disadvantage of G-ER and LBGR is that the packet delivery delay in both schemes may be unpredictable or very long.

As mentioned earlier in Chapter 2, employing the node mobility control can effectively increase contact opportunities among nodes and speed up the packet delivery in a DTN, thus mitigating the unpredictability problem of the packet delivery delay in the flooding based routing or the flooding controlled routing. For example, in [13, 55, 58], the ferry node is introduced to the network, and it moves from a source node to a destination node whenever the source node has packets to send to the destination node.

Obviously, with the help of the ferry node, the packet delivery delay becomes predictable. However, in these solutions [13, 55, 58], the ferry node should be deployed in a centralized manner. As a second example, the authors in [12] propose a mobility control based routing which requires no special node like the ferry but requires the mobile nodes themselves to move purposely to create new connections between a pair of nodes. Although this solution in [12] can shorten the packet delivery delay to some extent, it is built on a strict assumption that how each node moves in the network is known to all other nodes in advance.

Hence, as seen from the previous DTN studies (e.g., [12]) above, the mobility control is a promising technique to shorten the packet delivery delay in a DTN, but the difficulty lies in its deployment. The reason for this difficulty is that the mobility control based routing requires some intermediate nodes or ferry nodes to move in a cooperative way. However, most studies above target entity mobility which assumes independence among nodes' movement. By contrast, group mobility is actually very likely to be a ground to well support the node mobility control. This is because in some cooperative activities nodes in one group are willing to achieve a certain goal through cooperation in which the mobility of some nodes could be controlled. Some typical examples can be found in the disaster and emergency relief and the military field. Therefore, in this Chapter, we further exploit the group mobility by considering the node mobility control for the packet delivery in MANETs under isolated groups.

We propose two mobility control techniques under group mobility; the group formation enforcement and the group purposeful movement. The group formation enforcement requires some nodes in one group to be positioned strategically such that these nodes can help extend the reachability of the group. Thus, this technique can be

used to increase the contact opportunities between groups. The group purposeful movement considers creating a new connection between two groups by letting some nodes in one group move purposely to approach the other group which is currently out of contact. The new created connection provides a way for one group to deliver packets to the other. Hence, with this second technique, one group can expand its reachability and create more contact opportunities with other groups. Importantly, both two proposed mobility control techniques can be easily integrated into some existing DTN routing scheme to effectively expedite the packet delivery or reduce the packet delivery delay. In this Chapter, we focus on the study of how each proposed technique reduces the packet delivery delay in a DTN. To the best of our knowledge, none of the previous studies has considered the mobility control for the packet delivery in a DTN under group mobility.

More specifically, our work in this Chapter can be summarized as follows.

- We propose two simple mobility control techniques in DTN under group mobility; the group formation enforcement and the group purposeful movement. Both two techniques can be integrated into some existing DTN routing scheme to effectively expedite the packet delivery.
- For each proposed mobility control technique, we integrate it into G-ER, and then employ the stochastic analysis to derive the cumulative density function of the packet delivery delay in a simple network condition to show the reduced packet delivery delay.
- Particularly, we put emphasis on the performance study of the group purposeful movement mobility control technique under different network conditions. We firstly design a new DTN routing scheme called purposeful movement assisted routing (PMAR), in which the group purposeful

movement is implemented. Then, we integrate PMAR into G-ER and conduct extensive simulations to show the effectiveness of PMAR in reducing the packet delivery delay of G-ER in different practical network conditions

The rest of this Chapter is organized as follows. In Section 4.2, we give the stochastic analysis of how the two proposed mobility control techniques reduce the packet delivery delay, respectively. Then, PMAR is illustrated in Section 4.3, followed by the simulation study in Section 4.4. Lastly, Section 4.5 concludes this Chapter.

4.2 Analysis of the Two Mobility Control Techniques

In this Section, we first present the network model for our analysis. Then, based on this model, we give the stochastic analysis of how the two proposed mobility control techniques reduce the packet delivery delay of G-ER in MANETs under isolated groups, respectively.

4.2.1 Network Model

In this analysis, we suppose that there are a total of $N + 1$ ($N \geq 1$) independent groups moving within a $L \times L$ (km^2) network area, and that the radio transmission range of each node is r (in km). Other important assumptions related to the network model are stated as follows.

First, we illustrate how each group in the network is formed. The group mobility model used here is RPGM, but we assume that (1) each group is organized by a leader, and the leader locates at the logical center of the group; (2) the reference points of all members in one group also locates at the logical center of the group, and all members move randomly within the group coverage; and (3) each member is within the radio

transmission range of the leader, r . Thus, each group assumed here has a circular coverage with a radius of d ($d \leq r$) (in km), as shown in Figure 4.1.

Figure 4.1: The assumed group structure.

Second, we present how each group moves in the network area. The movement of each group conforms to RWMM, as introduced in Chapter 3. Here, we omit the details of the procedures for the movement of each group but point out the following important parameters related to the group movement. The velocity, v , for each leader is uniformly distributed within $[v_{min}, v_{max}]$ (in km/h), and the pause time for each group or leader is t_p once it reaches its selected destination point.

Third, similar to [4], we assume a simple traffic model for this analysis: among the $N + 1$ groups, only one group, say group x , generates one data packet P1 to send to another group, say group y . Note that this data packet can be generated by any node in group x .

Fourth, we further assume that (1) each group is capable of storing one data packet; and (2) if two groups meet (i.e., if any two nodes from two groups meet), the contact duration is long enough to allow any node in one group to send a data packet to the other group. Thus, if group x meets an intermediate group, say group z , which previously has not received P1, then group z can also obtain P1 after this meeting.

4.2.2 Cumulative Density Function of the Packet Delivery Delay in G-ER

With the abovementioned assumptions and the behavior of each group defined in G-ER in Chapter 3, we firstly derive the lower bound and the upper bound of the cumulative density function (CDF) of the packet delivery delay, T_{GE} , under G-ER when $t_p = 0$ and $r \ll L$. We denote $F_{T_{GE}}$ as the CDF of T_{GE} , and we let $LB(F_{T_{GE}})$ and $UB(F_{T_{GE}})$ represent the lower bound and the upper bound of $F_{T_{GE}}$, respectively. These two bounds will be considered as the benchmark in our paper. To derive the two bounds, we start with the computation of the following three parameters when $t_p = 0$ and $r \ll L$.

- The *inter-meeting time between two leaders*, t_{leader}^{IM} : this is the time interval between two consecutive meetings between two leaders. Note that a meeting between two nodes or leaders means the distance between them is less than the radio transmission range, r . It is shown in [116] that this t_{leader}^{IM} under RWMM is exponentially distributed, of which the intensity, λ_{leader}^{IM} , is given by

$$\lambda_{leader}^{IM} = \frac{2\omega r \mathbb{E}[V^*]}{L^2} \quad (1)$$

In formula (1), ω is a constant with the value about 1.3683; and $\mathbb{E}[V^*]$ is the average relative speed between the two leaders, and its calculation can be found in [116]. Thus, the average t_{leader}^{IM} is $1/\lambda_{leader}^{IM}$ (in hours).

- The *inter-connection time between two leaders*, t_{leader}^{IC} : this is the time interval between two consecutive connections between two leaders. Note that a connection between two leaders means the two leaders are meeting each

other or being connected by some of their members. Thus, we see that 1) this time interval is less than the inter-meeting time between two leaders above, i.e., $t_{leader}^{IC} < t_{leader}^{IM}$, because even the distance between two leaders is beyond r , they can be still connected by their members; and 2) when d approaches 0, t_{leader}^{IC} will approach t_{leader}^{IM} . Moreover, we here consider a new random variable t_{leader}^{ideal} , which is the time interval between two consecutive events that the distance between two leaders is within $r + 2d$. Similar to t_{leader}^{IM} , t_{leader}^{ideal} is also exponentially distributed, of which the intensity, λ_{leader}^{ideal} , is given by

$$\lambda_{leader}^{ideal} = \frac{2\omega(r + 2d)\mathbb{E}[V^*]}{L^2} \quad (2)$$

In fact, the value of $r + 2d$ is the maximal distance between two leaders when they are connected, as shown in Figure 4.2. However, two leaders may not be connected when they are $r + 2d$ apart, because there may happen to be no members near the boundary of the group coverage due to the random mobility of all group members. Hence, we have $t_{leader}^{ideal} < t_{leader}^{IC} < t_{leader}^{IM}$.

- The *inter-meeting time between two groups*, t_{group}^{IM} : this is the time interval between two consecutive meetings between two groups. As assumed above, each group is connected most of time, so if two groups meet each other (i.e., if any two nodes from two groups meet each other), their leaders must be connected, and vice versa. Thus, we have $t_{group}^{IM} = t_{leader}^{IC}$.

Now, we employ Ordinary Differential Equations (ODEs) [117, 118] to derive $LB(F_{T_{GE}})$. With the fourth assumption above, we can see that t_{group}^{IM} actually determines how fast the data packet is spread over groups in the network. This is to say, if two

groups meet frequently (i.e., if t_{group}^{IM} is small), then P1 will be spread quickly. Thus, due to $t_{group}^{IM} < t_{leader}^{IM}$ as shown above, we can derive $LB(F_{T_{GE}})$ by assuming t_{group}^{IM} conforms to the known distribution of t_{leader}^{IM} or by assuming $d = 0$. Suppose the packet delivery delay in G-ER is T'_{GE} under the assumption of $d = 0$, then we have $LB(F_{T_{GE}}) = F_{T'_{GE}}$. To derive $F_{T'_{GE}}$, two steps are involved.

Figure 4.2: Maximal distance in a connection between two leaders.

Step 1: Calculation of $X(t)$. Suppose $X(t)$ is dedicatedly used to count the number of groups (excluding the destination group) which have received the data packet P1 at time t . Then, $X(t)$ can be treated as a Markov chain with the transition rate, $R(t)$, expressed as follows.

$$R(t) = \lambda_{leader}^{IM} X(t) (N - X(t)) \quad (3)$$

For the Markov chain above, we can use the following differential equation to solve it [4, 117]:

$$X(t + \delta t) = X(t) + R(t)\delta t \xrightarrow{\text{yields}} X'(t) = R(t) = \lambda_{leader}^{IM} X(t) (N - X(t)) \quad (4)$$

The differential equation (4) above is separable and can be solved with the initial condition $X(0) = 1$, i.e., only the source group x owns P1 at time 0. Consequently, $X(t)$ is expressed as follows, of which the derivation is omitted here.

$$X(t) = \frac{N}{1 + (N - 1)e^{-tN\lambda_{leader}^{IM}}} \quad (5)$$

Step 2: Derivation of $F_{T'_{GE}}$. Suppose at time t the destination group y has not received P1. Then, let us consider $P(t < T'_{GE} \leq t + \delta t)$, the probability that group y receives P1 in the subsequent small time interval δt . Since there are $X(t)$ groups owning P1 at time t , the rate of group y receiving P1 at time t will be equal to $\lambda_{leader}^{IM} X(t)$. Thus, $P(t < T'_{GE} \leq t + \delta t)$ can be calculated as follows.

$$\begin{aligned} P(t < T'_{GE} \leq t + \delta t) &= P(T'_{GE} \leq t + \delta t) - P(T'_{GE} \leq t) \\ &= P(T'_{GE} > t)P(\text{group } y \text{ receives data packet in time } (t, t + \delta t] | T'_{GE} > t) \\ &= (1 - P(T'_{GE} \leq t))\lambda_{leader}^{IM} X(t)\delta t \end{aligned} \quad (6)$$

Due to $F_{T'_{GE}}(t) = P(T'_{GE} \leq t)$ and $F_{T'_{GE}}(t + \delta t) = P(T'_{GE} \leq t + \delta t)$, we have the following differential equation for $F_{T'_{GE}}(t)$ from equation (6):

$$F'_{T'_{GE}}(t) = \lambda_{leader}^{IM} X(t) (1 - F_{T'_{GE}}(t)) \quad (7)$$

The differential equation (7) is also separable and can be solved with the initial condition $F_{T'_{GE}}(0) = 0$. Finally, we obtain $F_{T'_{GE}}(t)$ or $LB(F_{T_{GE}})$ as shown in the following expression, of which the derivation is omitted here.

$$F_{T'_{GE}}(t) = LB(F_{T_{GE}}) = 1 - \frac{N}{N - 1 + e^{tN\lambda_{leader}^{IM}}} \quad (8)$$

Next, we proceed to derive $UB(F_{T_{GE}})$. Similarly, due to $t_{group}^{IM} > t_{leader}^{ideal}$, we can derive $UB(F_{T_{GE}})$ by assuming t_{group}^{IM} conforms to the known distribution of t_{leader}^{ideal} . Particularly, the steps to derive $UB(F_{T_{GE}})$ are the same as those to derive $LB(F_{T_{GE}})$ shown above. More specifically, we only need to replace λ_{leader}^{IM} with λ_{leader}^{ideal} in equation (8) to obtain $UB(F_{T_{GE}})$, which is given by the following equation.

$$UB(F_{T_{GE}}) = 1 - \frac{N}{N - 1 + e^{tN\lambda_{leader}^{ideal}}} \quad (9)$$

4.2.3 The Group Formation Enforcement

In the previous sub-section, we have shown the lower bound and the upper bound of $F_{T_{GE}}$. In fact, it is highly impossible to achieve the upper bound of $F_{T_{GE}}$ because of the random mobility of the group members around the leader in each group. Here, we propose a simple mobility control technique called the group formation enforcement to approach the upper bound of $F_{T_{GE}}$. The group formation enforcement imposes a constraint on the mobility of some members in each group such that a certain structure or formation appears in each group. Suppose each group leader leads at least four members, then the formation enforcement works as follows.

- Each leader randomly selects four members to locate them at the east, south, west, and north boundary of the group coverage as shown in Figure 4.3(a). And each of the four selected members is only allowed to move closely around the corresponding boundary.
- The other rest members can still move freely within the group coverage.

Obviously, with this formation enforcement technique, the reachability of each group will be extended. More specifically, if we fix the four selected members above at the four boundary positions, then, any node within a distance of $d + r$ to the leader is very likely to be connected to the leader, as shown in Figure 4.3(a). Thus, as shown in Figure 4.3(b), it is also very likely that two leaders within a distance of $2d + r$ can be connected, which indicates that $t_{leader}^{ideal} \approx t_{leader}^{IC}$ or $t_{leader}^{ideal} \approx t_{group}^{IM}$. Consequently, with the group formation enforcement in G-ER, the upper bound of $F_{T_{GE}}$ given by equation (9) can be approximated.

Figure 4.3: The extended group reachability and the connection between two leaders under the formation enforcement.

To show how the upper bound of $F_{T_{GE}}$ is approximated under the formation enforcement, we run simulation in NS2 to show the CDF of T_{GE} in the case of $d \approx r$. As a comparison, we also conduct simulations by deactivating the formation enforcement or allowing all members to move freely around the leader. We emphasize that all simulation results shown here are collected in the ideal case that the contact duration between two meeting groups is long enough to allow one data packet to be sent from one group to the other. Table 4-1 shows the main simulation parameters.

Table 4-1: Simulation parameters in the group formation enforcement

L	4km	d	0.1km	r	0.1km
v_{min}	4km/h	v_{max}	10km/h	t_p	0
Total group number		10	Member number per group		4

Before showing the simulation results of $F_{T_{GE}}$, we firstly determine $UB(F_{T_{GE}})$ and $LB(F_{T_{GE}})$ shown in equations (9) and (8), respectively. It can be seen from equations (8) and (9) that we only need to determine λ_{leader}^{IM} and λ_{leader}^{ideal} under the setting shown in Table 4-1 for $LB(F_{T_{GE}})$ and $UB(F_{T_{GE}})$, respectively. In [116] it has shown that $E[V^*] = 8.7km/h$ under RWMM when $t_p = 0$ and $[v_{min}, v_{max}] = [4, 10]$.

Thus, we get $\lambda_{leader}^M \approx 0.148$ from equation (1) and $\lambda_{leader}^{ideal} = 0.444$ from equation (2). Accordingly, $LB(F_{T_{GE}})$ and $UB(F_{T_{GE}})$ are already determined under our simulation setting, as shown in Figure 4.4.

Figure 4.4: Comparison between simulation results and analytical results for $F_{T_{GE}}$.

After hundreds of simulation times, the collected simulation results of $F_{T_{GE}}$ are shown in Figure 4.4. It can be observed that 1) with the group formation enforcement, G-ER delivers packets faster than that without mobility control, and 2) the group formation enforcement is an effective technique to improve G-ER to approach $UB(F_{T_{GE}})$.

4.2.4 The Group Purposeful Movement

Besides the technique of the group formation enforcement, we here introduce another mobility control technique called the group Purposeful Movement (PM). Due to the cooperative nature of each group mentioned earlier, the leader can command some of its members (maybe robots and hereafter called PM members) to move towards some other

group to create a new connection between them when this other group is not far away but currently out of connection. For example, as shown in Figure 4.5, when the leader X detects that group Y is currently three hops away, it can let two members, nodes I and J, to move fast to approach group Y. Obviously, if a connection can be created between group X and group Y, it can expedite the packet delivery. This is the motivation for this second mobility control technique in this Chapter.

Figure 4.5: Connection creation in the purposeful movement of a group.

In this sub-section, we integrate this PM into G-ER to form a new scheme called G-ER+ and analytically study how this PM reduces the packet delivery delay of G-ER. For simplicity, we have the following three assumptions for this study.

- (1) The radius, d , of the group coverage considered in this study is very small. As a result, $F_{T_{GE}}$ approaches $LB(F_{T_{GE}})$ and can be given by equation (8) approximately.
- (2) The maximal number of PM members is M ($M \geq 0$). As a result, it is likely that a connection can be set up between two groups if the two leaders are within a distance of $(M + 1)r$, as shown in Figure 4.5. For simplicity, we assume that as long as two leaders are within a distance of $(M + 1)r$, a new connection can always be established, and a data packet can be delivered from

one group to the other.

- (3) Only the source group, i.e., group x , will carry out the PM for establishing a new connection to the destination group, group y . Other intermediate groups are not required to execute the PM to deliver the packet from the source group.

With the three assumptions above, we now proceed to derive the CDF of the packet delivery delay, T_{PM} , in G-ER+. We denote $F_{T_{PM}}$ as the CDF of T_{PM} . Similarly, the derivation of $F_{T_{PM}}$ also involves two steps as shown below.

Step 1: Calculation of $X(t)$ in G-ER+. Here, $X(t)$ is defined the same as before. Because there is no PM between any two of the groups counted by $X(t)$ in G-ER+, the derived $X(t)$ in G-ER+ is very close to that in G-ER in the condition of $d \approx 0$. Thus, $X(t)$ in G-ER+ can be given by equation (5) approximately.

Step 2: Derivation of $F_{T_{PM}}$. Similar to the derivation of $F_{T'_{GE}}$ shown above, we need to firstly determine the rate of group y receiving P1 at time t . Since at time t there are $X(t)$ groups owning P1, including the source group, the total rate of group y receiving P1 is given by $(M + 1)\lambda_{leader}^{IM} + \lambda_{leader}^{IM}(X(t) - 1)$, where the first term is contributed by the source group and the rest is contributed by all intermediate groups, of which each owns the packet P1. Consequently, we obtain the following equation

$$\begin{aligned} P(T_{PM} \leq t + \delta t) - P(T_{PM} \leq t) &= (1 - P(T_{PM} \leq t))((M + 1)\lambda_{leader}^{IM} + \lambda_{leader}^{IM}(X(t) - 1))\delta t \\ &= (1 - P(T_{PM} \leq t))(M + X(t))\lambda_{leader}^{IM}\delta t \end{aligned} \quad (10)$$

Thus, from equation (10) we obtain the differential equation (11) as follows.

$$F'_{T_{PM}} = (1 - F_{T_{PM}})(M + X(t))\lambda_{leader}^{IM} \quad (11)$$

The differential equation (11) is also separable and can be solved with the initial

condition $F_{T_{PM}}(0) = 0$. We omit the details of the derivation and show $F_{T_{PM}}$ as follows.

$$F_{T_{PM}}(t) = 1 - \frac{Ne^{-tM\lambda_{leader}^{IM}}}{N - 1 + e^{tN\lambda_{leader}^{IM}}} \quad (12)$$

We now present the following three observations based on equation (12).

First, to achieve the packet delivery delay T_{PM} within some certain value of t at a given probability, we can set a minimal M , M_{min} , properly. For example, under the network setting shown in Table 4-1 but with $d \approx 0$, Table 4-2 shows several small values of M_{min} for achieving different values of t with $F_{T_{PM}}(t) \geq 0.8$. A larger M_{min} not shown in Table 4-2 is actually not practical because we expect that it is not possible to create a connection between two groups far apart due to the group mobility.

Table 4-2: M_{min} under network setting shown in Table 4-1 but with $d \approx 0$

$F_{T_{PM}}(t) \geq 0.8$				
t (in hours)	[2.375, 2.711)	[2.092, 2.375)	[1.852, 2.092)	[1.647, 1.852)
M_{min}	1	2	3	4

Second, the proposed PM technique works more effectively in a DTN with small N . That is, when N is small, the average packet delivery delay of G-ER will be reduced more by the PM technique. We use one example as shown in Figure 4.6 to show this observation. The results, i.e., the delay reduction of G-ER+ over G-ER, in Figure 4.6 are obtained based on equations (8) and (12) at different N . It can be seen that the average packet delivery delay of G-ER is reduced more by the PM technique when N is small. The main reason for this observation is that when N is small, the network is more disconnected and thus, it is more likely that the packet highly relies on PM technique to be delivered to the destination.

Figure 4.6: The delay reduction of G-ER+ over G-ER at different N .

Figure 4.7: The delay reduction of G-ER+ over G-ER at different λ_{leader}^{IM} .

Third, given fixed M and N , the delay reduction of G-ER+ over G-ER does not vary with λ_{leader}^{IM} . This actually can be derived from equations (8) and (12). However, for simplicity, we also use one example to show this observation. Figure 4.7 shows the delay reduction of G-ER+ over G-ER at different λ_{leader}^{IM} but with fixed M and N . Obviously, it can be seen that for any given M and N , the delay reduction keeps constant regardless of λ_{leader}^{IM} . Thus, the analysis here shows that the effectiveness of the PM

technique is mainly determined by M and N .

4.3 Purposeful Movement Assisted Routing

From the previous Section, we can see that the group formation enforcement is just a mobility control technique and it works independently of the DTN routing. However, the PM technique, as shown in Figure 4.5, further requires some specific mechanisms to initiate the purposeful movement of a group and deliver packets effectively along the newly established connection. Hence, in this Section we propose a new routing scheme called purposeful movement assisted routing (PMAR) to implement the PM technique. Then, by integrating PMAR into G-ER, we can study how it improves G-ER by simulation under different practical conditions.

First, we present the following two assumptions in PMAR: (1) each group member and each leader in the network is equipped with Global Positioning System (GPS); and (2) each leader is able to increase its transmission power to have a longer transmission range, R , in addition to the normal transmission range, r . Second, we present the two major objectives that PMAR needs to achieve; to establish a new connection between two disconnected groups if necessary and to deliver packets along the new established connection.

Now we proceed to introduce how PMAR is executed in a cooperative way in a group. There are eight major steps in PMAR as listed below: the first five steps are taken for the first objective, and the steps six and seven are taken for the second objective.

The first five steps used to create a new connection between groups are illustrated as follows.

1) Condition detection for executing PMAR. We here consider one factor for

triggering PMAR; the buffer utilization of the group. In particular, we only consider the buffer utilization of the inter-group packet generated by the group itself, and we set a threshold, B_t , for this buffer utilization to activate PMAR. The PM is organized by the leader, so the leader needs to firstly collect the buffer state information of its members. How to collect the member buffer state information for the leader will be described later. Once the leader detects that the inter-group packets mentioned above occupy B_t of the group buffer space, PMAR will be triggered. Specifically, before triggering PMAR, the leader needs to find from the collected buffer state information a destination group to which its group has the most number of packets.

2) *Searching for the destination group.* Once the destination group is found in *Step 1*), the leader of the source group proceeds to find the current location of that destination group. The group leader simply broadcasts a short message, PM_{search} , which contains the destination group ID, using the radio of the transmission range R . If the leader of the destination group receives this broadcast, it immediately replies a short message, PM_{reply} , which contains its current location, to the source group leader using the radio of the transmission range R . If the source group leader does not receive the reply from the destination group, it waits for a time interval, W_t , to go back to *Step 1*). The time interval W_t is used to avoid searching too frequently for the leader, and it is called scanning interval in this paper.

3) *Location determination for members to establish the new connection.* If the source group leader receives PM_{reply} from the destination group, it knows that a new connection can be set up. Thus, it needs to determine the number of PM members for establishing the new connection and the locations for them to move to. The new connection to the destination group will be set up as shown in Figure 4.5, in which the

two end points of the new connection are the leader **A** of the source group and any node in the destination group **B**. In particular, all PM members are arranged linearly between two leaders, and the distance between any two neighboring PM members is about r , as shown in Figure 4.5. Therefore, the number, N_{pm} , of the PM members required for this new connection is given by $N_{pm} = \lfloor D/r \rfloor$, where D is the current distance between the two leaders. Meanwhile, the location (X_n, Y_n) of the n th PM member, say H_n , can be found as follows.

$$X_n = X_l + n r \cos \theta ; n = 1, 2, \dots, N_{pm}; \quad (13)$$

$$Y_n = Y_l + n r \sin \theta ; n = 1, 2, \dots, N_{pm}; \quad (14)$$

where (X_l, Y_l) is the current location of the source group leader, and θ is the moving direction of the PM members as shown in Figure 4.5. In fact, this calculated location for each PM member is adjustable within some certain range to keep the two groups connected.

4) *PM member selection.* After calculating the location of each PM member, the source group leader needs to select members to execute the P. To ensure a successful connection, the best is to select members which can move fast. For simplicity, we assume that there are some members in each group dedicated to carrying out the PM.

5) *PM member moving to its designated location.* Once the leader has selected the PM members, it broadcasts a short message, PM_{start} , to those selected PM members to command them to move to the designated locations. We assume that each PM member, guided by its GPS, moves towards the designated location at a speed of S_{pm} . In the meantime, the leader stops moving forward until the PM ends as shown in Step 7), and the rest members, upon receiving PM_{start} , also stop moving forward but to move around the leader to keep the group connected. A connected group is necessary in many

scenarios, e.g., in the military operation, soldiers in the same group should normally stay together most of time to lower the probability to be attacked.

After the five steps above, a new connection could be established provided that the destination group does not go too far away. Subsequently, the packet delivery along this new connection (i.e., the second objective in PMAR) takes place.

Figure 4.8: Connection notification in PMAR.

6) *The packet delivery in PMAR.* Once the new connection to the destination group is established, the PM member who is in contact with the destination group sends a message called $PM_{connect}$ to inform its leader as illustrated in Figure 4.8. Then, the leader broadcasts a message to its members to request them to forward their packets to the destination group along the new created connection. For simplicity, we set a time duration, $t_{connect}$, for the packet delivery in this step. When the set time expires, those PM members will start returning.

7) *Termination of PMAR.* A simple method to terminate the new connection in PMAR is to let the PM members start returning to their original group coverage after they leave the group region for a period of t_{PM} . This t_{PM} is the sum of the time spent by the PM members on moving to their designated locations and the time required for the packet delivery in Step 6). Thus, t_{PM} is simply given by

$$t_{PM} = t_{connect} + N_{pm} r / S_{pm} \quad (15)$$

Note that this estimated t_{PM} is also contained in the broadcast message PM_{start} in *Step 4*).

8) *Movement recovery of the source group.* Once each PM member moves back into the current group coverage, the source group leader will be informed, and finally the whole group can continue moving forward.

Note that the communication complexity of PMAR mainly includes the number of both PM_{reply} and PM_{search} . This is because among all short control messages used in PMAR, the broadcast of PM_{reply} and PM_{search} uses the long radio transmission range, thus consuming much more energy than the others. In general, the number of PM_{reply} is much less than the number of PM_{search} , and the value of W_t determines the number of PM_{search} . In fact, a proper W_t should be set in relation to the parameters of group mobility, e.g., node speed, and we will discuss this in the simulation study.

Now we illustrate how the proposed PMAR is integrated into G-ER to cooperate for the packet delivery. First, each group member in G-ER is required to report its buffer state to its leader: whenever a member is ready to broadcast its periodic DSDV routing update in G-ER, its current buffer state is collected and carried in the routing update packet. This member buffer state mainly states the number of inter-group packets destined to each other group. Again, only the information of the inter-group packet generated by the group itself is collected in this report, because one group is not responsible for carrying out the PM for other groups. Second, with this member buffer state information, a group leader can determine whether to initiate PMAR as introduced in *Step 1*). If PMAR is not triggered in one group, then, that group will follow the procedures in G-ER to exchange packets with other groups. Hence, from the interaction between PMAR and G-ER, we can see that PMAR can be treated as an independent module to boost the performance of G-ER.

4.4 Simulation Study and Results

4.4.1 Simulation Model

The simulation study in this Section is conducted in NS-2 to show how PMAR improves G-ER under different practical network conditions, namely under different searching range, R , different group speed, and different group buffer size. First, for the searching range, R , two values are selected; $2r$ and $3r$. Second, we select the speed range $[v_{min}, v_{max}]$ from three non-overlapped ranges as will be shown later. Third, we vary the total buffer space in each group (i.e., the group buffer size) from an inadequate value to an adequate value. The “inadequate” (“adequate”) means the group buffer size is less (larger) than the total number of the data packet generated at the source nodes. We will study how these three factors interactively affect the effectiveness of PMAR. The main performance metric in this simulation study is the average packet delivery delay, t_d .

There are five groups in a $1km \times 1km$ network area with 9 members and 1 leader in each group, and each group moves conforming to RWMM. The normal radio transmission range, r , for each node is 0.1km. For the traffic model, we let only one group among the five groups be the source group to generate traffic to another group for simplicity. The source group will generate a total of 400 packets: five members in the source group will take turn to generate packets every 50 seconds from the beginning of the simulation; and each member will generate 80 packets at a rate of 1 packet per second. Here, we assume the buffer space of the source group is fixed and it is capable to store all generated packets. Thus, all data packets can be successfully delivered to the destinations in all routing schemes in our simulation. Table 4-3 shows other main default simulation parameters, among which the PM member speed, S_{pm} , is set at a relatively high value, i.e.,

15m/s, as compared to the group speed.

Table 4-3: Simulation parameters.

Total group number	5	Member number in each group	9
Group speed	1-4m/s	Pause time	40s
Group buffer size	100; 300; 500; 700	Radius of the group coverage	0.05km
Normal radio transmission range, r	0.1km		
PMAR searching range, R	0.2km; 0.3km		
Radio transmission rate	2Mb/s		
Data packet size	560 Bytes		
Scanning interval, W_t	15s		
Speed of PM members, S_{pm}	15m/s		
Connection duration in PMAR, $t_{connect}$	5s		
Buffer threshold in PMAR, B_t	30%		

4.4.2 PMAR at Different Group Buffer Sizes and Searching Ranges

Now we study how PMAR improves G-ER under the default parameters shown in Table 4-3 by changing the group buffer size, b , and the searching range, R . Figure 4.9 shows the packet delivery delay, t_d , of G-ER and G-ER with PMAR at different b and R .

First, it can be seen that PMAR reduces t_d of G-ER at all b and R . Particularly, we see that t_d of G-ER is significantly reduced by PMAR with a longer R ($= 3r$). This is because with a longer R , the destination group could be found earlier and thus, the packet can be delivered earlier. Table 4-4 shows the performances in terms of the packet delivery ratio, r_{pmar} , and the packet delivery delay, d_{pmar} , of PMAR at different b and R . Obviously, it can be seen that d_{pmar} at $R = 300m$ is much shorter than that at $R = 200m$. Actually, we will find later that this longer R takes effect only when the group moves at a low speed or when the new connection established in PMAR stays for long time.

Figure 4.9: The average packet delivery delay, t_d .

Second, for each R , as the group buffer size increases, the delay of G-ER is less reduced by PMAR. For example, when $R = 300m$, t_d of G-ER is reduced by about 35% at $b = 100$ whereas it is reduced by about 20% at $b = 700$. There are two reasons for this observation. On one hand, with larger b of each group, G-ER itself delivers packets faster, thus reducing t_d as shown in Figure 4.9. On the other hand, with larger b , the ability of each intermediate group in forwarding the packets of the source group is intensified. As a result, the packet number delivered by PMAR decreases. By comparing the percentage of the packet delivered by PMAR, r_{pmar} , at different b in Table 4-4, we see that r_{pmar} decreases to some extent at large b , especially at $R = 300m$. Consequently, the effectiveness of PMAR gradually diminishes as b increases.

Third, the cost in achieving the benefit of PMAR above is limited. That is, we find in the simulation that the total number of the transmitted PM_{search} and PM_{reply} is about 40 at $R = 300m$ and $W_t = 15s$. Besides, we also find that decreasing W_t increases the total cost, but cannot attain more benefit of PMAR. However, at a longer W_t , e.g., 30s, the total cost decreases, but the benefit of PMAR is reduced. Obviously, a proper W_t

should be determined by the group mobility model. With a large W_t , the source group is likely to miss the opportunity to find the destination group nearby, and a small W_t is unnecessary and increases the energy consumption.

Table 4-4: Performance of PMAR.

Group buffer size, b	Searching range, R			
	$R = 200$		$R = 300$	
	r_{pmar}	d_{pmar} (s)	r_{pmar}	d_{pmar} (s)
$b = 100$	31.5%	257.2	49.5%	173.2
$b = 300$	30.0%	236.8	48.0%	166.9
$b = 500$	27.7%	228.4	47.3%	163.4
$b = 700$	29.5%	224.4	43.7%	155.4

Table 4-5: Performance of G-ER with PMAR at different group speeds

Speed range	Searching range	t_d (seconds)	
		G-ER	G-ER with PMAR
1 – 4 m/s	$R = 200$	334.6	280.7
	$R = 300$		230.6
4 – 7 m/s	$R = 200$	191.8	161.5
	$R = 300$		140.3
7 – 10 m/s	$R = 200$	172.7	170.3
	$R = 300$		168.8

4.4.3 PMAR at Different Group Speeds

We now proceed to study how PMAR improves G-ER at different group speeds, s . In addition to the default speed range shown in Table 4-3, i.e., [1, 4] (m/s), we set two higher speed ranges for this study; [4, 7] (m/s) and [7, 10] (m/s). The simulation here is conducted at a relative high $b = 500$ with other default parameters listed in Table 4-3.

Table 4-5 shows the performance, in terms of t_d , between G-ER and G-ER with PMAR at different ranges of s . Obviously, as s increases, the improvement of G-ER with PMAR over G-ER decreases regarding the packet delivery delay. For example, as s

increases to $[7, 10]$ (m/s), the delay reduction at $R = 300$ even vanishes. There are two reasons for this observation. First, G-ER itself performs more effectively by delivering packets much faster as the group speed increases, as shown in Table 4-5. This is because there are more contact opportunities between any two groups at higher group speed, which allows the packet to be spread to groups faster. Second, the contribution (i.e., the delivered packet number) of PMAR is reduced at high s . One reason is due to the unstable connection established by PMAR at high s . In order to see this instability, we define the following two parameters for the connection stability:

- Pb_{pmar} : the probability of a successful connection in PMAR. After each PM member arrives at its designated location, the destination group may have moved away, which causes an unsuccessful connection.
- r_{pmar} : this is the same metric as defined in Table 4-4. Generally speaking, if the connection established in PMAR lasts for a longer time interval, r_{pmar} will become larger.

Table 4-6: Connection parameters in PMAR at different node speed when $R = 300$.

Speed Range, s	Connection Stability Parameters	
	Pb_{pmar}	r_{pmar}
1 – 4 m/s	1.0	47.3%
4 – 7 m/s	0.92	56.2%
7 – 10 m/s	0.91	26.2%

Table 4-6 shows Pb_{pmar} and r_{pmar} at different s when $R = 300$. Overall speaking, it shows that Pb_{pmar} and r_{pmar} decreases with s . The trend of these two parameters shows that a less stable connection in PMAR is established at a higher group speed. Hence, as the contribution of PMAR diminishes at high s , the improvement of PMAR made to G-ER almost diminishes.

4.5 Conclusion

In this Chapter, we further exploit the group mobility, and propose two mobility control techniques under isolated groups, i.e., the group formation enforcement and the group purposeful movement. Both two techniques are simple but effective in extending the reachability of the group, and they can be integrated into some existing DTN routing scheme under group mobility to expedite the packet delivery. We integrate them into G-ER and analytically show that 1) with the group formation enforcement, the upper bound of the cumulative density function of the packet delivery delay in G-ER can be approached; and 2) with the group purposeful movement, the packet delivery delay in G-ER can be greatly reduced in the scenario where a limited number of groups exist.

We put emphasis on the performance study of the second technique, i.e., the group purposeful movement, under different network conditions. Specifically, we firstly design a new routing scheme called purposeful movement assisted routing (PMAR) to implement this second technique. Then, we conduct simulations in different network conditions to evaluate the effectiveness of PMAR in improving G-ER, showing that PMAR works effectively when the group buffer size is small and the group speed is low.

Despite the advantages of the two proposed node mobility control techniques, they are only applicable in the scenarios where the mobility of some nodes in a group can be controlled. One typical application is in the military filed, e.g., the mobility of the soldiers or the tanks in one group can be controlled by the group leader.

Chapter 5

Studies on Network Coding

Conditions in Wireless Networks

In this Chapter, we study inter-flow network coding (XNC) conditions in wireless ad hoc networks. In particular, our study of XNC conditions in this Chapter assumes that the wireless link is reliable, and that all links have the same link capacity. As surveyed in Chapter 2, previous studies on XNC have pointed out various situations or coding structures in which XNC can be applied to a common intermediate node of multiple flows. Obviously, these surveyed coding structures may also appear among the intra-group flows (of which the source nodes and the destination nodes are within the same group) in G-ER or LBGR, and thus XNC can be applied to G-ER or LBGR for performance improvement.

However, we find that there exist two problems in most previous XNC schemes, and both can be solved for more performance gains in G-ER or LBGR. First, the network coding opportunity might be misidentified within some widely studied coding structures, which is actually detrimental to network performance. Second, the XNC conditions previously proposed are restrictive due to the limitation of overhearing. Overhearing is widely used to acquire the information for the packet decoding in most XNC schemes, but it only takes effect in the one-hop neighborhood of a node, which in turn restricts the

identification of coding opportunities. Hence, before proceeding to apply XNC to G-ER or LBGR, we first put efforts into solving these two problems in this Chapter.

5.1 Description of Problem 1

As seen from Chapter 2, XNC has been widely applied in wireless ad hoc networks to improve network performance. It allows the intermediate node to encode packets from different flows into a single mixed packet and broadcast it to different receivers. If each destination of those flows is able to retrieve the packet intended to it from the mixed/encoded packet, network performance will be improved by reducing the number of transmissions for the packet delivery. Then, to take advantage of XNC, network coding-aware routing [87] has been proposed in wireless ad hoc networks. The routing protocols [83, 101, 102] find coding opportunities and identify encoding nodes between two flows along their routes to improve network performance. However, one main problem of these routing protocols is that they may misidentify coding opportunities between two flows. That is, if one packet from one flow is re-encoded, then, the newly mixed packet may be un-decodable. Here, re-encoding a packet means encoding a packet that was coded previously.

Figure 5.1 shows an example in which the coding opportunity may be misidentified when the distributed coding-aware routing (DCAR) [83] (or even when the coding-oblivious routing scheme COPE [80]) is used. In Chapter 2, we have introduced the network coding conditions in DCAR, that is, DCAR employs the *generalized coding structure* to identify the coding opportunity. Thus, when DCAR is used in Figure 5.1, node 3 is the encoding node for flow 1 (f1) and flow 2 (f2), and node 4 is the decoding node for f1 and encoding node for another flow pair, f1 and flow 3 (f3). In this example,

a packet, $P1$, from $f1$ is encoded at node 3 with another packet, $P2$, from $f2$ to form packet, $P1 \oplus P2$. Packet $P1 \oplus P2$ is then decoded at node 4 to yield $P1$. After that, node 4 encodes $P1$ with $P3$ from $f3$ to form $P1 \oplus P3$. Now, $P1$ is considered a re-encoded packet because it was previously encoded with $P2$ and now encoded with $P3$. However, this is a misidentified coding opportunity, since when the new formed packet, $P1 \oplus P3$, reaches node 7, it will be un-decodable there. This is because node 7 overheard $P1 \oplus P2$ from node 3, not $P1$. We refer to this coding opportunity misidentification as the re-encoding problem, since it only appears in packet re-encoding. This re-encoding problem will also occur in [101, 102]. We omit the details here.

Figure 5.1: Illustrative topology for the failure of network coding.

In fact, the re-encoding problem shown above will emerge more frequently as more flows in the network are intersected with each other. This is because when one flow (like $f1$ in Figure 5.1) can be encoded with other two or more flows (like $f2$ and $f3$ in Figure 5.1) at different intersection nodes, this re-encoding problem may appear. Figure 5.2 shows four flows intersect with each other at four nodes, nodes a , b , c , and d , and each intersection node of two flows is identified as an encoding node in DCAR. It can be seen that all intersection nodes but node a will re-encode the packet, and then, similar to Figure 5.1, the re-encoding problem will arise.

Figure 5.2: A more complicated topology for the failure of network coding.

One way to solve this re-encoding problem in [83, 101, 102] is to avoid misidentifying the coding opportunities in packet re-encoding. To achieve this, we propose a principle called consistency of encoding and overhearing (CEO) for the encoding node to adhere to. The root cause of the un-decodability in Figure 5.1 is that there is an inconsistency between what was overheard at node 7, $P1 \oplus P2$, and what is used to mix with $P3$, $P1$. Hence, for successful decoding, the encoding node needs to be aware that the two packets from two flows to be encoded must be the same two packets overheard by these two flows. This is the principle behind CEO.

Specifically, CEO restricts the behavior of some encoding node(s). For example, if we apply CEO to DCAR in Figure 5.1, node 4 will deactivate the re-encoding of the packet from $f1$ and only encode $f1$'s packets that are not encoded at node 3; whereas the encoding at node 3 will not be affected.

To show the importance of CEO to XNC or network coding-aware routing schemes such as DCAR, we analyze the performance of DCAR and DCAR with CEO (DCAR-CEO) based on the topology in Figure 5.1. Then, to further compare the performance of DCAR and DCAR-CEO in other various scenarios, we first propose a distributed way for the encoding node in DCAR to adhere to CEO, and then conduct

simulations in different scenarios to show the advantage of DCAR-CEO.

5.2 Performance Analysis of DCAR and DCAR-CEO

We analyze the flow throughput of DCAR and DCAR-CEO based on the topology in Figure 5.1 at different flow rates, r_1, r_2, r_3 , of f1, f2, and f3, respectively. We employ the resource allocation or scheduling for the analysis [87]. T_i^j and $T_i^{(j,k)}$ are defined as the time fraction assigned to the unicast transmission from node i to node j and the broadcast transmission from node i to nodes j and k , respectively. This broadcast transmission refers to the encoded packet transmission at node 3 or node 4 in Figure 5.1. For interference-free scheduling, we emphasize that only the transmission from node 0 to node 3 and the transmission from node 2 to node 4 can be scheduled in parallel. In the analysis each link capacity is assumed to be 1, and without loss of generality, f3 is assumed to be the last flow to be established.

DCAR: we only consider the rate combination (r_1, r_2, r_3) within the rate region [114] of DCAR. Then, the throughput of f1 and f2 will be r_1 and r_2 , respectively, whereas the throughput of f3 may be less than r_3 due to the decoding failure at node 7. Hence, we here focus on the throughput computation of f3.

Case 1: $r_1 \leq r_2$. First, before f3 is set up, the resource allocation for f1 and f2 is as follows. For f1, $T_0^3 = T_4^5 = r_1$; for f2, $T_1^3 = r_2$ and $T_3^6 = r_2 - r_1$; for both f1 and f2, $T_3^{(4,6)} = r_1$, because all packets from f1 are encoded with r_1/r_2 packets from f2 at node 3. Thus, due to the non-overlapping scheduling for all transmissions above for f1 and f2, we have the residual time fraction $T' = 1 - 2r_1 - 2r_2$, which could be totally allocated to f3. Second, if f3 is set up with $r_3 \leq r_1$, it will not occupy the residual time fraction, T' , due

to the parallel transmissions mentioned above and the coding opportunities between f1 and f3 at node 4. This is achieved by assigning $T_2^4 = r_3$ for f3, $T_4^5 = r_1 - r_3$ for f1, and $T_4^{(5,7)} = r_3$ for both. Third, if f3 is set up with $r_3 > r_1$, extra time fraction from T' is needed for it, that is, we need additionally assign $T_2^4 = T_4^7 = r_3 - r_1$. Due to $T_2^4 + T_4^7 \leq T'$, we get the maximal achievable r_3 (r_3^{max}) = $r_1 + T'/2 = (1 - 2r_2)/2$. Hence, by considering decoding failure at node 7, the throughput of f3, λ_3 , is given by

$$\lambda_3 = \begin{cases} 0, & 0 < r_3 \leq r_1; \\ r_3 - r_1, & r_1 < r_3 \leq \frac{(1 - 2r_2)}{2}; \end{cases}$$

Under 1st condition $\lambda_3 = 0$ because all packets from f3 are encoded and thus un-decodable at node 7. Under 2nd condition $\lambda_3 = r_3 - r_1$ because a ratio of $(r_3 - r_1)/r_3$ packets from f3 are not encoded at node 4 and thus contribute to λ_3 .

Case 2: $r_1 > r_2$. First, before f3 is set up, the resource allocation for f1 and f2 is as follows. For f1, $T_0^3 = T_4^5 = r_1$, and $T_3^4 = r_1 - r_2$; for f2, $T_1^3 = r_2$; for both f1 and f2, $T_3^{(4,6)} = r_2$. Thus, we get $T' = 1 - 3r_1 - r_2$. Second, similar to **Case 1**, by considering T' for f3, we get $r_3^{max} = r_1 + T'/2 = (1 - r_1 - r_2)/2$. However, the throughput of f3 in this case is different from that in **Case 1**, because a ratio of $(r_1 - r_2)/r_1$ packets from f1, i.e., those packets not encoded at node 3, can be encoded at node 4 with packets from f3. We consider the optimal case that this encodable portion of packets from f1 is encoded prior to the rest portion at node 4. Thus, we get $\lambda_3 = r_3$ when $0 < r_3 \leq r_1 - r_2$. When $r_1 - r_2 < r_3 \leq r_1$, we have $\lambda_3 = r_1 - r_2$, because only the opportunities with the encodable portion of f1 contribute to λ_3 . When $r_1 < r_3 \leq r_3^{max}$, we get $\lambda_3 = r_3 - r_2$ because a ratio of r_2/r_3 packets from f3 are encoded at node 4 and un-decodable at node 7. Thus, λ_3 is given as

$$\lambda_3 = \begin{cases} r_3, & 0 < r_3 \leq r_1 - r_2; \\ r_1 - r_2, & r_1 - r_2 < r_3 \leq r_1; \\ r_3 - r_2, & r_1 < r_3 \leq \frac{1 - r_1 - r_2}{2}; \end{cases}$$

Clearly, by characterizing the region of (r_1, r_2) conditional on $r_3 = 0$, we can get each r_3^{max} at each (r_1, r_2) , and hence characterize λ_3 . The rate region of (r_1, r_2) is given by $\Omega_1 \cup \Omega_2$:

$$\Omega_1 := \{(r_1, r_2) : 2r_1 + 2r_2 \leq 1, 0 \leq r_1 \leq r_2\};$$

$$\Omega_2 := \{(r_1, r_2) : 3r_1 + r_2 \leq 1, r_1 > r_2 \geq 0\}.$$

DCAR-CEO: with the earlier redefined behavior of node 4 by CEO in Section 5.1, we analyze the flow throughput at different achievable (r_1, r_2, r_3) in DCAR-CEO. Since there is no decoding failure here, the throughput of each flow is equal to its flow rate. In what follows, we compute r_3^{max} at a given (r_1, r_2) .

Case 1: $r_1 \leq r_2$. Before f3 is established, the resource allocation for f1 and f2 and the residual time fraction are the same as those in **Case 1** of DCAR. With CEO, there is no coding opportunity at node 4 in this case. Since we can assign $T_2^4 \leq r_1$ without occupying the residual time fraction, r_3^{max} can be obtained as follows. **(a)** If $T' \leq r_1$, then we have $r_3^{max} = T' = 1 - 2r_1 - 2r_2$ by assigning $T_4^7 = T_2^4 = T'$. Hence, we have $\lambda_3 = r_3$, for $0 < r_3 \leq T'$. **(b)** If $T' > r_1$, then we have $r_3^{max} = (r_1 + T')/2 = (1 - r_1 - 2r_2)/2$ by assigning $T_4^7 = T_2^4 = (r_1 + T')/2$. Thus, we have $\lambda_3 = r_3$, for $0 < r_3 \leq (r_1 + T')/2$.

Case 2: $r_1 > r_2$. We also have the same resource allocation and the residual time fraction as those in **Case 2** of DCAR before f3 is set up. With CEO, however, a ratio of $(r_1 - r_2)/r_1$ packets from f1 could be encoded at node 4 in this case. Since we can assign $T_2^4 \leq r_1$ for free, we can have $r_3 \leq r_1 - r_2$ without occupying T' . Then, based on the resource allocation at $r_3 = r_1 - r_2$, we further assign T' to f3 to get r_3^{max} as follows.

(c) If $T' \leq r_2$, we have $r_3^{max} = r_1 - r_2 + T' = 1 - 2r_1 - 2r_2$ by additionally assigning $T_4^7 = T_2^4 = T'$. Thus, we have $\lambda_3 = r_3$, for $0 < r_3 \leq r_1 - r_2 + T'$. (d) If $T' > r_2$, then, we obtain $r_3^{max} = r_1 - r_2 + (T' + r_2)/2 = (1 - r_1 - 2r_2)/2$ by additionally assigning $T_4^7 = T_2^4 = (T' + r_2)/2$. Thus, we have $\lambda_3 = r_3$, for $0 < r_3 \leq r_1 - r_2 + (T' + r_2)/2$.

Similarly, we need to know the rate region of (r_1, r_2) conditional on $r_3 = 0$ to characterize r_3^{max} and λ_3 in DCAR-CEO. Obviously, DCAR and DCAR-CEO have the same region (r_1, r_2) conditional on $r_3 = 0$, which is given by $\Omega_1 \cup \Omega_2$ and shown in Figure 5.3(1).

Comparison: to plot λ_3 in DCAR and DCAR-CEO, we firstly divide the region of (r_1, r_2) into four areas corresponding to the four situations in DCAR-CEO from (a) to (d). Then, four figures, from Figure 5.3(2) to Figure 5.3(5), show λ_3 of the two schemes as the function of r_3 at any given (r_1, r_2) in the four regions, respectively. From the four figures we have two observations. First, if (r_1, r_2) is inside the rate region, DCAR-CEO basically outperforms DCAR, especially achieving a higher maximal λ_3 in all of the four figures. Second, interestingly, referring to Figure 5.3(2) and Figure 5.3(4), when one of the two boundaries ($2r_1 + 2r_2 = 1$ and $3r_1 + r_2 = 1$) in Figure 5.3(1) is reached, which means f1 and f2 have occupied all resource leaving $T' = 0$, both achieve the same maximal λ_3 such that $\lambda_3 = 0$ in Figure 5.3(2) and $\lambda_3 = r_1 - r_2$ in Figure 5.3(4). This indicates that the decoding failure at node 7 will not decrease the maximal throughput of f3 in DCAR in this special situation.

In summary, this analysis shows that CEO is important to XNC or network coding-aware routing in that it ensures the packet decoding, and thus network performance could be improved in most situations.

Figure 5.3: The rate region of (r_1, r_2) and the comparison between DCAR and DCAR-CEO.

5.3 A Distributed Way for Adherence of CEO

To adhere to CEO for the encoding node, it needs to know what packet from one flow was overheard by the other flow. To achieve this, we propose two new mechanisms and integrate them into DCAR to form DCAR-CEO.

5.3.1 Overheard Nodes Recording

This mechanism allows the encoding node(s) of two flows, such as node 3 and node 4 in Figure 5.1, to know which upstream nodes of one flow could be overheard by the downstream nodes of the other flow. This information can be directly obtained during the route discovery process in DCAR.

In DCAR, in order to identify a node as an encoding node between a new

incoming flow and existing flows, each node must execute two tasks. First, each node on an existing flow must record the entire route of its flow. This could be achieved after the flow route is established [83]. Second, in DCAR an on-demand routing strategy like DSR is used by an incoming flow for the route discovery. Based on this routing strategy, upon receiving the route request during route discovery for a new incoming flow, each node in DCAR must include its one-hop neighbors in the route request before propagating it to the downstream nodes. With this complete route and one-hop neighbor information, each node could determine whether it is an encoding node during the route reply (for details, please refer to [83]).

In DCAR-CEO, the coding opportunity identification remains the same as in DCAR, but the identified encoding node is required to record the *upstream overheard nodes* of one flow that can be overheard by its downstream node(s) of another flow. In Figure 5.1, node 3 and node 2, which are the overheard nodes on f1 and on f3, respectively, will be recorded by the intersection node 4 for the flow-pair, f1 and f3.

5.3.2 Packet Encoding/Decoding Recording

In DCAR-CEO, whenever packet encoding or decoding takes place at a node, that node is required to record that event in the packet header, which consists of two types of information. The first is the ID of that node. The second is the packet ID information, which indicates what packets are encoded after encoding or what packets are mixed if it is still an encoded packet after decoding. For example, in Figure 5.1, node 1 will record two packet IDs, P1 and P2, together with node 1's ID in the encoded packet header. This record will be carried along the route by subsequent downstream nodes. Then, in the event of re-encoding or decoding for that packet, these two kinds of information for that

new event will be appended in the packet header. With this record, any node on the flow path will know what packets are encoded together in the encoded packet transmitted by any of its upstream nodes. The overhead of this recording is limited because the size of this information is small and the number of encoding or decoding node is limited because of the small number of hops traversed by each packet.

5.3.3 Re-Encoding Decision at the Encoding Node

With the two mechanisms introduced above in DCAR-CEO, an encoding node will know what packets are transmitted by the overheard node(s) and what packets are overheard by the downstream node(s) of another flow. Hence, with the knowledge of what packets are overheard by the downstream node(s) of another flow, the encoding node can make accurate decisions on which packets to encode and which not according to CEO. For example, when node 4 in Figure 5.1 is about to encode P1 and P3, it will refer to the encoding/decoding record in these two packet headers. From the record of P1, if node 4 finds out that the transmitted packet at node 3 is an encoded packet, $P1 \oplus P2$, which is overheard at node 7, then it will not perform encoding because $P1 \oplus P2$ is different from the packet, P1, to be encoded with P3 at it. Otherwise, node 4 will encode the two packets from f1 and f3. Thus, we see that with the two mechanisms and CEO, all coding opportunities can be correctly identified and thus be effectively utilized.

5.4 Simulation Results for DCAR-CEO

In this section, we evaluate the performance of DCAR and DCAR-CEO by simulation in NS-2. Instead of using scheduling, we adopt IEEE 802.11 standard in the MAC layer in our simulation. The radio transmission range is 100m and the data transmission rate is

1Mbps. User Datagram Protocol (UDP) is used for the traffic source. For simplicity, we assume all flows have the same flow rate.

Figure 5.4 shows the overall throughput comparison between DCAR-CEO and DCAR in the topology of Figure 5.1. First, as expected, DCAR-CEO improves the overall throughput by about 6% as compared to DCAR at high flow rate. Actually, we find that the throughput of f3 is improved significantly (by 30%) as compared to DCAR (not shown as a figure here). Obviously, the packet un-decodability at node 7 leads to the lower performance of DCAR. We find in the simulation that about 50% packets encoded at node 4 in DCAR cannot be decoded at node 7 at high flow rate. Second, the two schemes perform the same at low flow rate whereas DCAR-CEO outperforms DCAR in Figure 5.3(2) and Figure 5.3(3) at $r_1 = r_2 = r_3$ in the analysis. Since our simulation uses IEEE 802.11 which is a random-access MAC, not a scheduling based MAC, few coding opportunities will appear at node 3 between f1 and f2 at low flow rate. Thus, few packets from f1 will be re-encoded at node 4 with packets from f3 resulting in fewer decoding failures for f3 at node 7 in DCAR. Hence, DCAR-CEO performs the same as DCAR at low flow rate.

Figure 5.4: Performance comparison between DCAR and DCAR-CEO in the illustrative topology.

Figure 5.5: Performance comparison between DCAR and DCAR-CEO in the grid topology.

Next, we employ the 4×4 grid topology used in [83] for study. Eight flows will be randomly generated at different time. Figure 5.5 shows the overall throughput of the 8 flows. It can be seen that DCAR-CEO outperforms DCAR. At high flow rate, 40kbps, the performance improvement of DCAR-CEO over DCAR is about 8%. This is because we find in this simulation that DCAR-CEO is able to decode all encoded or re-encoded packets whereas DCAR is not able to do this.

Lastly, we compare DCAR-CEO and DCAR in a $400\text{m} \times 400\text{m}$ network with 30 nodes randomly displaced and eight flows randomly generated. Figure 5.6 shows the performance comparison between them. Again, DCAR-CEO improves the overall throughput of DCAR by about 7% at high flow rate.

Figure 5.6: Performance comparison between DCAR and DCAR-CEO in random networks.

5.5 Description of Problem 2

From this Section, we present the second problem in most previous XNC schemes and propose the new technique of virtual overhearing (VOH) to relax the XNC conditions in previous XNC schemes and hence identify additional network coding opportunities in wireless ad hoc networks.

As introduced earlier, with XNC, a common intermediate node of several flows could mix multiple native packets from these flows into one mixed packet using XOR [80] or random linear combination [119, 120] and then broadcast that mixed packet to serve all these flows. Later, each of those native packets will be decoded from that mixed packet on the way to its destination or at its destination. In fact, to retrieve the native packet from the mixed packet, additional information or packets must be needed; this additional information is referred to as the *side information* [105]. Now, we use a simple XNC example to illustrate what the side information is and how it is acquired for the packet decoding.

Figure 5.7(a) shows this XNC example. Nodes S1 and S2 send packet P1 to D1 and P2 to D2, respectively, and node M is the relay node for both of them. With the traditional store-and-forward routing scheme, four transmission times are needed. With XNC that exploits overhearing, three transmission times are sufficient: node M mixes P1 and P2 into $P1 \oplus P2$ and broadcasts it to nodes D1 and D2; then, node D1 (D2) decodes its desired packet P1 (P2) from the mixed packet $P1 \oplus P2$ and its overheard packet P2 (P1). Note that the packet P1 (P2) used by node D2 (D1) to decode the mixed packet is so-called the *side information*, which is overheard by node D2 (D1) and necessarily needed for the packet decoding.

Figure 5.7: Some typical coding structures in XNC.

From the coding structures surveyed in Chapter 2, the side information can be acquired by a decoding node in XNC in two manners. First, the side information is actually the packet which was previously sent by that decoding node. The reverse carpooling structure and the k -tuple coding structure fall into this category. Second, the side information can be overheard by that decoding node from others. Many coding structures fall into this category, e.g., the two-hop coding structure and the generalized coding structure shown in Figure 5.7(a) and (b), respectively. In fact, with overhearing, more coding opportunities can be discovered. Obviously, when the side information cannot be acquired in these two manners, no coding opportunity will be identified in currently proposed XNC solutions. For example, in Figure 5.7(a), without overhearing at node D_1 (D_2), node M will not be an encoding node, because the encoded packet, $P_1 \oplus P_2$, sent from node M cannot be decoded at node D_1 (D_2). Hence, from this example we can see that although more coding opportunities can be found with overhearing, such a overhearing technique can only overhear the side information from one-hop neighbors of a node, which actually restricts the network coding opportunities to be found. We refer to

this overhearing as *traditional* overhearing in this paper.

Figure 5.8: Network coding with virtual overhearing.

This Chapter explores coding opportunities when the side information cannot be obtained by traditional overhearing. For example, in Figure 5.8, existing XNC solutions could not find any coding opportunity at node 1, i.e., the intersection node of flow 1 (f1) and flow 3 (f3). This is because the downstream node, node 4, of node 1 on f3 cannot overhear any packet sent by f1. By contrast, with our proposed VOH technique, packets from f1 and f3 could still be mixed at node 1. The idea is to allow node 5, as a helping node, on flow 2 (f2) to assist as follows.

- Instead of just sending a packet, P2, of f2 from node 4 to node 0, node 5 encodes P2 with packet P1, which is overheard from node 0 on f1, to form $P1 \oplus P2$.
- Then, node 5 broadcasts $P1 \oplus P2$ to node 0 and node 4.

Consequently, the encoded packet, $P1 \oplus P2$, can be decoded as follows.

- Upon receiving $P1 \oplus P2$, node 0 can still decode P2 from $P1 \oplus P2$ and its own stored native packet P1.
- Similarly, upon overhearing $P1 \oplus P2$, node 4 can decode P1 from $P1 \oplus P2$ and the previously stored packet P2.

In this way, node 4 *virtually* overhears (or seems to be able to overhear) the

packets sent by node 0 of f1. We refer to this technique as *virtual* overhearing, with which one node could virtually overhear other nodes beyond one hop. Thus, node 1 now can become an encoding node for f1 and f3 because one of its downstream nodes on f1 (f3) can overhear the packets from f3 (f1), which guarantees the encoded packet to be decoded at node 3 (node 4). Note that the assistance of node 5 for f1 and f3 does not affect the performance of f2 because P1 is piggybacked onto P2 that f2 needs to send anyway.

Hence, from this example, we can see that (1) VOH could serve as a supplement to traditional overhearing to obtain the side information, and thus (2) additional coding opportunities can be found to boost network throughput. To the best of our knowledge, virtual overhearing has not been proposed previously.

Based on the proposed VOH, our work in the remaining sections in this Chapter is summarized as follows.

- First, we generalize the conditions of VOH from the example in Figure 5.8.
- Second, we illustrate how VOH relaxes network coding conditions to increase coding opportunities. We also present several important applications of VOH.
- Third, we analyze network performance based on the topology in Figure 5.8 with and without VOH, respectively. It shows that the network capacity with VOH could be improved by over 12%, and that in some situation the network throughput can be improved by 50%.
- Fourth, we use NS2 to conduct simulation based on the topology in Figure 5.8 to show the advantage of network coding with VOH.

5.6 Virtual Overhearing and Its Applications

5.6.1 Conditions of Virtual Overhearing

As an example, Figure 5.8 has indicated that with VOH one node can obtain the packets sent by another node that is two or more hops away at no cost. Here, we generalize the conditions under which VOH exists. The conditions are described as follows.

Suppose that there are two flows, $f1$ and $f2$, in the network, and that the path of $f1$ comprises at least three nodes. Without loss of generality, consider *any* three nodes on $f1$, say nodes **a**, **b**, and **c**, with node **a** being upstream to **b** and node **b** being upstream to node **c**. If nodes **b** and **c** can overhear the packets sent by some node(s) on $f2$, then, VOH exists at node **a**, and the node on $f2$ which is directly overheard by node **b** can be virtually overheard by node **a**.

Figure 5.9: Conditions of VOH.

Figure 5.9(a) shows a general example of VOH. In Figure 5.9(a), only two nodes, nodes **b** and **c**, on $f1$ can overhear the packet, P_x , sent by nodes x and y on $f2$, respectively. Thus, according to the conditions above, any of the upstream nodes, node **d**

to node SI , of node b on $f1$ can virtually overhear the packets sent by node x on $f2$. In the following, we use this general example to show the correctness of the conditions above or, equivalently, show how the packet, P_x , sent by node x can reach those nodes, node d to node SI , on $f1$ at no cost.

Table 5-1: Virtual overhearing process.

Round number	Transfer of the overheard packet, P_x , on $f1$				
	Encoding node	Packet from $f1$	Encoded packet	(decoding node, decoded packet)	
1	Node b	$P1$	$P1 \oplus P_x$	(node c , $P1$)	(node d , P_x)
2	Node d	$P2$	$P2 \oplus P_x$	(node b , $P2$)	(node a , P_x)
...	...	...	...	...	...
N	Node n	P_n	$P_n \oplus P_x$	(next hop of node n , P_n)	(previous hop of node n , P_x)

VOH in Figure 5.9(a) works as follows. In the first round, node b encodes a packet, $P1$, from $f1$ with the overheard packet, P_x , from $f2$ to form $P1 \oplus P_x$, and broadcasts it. Then, $P1$ can be retrieved from the encoded packet, $P1 \oplus P_x$, at node c , because node c can overhear P_x from node y . And P_x can be decoded from $P1 \oplus P_x$ and the previously stored packet, $P1$, at node b 's immediate upstream node, node d . In the second round, node d can encode P_x with another packet, $P2$, from $f1$ and broadcasts this encoded packet, $P2 \oplus P_x$, to its upstream node a and downstream node b . Then, using the overheard packet, P_x , node b decodes $P2$ from $P2 \oplus P_x$. The second upstream node, node a , of node b then decodes P_x from $P2 \oplus P_x$ and the previously stored $P2$. In this way, after n rounds, P_x can be propagated to the upstream node n hops away from node b on $f1$. Eventually, P_x could reach the source node, SI . Table 5-1 shows how the packet P_x sent by node x on $f2$ can reach those nodes from d to SI on $f1$. Thus, if a node can

virtually overhear another node, it means it can obtain the packet sent by that another node at no cost.

In fact, VOH widely exists in wireless ad hoc networks. Figure 5.9(b) shows another example of VOH in which two flows intersect at one intermediate node, node M. In this situation, node M's upstream node(s) on f1 (f2), which cannot directly overhear the packet from f2 (f1), can virtually overhear or obtain the packets from f2 (f1). For simplicity, we only look at virtual overhearing at nodes of f1: by treating nodes 1 and M on f1 as nodes *b* and *c* in Figure 5.9(a), respectively, nodes on f1 from node 3 to node S1 can virtually overhear node M on f2. Since two flows intersect with each other very often in wireless networks, especially in multi-hop ad hoc networks, VOH widely exists there.

5.6.2 Relaxed Network Coding Conditions

As seen from those examples in Figure 5.8 and Figure 5.9, VOH, as compared to traditional overhearing, actually explores the broadcast nature of the wireless channels to a further step. That is, with VOH, one node can obtain the packets sent by another node beyond one hop away. Hence, in addition to traditional overhearing, VOH provides a new way for a node to obtain packets from other nodes and thus, as shown in Fig. 2, additional coding structures or coding opportunities could be found. The discovery of these additional coding opportunities indicates that the network coding conditions are relaxed by virtual overhearing. In what follows, we first show another two examples in which coding opportunities are brought only by virtual overhearing, and then give the detailed coding conditions under virtual overhearing.

First, we show an extension of the example in Figure 5.8. Three flows are established and intersected with each other as shown in Figure 5.10(a). At each

intersection node (e.g., node 1), only one flow's downstream node (e.g., node D2 on f2) can directly overhear some upstream node (e.g., node S1) of another flow. Hence, no intersection node here can be identified as an encoding node in traditional network coding schemes [83, 85, 87, 102]. However, with VOH, each intersection node, node 1, node 2 or node 3, in Figure 5.10(a) will be the encoding node for the two flows via it. Similar to node 5 in Figure 5.8, nodes 4, 5, and 6 in Figure 5.10(a) are forced in their transmissions to encode the other flow's packets (including the encoded packets from the other flow) overheard from nodes 1, 2, and 3, respectively. Then, as illustrated in Figure 5.9(b), node 1 on f2, node 2 on f3, and node 3 on f1 can virtually overhear node 3 on f3, node 1 on f1, and node 2 on f2, respectively. Thus, each of the three intersection nodes finally becomes an encoding node.

Figure 5.10: Relaxed coding conditions with virtual overhearing.

Second, we show an example in which coding conditions are further relaxed with

VOH. Four flows are established in the network of Figure 5.10(b). Particularly, f_1 and f_2 intersect at node 2, and at node 2, none of the f_1 's (f_2 's) downstream nodes can directly overhear any of f_2 's (f_1 's) upstream nodes. Thus, node 2 is not an encoding node for f_1 and f_2 in traditional network coding solutions such as COPE in [80] and DCAR in [83]. However, node 2 will be an encoding node if VOH is applied. That is, with the help of f_3 and f_4 , we can impose both node 10 on f_3 and node 11 on f_4 to behave like the helping node, node 5, in Figure 5.8. Then, node 3 on f_1 and node 7 on f_2 can virtually overhear node 6 on f_2 and node 1 on f_1 , respectively. Thus, node 2 becomes an encoding node for f_1 and f_2 .

Therefore, the three examples in Figure 5.8 and Figure 5.10 show that with our proposed VOH, XNC is still possible when side information cannot be obtained from traditional overhearing. We now present the network coding conditions for two flows under virtual overhearing. Here, we consider how the coding conditions in the generalized coding structure [83] are relaxed by VOH. By $Ngb(a)$, $U(a, f)$, and $D(a, f)$, we denote node a 's one-hop neighbors, all node a 's upstream nodes on flow f , and all node a 's downstream nodes on flow f , respectively. In addition, we use $voh(a, b, f)$ to represent that node a can virtually overhear node b on flow f . Suppose two flows, say f_1 and f_2 , intersect at node c , then, for node c to be an encoding node for f_1 and f_2 , we have:

The coding conditions under VOH: (1) there exists $d_1 \in D(c, f_1)$, such that $d_1 \in U(c, f_2)$, $d_1 \in Ngb(s_1)$, or $voh(d_1, s_1, f_2)$, where $s_1 \in U(c, f_2)$, and (2) there exists $d_2 \in D(c, f_2)$, such that $d_2 \in U(c, f_1)$, $d_2 \in Ngb(s_2)$, or $voh(d_2, s_2, f_1)$, where $s_2 \in U(c, f_1)$.

In fact, if we remove $voh(d_1, s_1, f_2)$ in condition (1) and $voh(d_2, s_2, f_1)$ in

condition (2), then, the above conditions become the conditions in the generalized coding structure [83], which is obviously more restrictive.

5.6.3 Application of Virtual Overhearing

Based on the three examples shown in Figure 5.8 and Figure 5.10, we now present several applications of VOH by relating the three examples to some real scenarios.

First, VOH can be applied to wireless mesh or ad hoc networks to increase coding opportunities. Since these networks are generally multi-hop networks, it is most likely that the traffic flows are established similar to those examples shown in both Figure 5.8 and Figure 5.10. Hence, VOH could be effectively used with network coding in wireless mesh or ad hoc networks to boost network performance.

Second, VOH will be very useful in multi-hop cellular networks (MCN) [121]. MCN is the integration of traditional single-hop cellular networks and multi-hop ad hoc networks, and it is considered as a promising candidate of the fourth generation (4G) wireless networks [121, 122]. Figure 5.8 actually represents a typical example of the communications in MCN. Mobile stations, MS 0 and MS 3, send packets to MSs 2 and 4 via the base station, BS 1, respectively. Then MS 4 sends packets to MS 0 via the fixed or mobile relay station 5. Hence, network coding with VOH could be effectively applied to MCN.

Third, VOH will be also useful in traditional single-hop cellular networks. Consider the communications in Fig. 2 again, but firstly direct MS 4's packets to BS 1, not node 5, and then to MS 0. Then, instead of sending the native packet P2 from MS 4, BS 1 broadcasts $P1 \oplus P2$. As a result, MS 4 can virtually overhear MS 0, and hence BS 1 can encode two packets from MSs 0 and 3.

5.7 Performance of XNC with Virtual Overhearing

We have shown in Section 5.6 that VOH could bring additional coding opportunities and thus, we are interested to study how network performance will benefit from these additional opportunities. Here, we analyze the two schemes below to show the performance benefits from the additional coding opportunities in the topology shown in Figure 5.8. The reasons why the analysis targets the topology of Figure 5.8 are explained as follows. On one hand, as mentioned in the applications of virtual overhearing, Figure 5.8 shows a typical network scenario, e.g., the cellular network, which widely exists in our life. Thus, by studying the benefits of virtual overhearing in the topology of Figure 5.8, we will know how useful VOH is to some current deployed networks. On the other hand, the analysis approach employed here, in Figure 5.8, can be extended to other similar scenarios.

Traditional scheme (TS): The traditional scheme here refers to either the traditional routing scheme or the traditional network coding scheme such as COPE and DCAR. We assume in TS that the three flows, f_1 , f_2 and f_3 , are established as shown in Figure 5.8. Then, the traditional network coding scheme actually performs the same as the traditional routing scheme because as mentioned above no coding opportunity can be found at node 1 in Figure 5.8 in the traditional network coding scheme. Hence, each intermediate node under TS only forwards the native packets it received previously.

Network coding with virtual overhearing (NC-VOH): Similarly, the three flows are also established as shown in Figure 5.8. Then, node 5 under NC-VOH encodes one native packet from f_2 with one overheard packet from f_1 in each of its transmissions to node 0. For node 1 under NC-VOH, it can encode two native packets from f_1 and f_3 ,

respectively, and then broadcast the encoded packet to nodes 2 and 4. However, for successful decoding at node 4 in NC-VOH, the network coding at node 1 should be restricted by

$$R_{nc} \leq R_{voh};$$

where R_{nc} denotes the packet encoding rate at node 1 or the encoded packet transmission rate at node 1, and R_{voh} denotes the virtual overhearing rate, i.e., the arrival rate of f1's packets at node 4 through virtual overhearing. The reason for this constraint is that if node 4 receives the encoded packets from node 1 faster than it virtually overhears the packets from f1, then many encoded packets will be un-decodable at node 4. Thus, one *critical rule* in using virtual overhearing for network coding is that *when performing network coding, the encoding node should be aware of the virtual overhearing rate at its decoding node and it should not encode packets faster than the virtual overhearing rate*. In the sub-section below, we assume the restriction above can be satisfied and all encoded packets can be successfully decoded at node 4.

5.7.1 Comparison of Rate Region

To show how NC-VOH outperforms TS based on the topology in Figure 5.8, we characterize and compare the rate regions of the two schemes above. The rate region means all feasible flow rate combinations of the three flows which can be supported by the network. We use (r_1, r_2, r_3) to express the flow rate combination, where r_1 , r_2 , and r_3 represent the flow rates of f1, f2, and f3, respectively.

We employ the resource allocation or the scheduling for this analysis [87]. For interference-free scheduling in Figure 5.8, we assume that the transmission from node 3 to node 1 can be scheduled in parallel with that from node 5 to node 0 due to relatively

large distances from node 5 to node 1 and from node 3 to node 0, and that other transmissions can only be scheduled in sequence. As defined in Section 5.2, T_i^j and $T_i^{(j,k)}$ are used to represent the time fractions assigned to the unicast transmission from node i to node j and the broadcast transmission from node i to nodes j and k , respectively. Note that the broadcast transmission here only refers to the encoded packet broadcast at node 1 in Figure 5.8. Then, by assuming that each link capacity is 1 in Figure 5.8, we have the following two sets of constraints for all feasible (r_1, r_2, r_3) in TS and NC-VOH.

For TS:

$$r_1 \leq T_0^1 \quad (1)$$

$$r_1 \leq T_1^2 \quad (2)$$

$$r_2 \leq T_4^5 \quad (3)$$

$$r_2 \leq T_5^0 \quad (4)$$

$$r_3 \leq T_3^1 \quad (5)$$

$$r_3 \leq T_1^4 \quad (6)$$

$$\max\{T_5^0, T_3^1\} + T_0^1 + T_1^2 + T_4^5 + T_1^4 \leq 1 \quad (7)$$

For Inequality (7), due to the permitted parallel transmission scheduling mentioned earlier, the larger time fraction between T_5^0 and T_3^1 is considered as the time occupied from the system.

For NC-VOH:

$$r_1 \leq T_0^{(1,5)} \quad (8)$$

$$r_1 \leq T_1^2 + T_1^{(2,4)} \quad (9)$$

$$r_2 \leq T_4^5 \quad (10)$$

$$r_2 \leq T_5^{(0,4)} \quad (11)$$

$$r_3 \leq T_3^1 \quad (12)$$

$$r_3 \leq T_1^4 + T_1^{(2,4)} \quad (13)$$

$$T_1^{(2,4)} \leq r_2 \quad (14)$$

$$\max\{T_5^{(0,4)}, T_3^1\} + T_0^{(1,5)} + T_1^2 + T_4^5 + T_1^4 + T_1^{(2,4)} \leq 1 \quad (15)$$

Note that Inequality (14) actually corresponds to the restriction mentioned above in NC-VOH because $T_1^{(2,4)}$ denotes the number of encoded bits received by node 4 per time unit and r_2 , the flow rate of f2, amounts to the number of virtually overheard bits at node 4 per time unit. Hence, with Inequality (14) above, all encoded packets from f3 can be successfully decoded at node 4.

Next, we present how to compute the rate regions of TS and NC-VOH. Two steps are involved for both two schemes. The first is to compute the maximal $r_3(r_3^{max})$ given a feasible fixed (r_1, r_2) . The second step is to characterize all feasible (r_1, r_2) conditional on $r_3 = 0$. Then, we obtain all feasible (r_1, r_2, r_3) , i.e., the rate region.

Step 1 in TS: To support a given (r_1, r_2) with the least resource occupied, the allocation is done as follows: for f1, $T_0^1 = T_1^2 = r_1$; for f2, $T_4^5 = T_5^0 = r_2$. Thus, according to (7), we have $\max\{r_2, T_3^1\} + T_1^4 + 2r_1 + r_2 \leq 1$. It is obvious that when r_3 reaches r_3^{max} , we must have $r_3^{max} = T_3^1 = T_1^4$ and all resources are already allocated, i.e., $\max\{r_2, T_3^1\} + T_1^4 + 2r_1 + r_2 = 1$. Thus, we can obtain r_3^{max} in the following two cases.

Case 1: Suppose $r_3^{max} \leq r_2$, then we have $r_3^{max} + 2r_1 + 2r_2 = 1$, i.e., $r_3^{max} = 1 - 2r_1 - 2r_2$. Thus, when $2r_1 + 3r_2 \geq 1$, we get $r_3^{max} = 1 - 2r_1 - 2r_2$.

Case 2: Suppose $r_3^{max} > r_2$, we have $2r_3^{max} + 2r_1 + r_2 = 1$, i.e., $r_3^{max} = (1 - 2r_1 - r_2)/2$. Thus, when $2r_1 + 3r_2 < 1$, we get $r_3^{max} = (1 - 2r_1 - r_2)/2$.

Combining the two cases above, we have r_3^{max} in TS as follows:

$$r_3^{max} = \begin{cases} 1 - 2r_1 - 2r_2, & 2r_1 + 3r_2 \geq 1 \\ (1 - 2r_1 - r_2)/2, & 2r_1 + 3r_2 < 1 \end{cases} \quad (16)$$

Step 2 in TS: When $r_3 = 0$, we have the following five constraints for (r_1, r_2) :

$r_1 \leq T_0^1, r_1 \leq T_1^2, r_2 \leq T_4^5, r_2 \leq T_5^0$, and $T_0^1 + T_1^2 + T_4^5 + T_5^0 \leq 1$. Thus, from the five constraints, we can compute the rate region of (r_1, r_2) conditional on $r_3 = 0$, which is given by $\Omega = \Omega_1 \cup \Omega_2 \cup \Omega_3$:

$$\Omega_1 := \{(r_1, r_2) | r_1 = 0, r_2 \leq 1/2\};$$

$$\Omega_2 := \{(r_1, r_2) | r_2 = 0, r_1 \leq 1/2\};$$

$$\Omega_3 := \{(r_1, r_2) | 2r_1 + 2r_2 \leq 1\}.$$

Step 1 in NC-VOH: To support a given (r_1, r_2) with the least resource occupied, the allocation is done as follows: for f1, $T_0^{(1,5)} = T_1^2 + T_1^{(2,4)} = r_1$; for f2, $T_4^5 = T_5^{(0,4)} = r_2$. Thus, according to (15), we have $\max\{r_2, T_3^1\} + 2r_1 + r_2 + T_1^4 \leq 1$. Similarly, when r_3^{max} is reached, we have $\max\{r_2, T_3^1\} + 2r_1 + r_2 + T_1^4 = 1$. Combining this equation with Inequalities (12) and (13), we need to set $T_1^{(2,4)}$ as large as possible or to encode packets from f1 and f3 as many as possible to achieve r_3^{max} . Consider the following two cases.

Case 1: If $r_1 \geq r_2$, we can set $T_1^{(2,4)} = r_2$ according to (14). Thus, we get $r_3^{max} = T_1^4 + r_2 = T_3^1 \geq r_2$. Finally, we have $2r_3^{max} + 2r_1 = 1$ or $r_3^{max} = (1 - 2r_1)/2$.

Case 2: If $r_1 < r_2$, we can set $T_1^{(2,4)} = r_1$ according to the least resource allocation for (r_1, r_2) above. Thus, we get $r_3^{max} = T_3^1 = T_1^4 + r_1$ and $\max\{r_2, r_3^{max}\} + r_3^{max} + r_1 + r_2 = 1$. Similar to the computation of r_3^{max} in Step 1 in TS, we consider the following two situations. **(1)** If $r_3^{max} \leq r_2$, we have $r_3^{max} + r_1 + 2r_2 = 1$ or $r_3^{max} = 1 - r_1 - 2r_2$. In other words, when $1 - r_1 - 2r_2 \leq r_2$, i.e., $r_1 + 3r_2 \geq 1$, we get $r_3^{max} = 1 - r_1 - 2r_2$. **(2)** If $r_3^{max} > r_2$, we have $2r_3^{max} + r_1 + r_2 = 1$ or $r_3^{max} = (1 - r_1 - r_2)/2$. In other words, when $r_1 + 3r_2 < 1$, we have $r_3^{max} = (1 - r_1 - r_2)/2$.

Combining Case 1 and the two situations in Case 2, r_3^{max} of NC-VOH is given by

$$r_3^{\max} = \begin{cases} (1 - 2r_1)/2, & r_1 \geq r_2 \\ 1 - r_1 - 2r_2, & r_1 < r_2 \text{ and } r_1 + 3r_2 \geq 1 \\ (1 - r_1 - r_2)/2, & r_1 < r_2 \text{ and } r_1 + 3r_2 < 1 \end{cases} \quad (17)$$

Step 2 in NC-VOH: Obviously, the rate region of (r_1, r_2) conditional on $r_3 = 0$ here is the same as that in TS and is characterized by $\Omega_1 \cup \Omega_2 \cup \Omega_3$.

For a clear view of the analysis results above, we show $r_3^{\max}$ of TS and NC-VOH at each given (r_1, r_2) within Ω in Figure 5.11(b) and Figure 5.11(c), respectively, based on (16) and (17). Figure 5.11(a) shows the region of Ω .

Before we show the performance improvement of NC-VOH over TS, we first verify the correctness of the analysis above. We use MATLAB to solve the linear programs above for TS and NC-VOH to obtain the maximal r_3 at different sampled (r_1, r_2) within Ω , respectively. Figure 5.12 shows the comparison of the maximal r_3 obtained from our analysis and from computer computation for both TS and NC-VOH (blue for TS and red for NC-VOH). It can be seen in the figure that our analysis provides the same maximal r_3 at each sampled (r_1, r_2) as that from the computer computation for both TS and NC-VOH, thus showing the correctness of our analysis of the two schemes.

Figure 5.11: $r_3^{\max}$ at different (r_1, r_2) in TS and NC-VOH.

Figure 5.12: Verification of the correctness of the analysis.

Comparison of r_3^{max} : We compare r_3^{max} between TS and NC-VOH. First, from both Figure 5.11 and Figure 5.12, we can find that r_3^{max} in NC-VOH is always higher than that in TS at any given (r_1, r_2) inside Ω . In particular, at the edge of the rate region Ω (i.e., $2r_1 + 2r_2 = 1$), r_3 is not allowed to be setup in TS due to $r_3^{max} = 0$, whereas it can be established in NC-VOH even with $r_3^{max} = 1/4$ at $(r_1, r_2) = (1/4, 1/4)$. This gain is mainly due to the coding opportunities brought by virtual overhearing. Second, to clearly see how NC-VOH outperforms TS with respect to r_3^{max} , we show the throughput gain of r_3 in Figure 5.13. The gain is defined as the ratio of r_3^{max} in NC-VOH over that in TS at each (r_1, r_2) , and it can be obtained using (16) and (17). From Figure 5.13, it can be seen that the throughput gain of r_3 is limited at low flow rates of f_1 and f_2 , whereas the gain is sharply increasing at high flow rates of f_1 and f_2 , especially near $(r_1, r_2) = (1/4, 1/4)$. This is because at low (high) r_1 and r_2 , few (many) coding opportunities are found at node 1, thus leading to limited (large) performance improvement.

Figure 5.13: Throughput gain of f3 in NC-VOH.

Comparison of network capacity: We first compute the maximal overall throughput, γ , of TS and NC-VOH at each given (r_1, r_2) , which is the sum of r_1 , r_2 , and r_3^{max} .

In TS, using (16) we can get γ as follows.

$$\gamma = \begin{cases} 1 - r_1 - r_2, & 2r_1 + 3r_2 \geq 1 \\ (1 + r_2)/2, & 2r_1 + 3r_2 < 1 \end{cases} \quad (18)$$

Thus, we can calculate the maximal γ (γ^{max}) in (18) over all (r_1, r_2) in Ω as the network capacity of TS. That is, from (18) we have $\gamma^{max} = 2/3$ when $(r_1, r_2) = (0, 1/3)$.

In NC-VOH, using (17) we obtain γ as follows.

$$\gamma = \begin{cases} (1 + 2r_2)/2, & r_1 \geq r_2 \\ 1 - r_2, & r_1 < r_2 \text{ and } r_1 + 3r_2 \geq 1 \\ (1 + r_1 + r_2)/2, & r_1 < r_2 \text{ and } r_1 + 3r_2 < 1 \end{cases} \quad (19)$$

Thus, we can obtain the network capacity of NC-VOH by computing γ^{max} in (19): it can be found that $\gamma^{max} = 3/4$ when $(r_1, r_2) = (1/4, 1/4)$ in NC-VOH. Hence, as compared to TS, NC-VOH can improve the capacity of the network in Figure 5.8 by

12.5%. Note that the capacities of TS and NC-VOH are achieved at different (r_1, r_2) and thus, we are interested to compare γ of the two schemes at the same (r_1, r_2) . We find that the performance improvement of NC-VOH could be much higher at high flow rates of f_1 and f_2 , e.g., γ of NC-VOH and TS at $(r_1, r_2) = (1/4, 1/4)$ is $3/4$ and $1/2$, respectively, thus leading to 50% performance improvement in NC-VOH.

5.7.2 Impact of the Unawareness of $R_{nc} \leq R_{voh}$

In the above analysis of NC-VOH, we show the performance improvement in NC-VOH by assuming the *critical rule* $R_{nc} \leq R_{voh}$ is satisfied or by setting $T_1^{(2,4)} \leq r_2$ as one constraint in NC-VOH. Here, we study the impact of the unawareness of this rule in NC-VOH. Note that this unawareness means the encoding node, i.e., node 1, in Figure 5.8 encodes packets in a greedy way: whenever there are two packets from f_1 and f_3 waiting for transmission at node 1, they will be encoded at node 1 and broadcast by node 1. Thus, to analyze the performance of NC-VOH with the unawareness of this critical rule, we only need to remove the constraint of (14), i.e., $T_1^{(2,4)} \leq r_2$, from the previous set of constraints for NC-VOH. With the new formed set of constraints for NC-VOH, we now present the analysis to show the impact of the unawareness of $R_{nc} \leq R_{voh}$.

First, we calculate the maximal r_3 at each given (r_1, r_2) by solving the new set of constraints. We denote the maximal r_3 here as $r_3'^{max}$. The solution for obtaining $r_3'^{max}$ is similar to the first step used to compute r_3^{max} in NC-VOH above. We omit the details here and give $r_3'^{max}$ as follows.

$$r_3'^{max} = \begin{cases} 1 - 2r_1 - r_2, & r_1 \geq r_2 \text{ and } 3r_1 + r_2 \geq 1 \\ (1 - r_1 - r_2)/2, & r_1 \geq r_2 \text{ and } 3r_1 + r_2 < 1 \\ 1 - r_1 - 2r_2, & r_1 < r_2 \text{ and } r_1 + 3r_2 \geq 1 \\ (1 - r_1 - r_2)/2, & r_1 < r_2 \text{ and } r_1 + 3r_2 < 1 \end{cases} \quad (20)$$

Second, we calculate the maximal effective throughput of f3, $\gamma_3'^{max}$, at each given (r_1, r_2) . With the unawareness of $R_{nc} \leq R_{voh}$, we have mentioned earlier that when $R_{nc} > R_{voh}$, some encoded packets from f3 or a ratio of $(R_{nc} - R_{voh})/r_3$ packets from f3 will be un-decodable at node 4. These un-decodable packets will not contribute to the throughput of f3, causing $\gamma_3'^{max} = r_3'^{max} - (R_{nc} - R_{voh}) < r_3'^{max}$. Thus, to compute $\gamma_3'^{max}$, we need to determine R_{nc} (or $T_1^{(2,4)}$) and R_{voh} (or r_2) when $r_3'^{max}$ at a given (r_1, r_2) is achieved. Due to the greedy encoding at node 1, we have $T_1^{(2,4)} = \min\{r_1, r_3'^{max}\}$, which we find also holds in the first step in solving the set of constraints to obtain $r_3'^{max}$. Hence, with each obtained $r_3'^{max}$ at each (r_1, r_2) within Ω , $T_1^{(2,4)}$ and r_2 are already determined, and thus $\gamma_3'^{max}$ can be readily calculated, which is given by

$$\gamma_3'^{max} = \begin{cases} r_2, & r_1 \geq r_2 \text{ and } 3r_1 + r_2 \geq 1 \\ (1 - 3r_1 + r_2)/2, & r_1 \geq r_2 \text{ and } 3r_1 + r_2 < 1 \\ 1 - r_1 - 2r_2, & r_1 < r_2 \text{ and } r_1 + 3r_2 \geq 1 \\ (1 - r_1 - r_2)/2, & r_1 < r_2 \text{ and } r_1 + 3r_2 < 1 \end{cases} \quad (21)$$

Figure 5.14: Impact of the unawareness of the critical rule.

To show how the unawareness of the *critical rule* impacts network performance in NC-VOH, we compare $\gamma_3'^{max}$ in (21) with r_3^{max} in (17), which is the maximal

throughput of f3 at each (r_1, r_2) under the awareness of the critical rule. Figure 5.14 plots the gain or the ratio of $\gamma_3'^{max}$ over r_3^{max} at different (r_1, r_2) within Ω . Two observations can be made here. First, it can be seen that when $r_1 < r_2$ we have $\gamma_3'^{max} = r_3^{max}$, which indicates the unawareness of the critical rule has no impact on network performance. This is because due to $T_1^{(2,4)} = \min\{r_1, r_3'^{max}\} \leq r_1$ and $r_1 < r_2$, $R_{nc} \leq R_{voh}$ always holds in this situation and thus, all encoded packets can be decoded at node 4. Second, we can find $\gamma_3'^{max} \leq r_3^{max}$ always holds when $r_1 \geq r_2$. Especially, at large r_1 and small r_2 , $\gamma_3'^{max}$ is much lower than r_3^{max} . This is because the overhearing rate, R_{voh} , is very low at small r_2 and thus, it causes many encoded packets from f3 un-decodable at node 4. Hence, this analysis shows that being aware of the *critical rule*, i.e., $R_{nc} \leq R_{voh}$ in performing virtual-overhearing-supported network coding is necessary for performance improvement.

5.7.3 Discussion

Although the analysis of NC-VOH above is based on the topology in Figure 5.8, it can be similarly extended to other more complicated scenarios such as those shown in Figure 5.10(a) and (b). Here, we particularly discuss the performance analysis of NC-VOH in a general network where flows are randomly established. In this general case, the formulation of NC-VOH actually becomes challenging. The reason for this difficulty can be explained as follows. In NC-VOH, whether a node is an encoding node or not depends on how the paths of the flows are established, and thus cannot be determined locally for this node under the random network setting. Without knowing which flows' packets can be encoded, the node is not likely to consider network coding-based resource allocation to serve any two flows simultaneously, which makes the formulation of NC-VOH

challenging in the random network. In fact, this is also why most analysis work on XNC [81, 87-89] is based on the two-hop coding structure [80] where a node can locally determine which flows' packets can be encoded. However, as an optional choice, we may choose to design a distributed algorithm of NC-VOH to study its performance in the random network, which is actually presented in Chapter 6.

5.8 Simulation Results for NC-VOH

In the previous Section, we have shown the improved performance of NC-VOH using ideal interference-free scheduling. Here, we still attempt to show the benefits of NC-VOH under a practical and commonly used random-access MAC, i.e., IEEE 802.11 standard, in different networks. Specifically, we will use NS-2 to conduct the simulation for NC-VOH and TS to show (1) the additional coding opportunities brought by VOH can improve network performance; and (2) the unawareness of the virtual overhearing rate or the greediness in packet encoding at the encoding node such as node 1 in Figure 5.8 could detriment network performance.

To show the above two objectives, we design our simulation as follows. In all simulations we force the encoding nodes to encode packets in a greedy way. But we will properly adjust the flow rates of some network flows such as f2 in Figure 5.8 to make the *critical rule* being followed or violated so as to show objective (1) or (2). Other simulation parameters like the data transmission rate keep the same as in the study for DCAR-CEO in Section 5.4.

5.8.1 Performance Study of NC-VOH in Figure 5.8

In this study, we will show both objectives, (1) and (2) above. We design the simulation

as follows: the flow rates of f1 and f3 in Figure 5.8 are fixed at a high value, 160kbps, to enable rich coding opportunities at node 1, but the flow rate of f2 will be set from a low value, 40kbps, to a high value, 120kbps. Obviously, the critical rule will be followed most likely at the high flow rate of f2, and will be violated at the low flow rate of f2.

Figure 5.15: Flow throughput comparison in network of Figure 5.8.

After a number of simulations, the average flow throughputs with 90% confidence interval for NC-VOH and TS are shown in Figure 5.15. First, for f2, as explained earlier, VOH has no impact on it: the throughput of f2 in NC-VOH basically keeps the same as in TS at all flow rates of f2. Second, for f3, we see that it benefits from network coding in NC-VOH at high flow rate of f2, but gets hurt at low flow rate of f2. This is because the high (low) flow rate of f2 enables a high (low) VOH rate at node 4 on f3, which guarantees nearly 100% (only 62%) encoded packets of f3 to be decoded as shown in Table 5-2. One may notice from Table 5-2 that for f3 to decode most encoded packets, the flow rate of f2 can be much lower (> 80 kbps) than 160kbps. The reason is that due to the random-access MAC, no more than 50% of the transmitted packets at

node 1 are encoded packets in the simulation, yielding $R_{nc} < 80\text{kbps}$ on average. Third, for f1, however, we see that it always benefits from network coding in NC-VOH regardless of the flow rate of f2, e.g., its throughput at high flow rate of f2 is about 34% higher than that in TS. This is because f1 relies on traditional overhearing for packet decoding, which is more reliable.

Other performance parameters of NC-VOH and TS are shown in Table 5-2. At high flow rate of f2 ($\geq 100\text{kbps}$), the overall network throughput of NC-VOH is about 22% higher than that of NC-VOH. Meanwhile, NC-VOH achieves shorter packet delivery delay than TS due to the higher achieved throughputs of f1 and f3 in NC-VOH.

Table 5-2: Performance parameters in NC-VOH and TS

Flow rate of f2 (kbps)	Overall throughput (kbps)		Average delay (s)		Decoded packet ratio
	NC-VOH	TS	NC-VOH	TS	NC-VOH
40	342.7	334.8	0.17	1.05	62.3%
60	351.9	327.3	0.83	1.53	70.8%
80	371.1	318.1	1.17	1.57	91.5%
100	379.8	313.2	1.18	1.60	99.5%
120	374.9	310.1	1.79	2.17	99.6%

5.8.2 Performance Study of NC-VOH in Figure 5.10(b)

Next, we study NC-VOH and TS in the network within the circle in Figure 5.10(b) to show only the objective (1) above. The flow rates of f3 and f4 are set equally at a relatively high value, 70kbps, and the flow rates of f1 and f2 will vary simultaneously from 100kbps to 140kbps. Figure 5.16 shows the achieved throughputs of f1 and f2 in NC-VOH and TS. Again, we can see in Figure 5.16 that at high flow rate, $\geq 130\text{kbps}$, due to rich coding opportunities at the encoding node, node 2, of f1 and f2, the throughputs of

f1 and f2 in NC-VOH are both improved by about 17%, as compared to TS. We notice that most encoded packets of both f1 and f2 in NC-VOH are successfully decoded at their decoding nodes, which ensures the higher performance of NC-VOH.

Figure 5.16: Flow throughput comparison in network of Figure 5.10(b).

5.9 Conclusion

This Chapter studies the XNC conditions in most previous XNC schemes and addresses two problems in those schemes. The following summarizes our work in this Chapter.

As the first part in this Chapter, we present and solve the coding opportunity misidentification in most previous XNC schemes. We propose CEO principle to guarantee successful decoding of all encoded or re-encoded packets in XNC or network coding-aware routing in wireless networks as long as overhearing is reliable. We also propose two new mechanisms to facilitate the adherence of CEO for the encoding node. Specifically, with the two mechanisms, an encoding node is able to correctly identify each coding opportunity. By both analysis in a simple topology and simulations in various scenarios, we show that network performance can be improved by CEO.

As the second part in this Chapter, we propose a new technique called virtual

overhearing (VOH) to explore network coding opportunities in wireless networks when the side information for the decoding of the network-coded packets cannot be obtained from traditional overhearing. With VOH, a node is able to obtain the packets at no cost from another node beyond one hop. Thus, it is more likely for this node to decode encoded packets. This is why VOH can be used to increase coding opportunities. In the meantime, by generalizing the conditions of VOH, we point out that it widely exists among flows, especially among those intersecting flows, in wireless networks.

Importantly, we have demonstrated the advantage of network coding with VOH in improving network performance by both analysis and simulation in some representative topologies. In particular, we point out one critical rule in using VOH for network coding. That is, to benefit network performance, the packet encoding rate, R_{nc} , at the encoding node should not be faster than the virtual overhearing rate, R_{voh} , at the decoding node. Otherwise, encoding packets from different flows are instead detrimental to network performance. Both our analysis and simulation for NC-VOH have shown the importance of this rule.

Lastly, it can be expected that with either this re-encoding problem being solved or the XNC conditions being relaxed, XNC can be better applied to G-ER or LBGR for performance improvement. We will discuss how to apply CEO and VOH to G-ER or LBGR in the last Chapter of this Thesis.

Chapter 6

Network Coding-Aware Routing with Virtual Overhearing

The previous Chapter has shown that VOH can be used to increase network coding opportunities and attain network performance gain. However, the attained performance gain is only shown in two simple network topologies. Hence, in this Chapter we attempt to show the advantages of network coding with VOH in other various network topologies. Towards this goal, we first design some network coding-aware routing scheme(s) based on VOH in a general wireless ad hoc network; and then, we conduct extensive simulations in various situations to evaluate the benefits of VOH. The rest of this Chapter is organized as follows. In Section 6.1, we present two ways to utilized VOH for increasing coding opportunities, which is critical to the design of the network coding-aware routing. Section 6.2 introduces four definitions closely related to VOH. Then, our proposed two routing schemes are illustrated in Sections 6.3 and 6.4, respectively. Section 6.5 gives the complexity analysis of DCAR-VOH+. The simulation results are presented in Section 6.6, and Section 6.7 concludes this Chapter.

6.1 Introduction

We have mentioned in Chapter 2 that network coding-aware routing can be employed to

find more coding opportunities and thus benefit network performance more. And we have also mentioned in Chapter 5 that VOH can be used by the decoding node to acquire the side information and additional coding opportunities can be found. Hence, to take advantage of both to benefit network performance, we propose some network coding-aware routing scheme(s) based on VOH in wireless ad hoc networks.

VOH could be used for increasing coding opportunities in at least two ways. Let us refer to Figure 6.1 which shows a more general example of XNC that exploits VOH. In this example, existing XNC solutions or coding-aware routing protocols could not find any coding opportunity at node 1, the intersection node of flow 1 (f1) and flow 3 (f3). This is because none of the downstream nodes of f3, node 5 or 6, can overhear packets of f1. However, with VOH or with the help of flow 2 (f2), node 6 is able to obtain the packets sent by node 0 of f1, details of which can be found from Table 5-1 in the previous Chapter. Then, the two ways are presented as follows.

- (1) Among the three flows, if f3 is the last flow to enter the network, it could be made aware of the virtual overhearing at node 6 and thus take it into consideration in the route discovery process. In other words, new incoming flows are more likely to find coding opportunities thanks to VOH.
- (2) If f2 is the last flow to be established, it could *create* a coding opportunity between f1 and f3 enabled by VOH. All previously proposed coding-aware routing protocols [81, 83, 87, 99, 101, 102] only attempt to find coding opportunities between a new flow and existing flows. This is because with only traditional overhearing, if two existing flows are not mixable, they will never be mixable regardless of the other flows. In contrast, with VOH, new flows can create coding opportunities for existing flows.

Figure 6.1: A more complicated example of network coding with VOH.

To apply VOH, we propose two routing protocols in this Chapter: distributed coding-aware routing with virtual overhearing (DCAR-VOH) and its enhanced version DCAR-VOH+. DCAR-VOH implements the first manner of usage whereas DCAR-VOH+ implements both manners of usage. Our extensive simulations indicate that VOH provides an effective way to discover additional coding opportunities in various network topologies, resulting in improved network performance. The positive effect of the second feature stands out especially.

6.2 Virtual Overhearing and Its Discovery

In this Section, we introduce four definitions related to virtual overhearing: virtual overhearing node, virtual neighbor, virtual encoding node, and virtual decoding node. Then, we describe how to discover a virtual neighbor for a node.

6.2.1 New Definitions in Virtual Overhearing

Suppose that there are two flows, f_1 and f_2 , in the network, and that the path of f_1 comprises at least three nodes. Without loss of generality, consider *any* three nodes on f_1 , say nodes **a**, **b**, and **c**, with node **a** being upstream to node **b** and node **b** being upstream to node c . If nodes **b** and **c** can overhear the packets sent by some node on f_2 , then 1)

node **b** is a *virtual encoding* node, and node **c** is a *virtual decoding* node to node **b**; 2) node **a** is a *virtual overhearing* node on f1; and 3) the node on f2 overheard by node **b** is a *virtual neighbor* to node **a**. Referring to the general network topology in Figure 6.2 (which is already shown in Chapter 5), nodes **b** and **c** on f1 can overhear the packets sent by nodes **x** and **y** on f2, respectively. Therefore, node **c** is a virtual decoding node to node **b**, which is a virtual encoding node on f1. Meanwhile, all upstream nodes of node **b** on f1, from node **d** to **S1**, are virtual overhearing nodes, and node **x** on f2 is a virtual neighbor to these virtual overhearing nodes on f1. Similarly, we can find in Figure 6.1 that node 8 on f2 is a virtual encoding node and node 0 on f2 is a virtual decoding node to node 8, and that node 0 on f1 is a virtual neighbor to node 6 on f2.

Figure 6.2: Four definitions in a general wireless network.

From the above definition, we here show three notable features in VOH. First, normally, a virtual neighbor is at least two hops away from a virtual overhearing node, e.g., node **x** in Figure 6.2 is two hops from node **d**. Second, a virtual encoding node may have more than one virtual decoding node, e.g., if a node on f1 downstream to node **c** in Figure 6.2 can also overhear some node on f2, it can become a virtual decoding node to the virtual encoding node, node **b**, on f1. Lastly, the virtual-neighbor relationship may not be symmetric. As shown in Figure 6.2, if node **x** is a virtual neighbor of node **d**, node

d is not necessarily a virtual neighbor of node **x**.

Lastly, we emphasize that (1) the packets sent by a virtual neighbor node could be virtually overheard by the virtual overhearing node at no cost, the process of which is actually already shown in Table 5-1 in Chapter 5; and (2) the packet encoded by a virtual encoding node can be decoded at any virtual decoding node of that virtual encoding node.

6.2.2 Virtual Neighbor Discovery

In this sub-section, we propose a *distributed* mechanism for a node on a flow to discover all potential virtual neighbors. In order to do that, two requirements are imposed.

First, every node on an existing flow records the entire path consisting of the node IDs of all of the nodes on that flow and the flow ID of that flow. Note that the flow ID is uniquely represented by the structure of (*source ID, flow sequence number*), where the flow sequence number is generally assigned in an increasing order to the subsequent flows generated from a source. The entire path can be directly obtained because source routing is normally adopted by the coding-aware routing [83, 99, 101, 102]. Second, once a new flow is established, every node except the destination node on that flow broadcasts the new flow ID to its neighbors. We employ Hello message broadcast to carry this new flow ID information. Upon receiving this broadcast, each neighbor not on the same flow creates a local record in the form of (*neighbor_flow_id, hello_source*) where *neighbor_flow_id* is the new flow ID in the Hello message and *hello_source* is the node sending this Hello message. This record is called the *overhearing record* (OHR).

The virtual neighbor discovery procedure involves two steps:

Step 1: Reporting OHR to the upstream nodes – This step is triggered by one of the following two situations:

- (1) A node on some existing flow(s) creating a new OHR – If a node on some existing flow(s) creates a new OHR, then, for each existing flow, it generates a packet containing the new OHR and the existing flow ID, *my_flow_id*, and forwards that packet to all of its upstream nodes on that existing flow. Upon receiving this packet, each upstream node extracts and stores the information locally in the form of (*neighbor_flow_id*, *my_flow_id*, *hello_source*, *downstream_node*) where the *downstream_node* is the node sending this packet. We refer to this as the *reported overhearing record* (ROHR). Note that the step in this situation will allow a node on some existing flow(s) to discover some virtual neighbors from a newly established flow.
- (2) A node with some previously stored OHRs appearing on a newly established flow – If a node on a newly established flow has previously stored OHRs, it creates a packet to contain all of those OHRs with its new flow ID and forwards this packet to its upstream nodes along the new flow. Upon receiving that packet, each upstream node creates an ROHR for each OHR contained in the packet. The step in this situation will allow a node on a newly established flow to find some virtual neighbors from existing flows.

Step 2: Extracting virtual neighbors from ROHRs – After Step 1, a node may have stored some ROHRs. Then, it tries to find some virtual neighbors from those ROHRs. Among those ROHRs, if there are two records with the same first two elements, i.e., *neighbor_flow_id* and *my_flow_id*, a new virtual neighbor is found according to the definition in Section 6.2. The newly found virtual neighbor is locally stored as the virtual neighbor record (VNR) in the form of (*neighbor_flow_id*, *my_flow_id*, *VN_id*) where the *VN_id* is the virtual neighbor's ID, i.e., the *hello_source* stored in one of the two ROHRs.

Figure 6.3: Virtual neighbor discovery.

Figure 6.3 shows an example of the virtual neighbor discovery process in the first situation. Suppose $f2$ is a newly established flow. Then, after nodes **b** and **c** on $f1$ receive Hello messages from nodes **x** and **y** on $f2$, respectively, each of them creates a new OHR locally and forwards that OHR to the upstream nodes on $f1$. After node **a** on $f1$ receives two OHRs from its downstream nodes, it creates two ROHRs, $(f2, f1, x, b)$ and $(f2, f1, y, c)$. With these two ROHRs, node **a** knows that node **x** on $f2$ is a new virtual neighbor according to the definition in Section 6.2.1 and records it as a new VNR in the form of $(f2, f1, x)$ locally.

6.3 DCAR-VOH

In this Section, we describe how DCAR-VOH makes use of VOH to discover the coding opportunity during the setup of a new incoming flow. The route discovery, the route selection, and the virtual overhearing initiation are three main steps in DCAR-VOH.

6.3.1 Route Discovery

When a source intends to set up a flow to a destination, it needs to collect all possible paths to the destination and select the best one among them. We here employ the on-

demand routing like DSR for the route discovery for an incoming flow.

First, to collect those paths, the source broadcasts a RREQ packet. The RREQ packet consists of the source ID, the destination ID, and a flow sequence number.

Next, upon receiving an RREQ, if a node is not the destination and the sequence number in that RREQ is larger than the one it previously received from this source, the node then updates its local record of the sequence number for this source and further broadcasts this RREQ. Before broadcasting, the node appends its ID in the RREQ packet so as to eventually form the entire path. It also appends all of its 1-hop neighbors and VNRs in this packet. The 1-hop neighbor is directly obtained from the Hello message and the VNRs are obtained from the mechanism in Section 6.2.

Once this RREQ eventually reaches its destination, the destination then sends a RREP packet back to the source along the opposite path recorded in the RREQ packet header.

During the forwarding of the RREP packet back to the source, each intermediate node needs to execute the following two tasks.

- (1) Coding opportunity identification. If this intermediate node is on some existing flows, it needs to identify whether it could be an encoding node between the new incoming flow and any of those existing flows. Here we treat the virtual neighbor in the same way as the 1-hop neighbor. Thus, we adopt the same two conditions as in DCAR as mentioned in Chapter 2 to identify the coding opportunity between two flows. However, a higher priority is given to *direct* overhearing than to virtual overhearing. This means that if the network coding between two flows can be simultaneously supported by direct overhearing and virtual overhearing, direct overhearing

will be used. This is because direct overhearing is more reliable than virtual overhearing, which will be shown in the simulation.

- (2) Channel congestion state (CCS) report. As introduced in Chapter 2, for an optimal flow path, it should be less congested. Thus, we also consider the congestion state of each path in DCAR-VOH. That is, each intermediate node in DCAR-VOH is required to report its CCS [83] in the RREP so that the source can evaluate the returned routes and choose the best route. We adopt the same metric, the modified interface queue length (MIQ), defined in DCAR to reflect the CCS. The MIQ length at one node is the required number of transmissions for that node to send all packets in the interface queue, and this can be calculated based on the coding graph [83]. This coding graph is built based on the coding opportunities identified from (1) for two flows. Details of the calculation of the MIQ length can be found in [83]. Then, CCS of one node is simply the sum of the MIQ length at this node and all of its neighbors. In order to know the up-to-date MIQ length values of all neighbors, every node is required to contain its current calculated MIQ length in the Hello message.

6.3.2 Route Selection

Once all RREP packets are returned, the source first evaluates each route by summing up the recorded MIQ length values of all intermediate nodes in the reply. Then, it chooses the route with the smallest sum for the new incoming flow. During the packet transmission process, this route path will be appended in the packet header.

6.3.3 Virtual Overhearing Initiation

Once an intermediate node knows that it is an encoding node for a new flow and the network coding is supported by virtual overhearing, the intermediate node needs to initiate the virtual overhearing. This initiation will be executed once this encoding node receives the first packet from the source of this new flow. For example, in Figure 6.1, once node 1 receives a packet from f3 for the first time, it will inform node 8 on f2 to start mixing the overheard packet from f1 and the packet from f2 so that the decoding node of f3, node 6, can get the packet from f1 later.

Two steps are needed in the virtual overhearing initiation as shown in Figure 6.4.

Step 1: *Encoding node informing the decoding node about the virtual neighbor record.* In the previous coding opportunity identification, the encoding node came to know the decoding node of the new flow and its virtual neighbor. Now, the encoding node starts sending a packet carrying this VNR to this decoding node. In Figure 6.1, for example, node 1 knows that node 6 is the decoding node of f3 and node 0 is its virtual neighbor from the VNR $(f1, f2, 0)$ contained in the RREP. Then, node 1 sends this VNR to node 6 once it receives the first packet from f3.

Step 2: *Start of virtual overhearing.* Upon receiving the VNR, $(neighbor_flow_id, my_flow_id, VN_id)$, in Step 1, the decoding node then needs to inform the corresponding virtual encoding node on flow, my_flow_id , to start mixing the overheard packet from flow, $neighbor_flow_id$. The process works as follows. The decoding node sends a packet containing the information of $neighbor_flow_id$ and VN_id to its first downstream node of flow, my_flow_id . Then, this packet will be continuously forwarded to the virtual encoding node which can virtually overhear the node, VN_id , on the flow $neighbor_flow_id$. Then, this virtual encoding node will start mixing the

overheard packets from flow *neighbor_flow_id*, and the mixed packets will be eventually decoded at any virtual decoding node of this virtual encoding node. All nodes on flow *my_flow_id* between the virtual encoding node and the decoding node of the new flow will further mix the overheard packets so that those packets can finally reach the decoding node, the main process of which is depicted in Table 5-1 in Chapter 5.

Figure 6.4: Virtual overhearing initiation in DCAR-VOH.

Note that the virtual encoding node only selects for mixing those packets that are overheard after it receives the virtual overhearing ‘start’ command to mix. All of those packets previously overheard by the virtual encoding node are ignored.

6.4 DCAR-VOH+

In this Section, we introduce the enhanced scheme, DCAR-VOH+, which is designed on top of DCAR-VOH and allows a new incoming flow to create coding opportunities for two previously unmixable existing flows. The coding opportunity creation for two unmixable existing flows in DCAR-VOH+ is presented below.

Figure 6.5: Coding opportunity creation in a more general case.

In Section 6.1, we introduced a new concept of *creating* coding opportunities for two unmixable existing flows by utilizing virtual overhearing. Here, we present a more general case for this new concept. The network in this case is already shown in Figure 5.10(b), but for the convenience of illustration, we repeat to show it in Figure 6.5. In this figure, if direct overhearing between two flows, f1 and f2, is not possible, the intersection node, node 2, cannot become an encoding node for f1 and f2 in either DCAR or DCAR-VOH. However, as illustrated in Figure 6.5, thanks to virtual overhearing, node 2 could still be the encoding node for f1 and f2 after the establishment of f3 and f4. Hence, in order to make use of this property of virtual overhearing, DCAR-VOH+ is proposed.

DCAR-VOH+ is based on DCAR-VOH, but incorporates the following two new mechanisms.

- (1) *Flow overhearing status recording at the non-encoding intersection node.* For each non-encoding intersection node of two flows, say f1 and f2, if all downstream node(s) of f1 (f2) can neither directly nor virtually overhear the upstream node(s) of f2 (f1), the intersection node is required to record this situation. This kind of information is referred to as the flow overhearing status, and it can be obtained during the route discovery for a new flow in DCAR-

VOH when the coding opportunity identification is executed by the intersection node. In Figure 6.1, for example, in the absence of f_2 , the intersection node of f_1 and f_3 , i.e., node 1, will make a record that none of the downstream nodes of f_3 could directly or virtually overhear the upstream nodes of f_1 .

(2) *New VNR reporting.* As mentioned earlier, with the virtual neighbor discovery mechanism, each node may create some new VNRs after a new incoming flow is established. A new VNR creation means that the abovementioned flow overhearing status might be changed. In DCAR-VOH+, once a node on some existing flow(s) creates a new VNR, it sends a packet to report that VNR to the upstream nodes of each existing flow. This packet carries the existing flow ID, this node ID, and the new VNR, (*neighbor_flow_id*, *my_flow_id*, *VN_id*). Refer to Figure 6.1 again. Once node 0 is detected by node 6 as a new virtual neighbor after the establishment of f_2 , node 6 then reports this new VNR (f_1 , f_2 , 0) to all of its upstream nodes on f_3 , including node 1.

Upon receiving the reported VNR from (2), a node knows that some downstream node of the existing flow indicated in the packet can virtually overhear some upstream node of flow *neighbor_flow_id* in VNR. Hence, with the acquisition of this information, a non-encoding intersection node of two flows may update its recorded flow overhearing status between the two flows. Then, this node can determine whether new coding opportunities appear between the two flows. For example, in Figure 6.1, once node 1 receives the newly reported VNR (f_1 , f_2 , 0) from node 6, node 1 knows that some downstream node of f_3 now can virtually overhear node 0 on f_1 , i.e., one of its upstream

nodes on f1, and that the flow overhearing status between f1 and f3 can be updated. Thus, node 1 finally knows that it is now an encoding node of f1 and f3, and that it needs to initiate the virtual overhearing process as introduced earlier.

With the two features of virtual overhearing implemented in DCAR-VOH+, we can see that DCAR-VOH+, as compared to DCAR and DCAR-VOH, can maximally relax the network coding conditions and discover the most coding opportunities. Thus, we will see from the simulation that DCAR-VOH+ can benefit network performance most in many different scenarios. However, it may suffer from a dependence on virtual overhearing in some situations, as will be also observed from the simulation below.

We now present the system implementation of DCAR-VOH+ in NS-2. Figure 6.6 shows how DCAR-VOH+ is implemented at each node. Four modules are closely related to DCAR-VOH+: the routing module, the hello broadcast module, the decoding module, and the interface queue module.

Figure 6.6: DCAR-VOH+ architecture.

First, it can be seen from Figure 6.6 that the routing module is to manage all

flows via one node, mainly responsible for the packet forwarding, the flow route discovery, the coding opportunity identification, and the overhearing status recording between any two established flows. Second, the hello broadcast module is to manage the neighbor information, including the 1-hop neighbor and the virtual neighbor. Third, two queues reside in the decoding module: the overhearing queue and the decoding waiting queue. The former is used to store both directly overheard packets and virtually overheard packets. The latter is uniquely used in DCAR-VOH+ for storing those encoded packets, of which the decoding is supported by VOH, because it is possible in DCAR-VOH+ for the encoded packet to arrive at its decoding node earlier than the packet to be virtually overheard for decoding. For example, in Figure 6.6, if node 8 is congested, packet P1, which needs to be virtually overheard by node 6, may arrive at node 6 later than the encoded packet $P1 \oplus P2$. Hence, with this decoding waiting queue, the encoded packets could be decoded later. Lastly, the interface queue module is used to store those packets waiting for a free channel. Each time the interface queue module is about to send a data packet to the MAC layer, it refers to the coding graph maintained by the routing module to see whether any other queued packet can be mixed with this data packet.

6.5 Complexity of DCAR-VOH+

We now analyze the communication complexity of DCAR-VOH+ in terms of the number of control messages. By comparing to DCAR, we claim that DCAR-VOH+ has the same communication complexity. The reasons are given as follows.

First, the virtual neighbor discovery in DCAR-VOH+ can be performed without any additional packet to be generated. On one hand, the Hello message is used in both DCAR and DCAR-VOH+, and the broadcast of a new flow ID in DCAR-VOH+ is

carried out in the Hello message transmission. On the other hand, a new created OHR can be piggybacked onto the data packet to be overheard by the upstream nodes of a flow. Referring to the example in Figure 6.3 again, when node **b** sends f_1 's packets, it can piggyback the new created OHR onto f_1 's packets, and let its upstream node on f_1 overhear the OHR. Later, node **b**'s upstream node(s) which receives the new OHR can further do the same to let all node **b**'s upstream nodes on f_1 be informed of this OHR. Second, the communication complexity of the route discovery for a flow in DCAR-VOH+ is actually the same as that in DCAR. Both DCAR-VOH+ and DCAR adopt the same route request and route reply as DSR, which leads to a communication complexity of $O(2N)$ [128], where N denotes the number of nodes in the network. Third, the virtual overhearing initiation in DCAR-VOH+ also does not require any new control packet to be generated. Referring to Figure 6.4 again, the information carried in the two steps can actually be piggybacked onto the data packets. For example, the information of VNR in Step 1 can be piggybacked onto f_3 's packets in Figure 6.4. Therefore, the above three reasons show that DCAR-VOH+ has the same communication complexity as DCAR in terms of the number of the control packet.

6.6 Simulation Results and Discussion

In this Section, we evaluate the performance of the two proposed routing protocols, DCAR-VOH and DCAR-VOH+, in different network scenarios. The two well-known network coding schemes, COPE and DCAR, are used for comparison. The main performance metrics are the flow throughput and network throughput.

The simulation is conducted in NS-2. IEEE 802.11 standard is adopted as the MAC layer, and the data transmission rate is fixed at 1Mbps. The radio transmission

range is 100 meters, and the carrier sensing range is 250 meters. The lengths of the interface queue, the overhearing queue, and the decoding waiting queue are set at 50 packets, 500 packets, and 1500 packets, respectively. The Hello broadcast interval is 1 second. UDP is used as the transport layer protocol for each traffic source.

6.6.1 Performance Evaluation of DCAR-VOH

6.6.1.1 DCAR-VOH in the Illustrative Topology

First, we evaluate how network coding with VOH can improve network performance by referring to the scenario defined in Figure 6.1. Two case studies are considered.

Case 1: Fixed flow rate of f2. The first incoming flow is f1, and the last incoming flow is f3. We vary the flow rate of f1 and f3 simultaneously from 40kbps to 130kbps, and fix the flow rate of f2 at a relatively high level, 80kbps.

Figure 6.7 shows the throughput of each flow in DCAR-VOH and DCAR. First, it can be seen that at a high flow rate of 100kbps to 130kbps, the throughput of f1 and f3 in DCAR-VOH is about 18% and 15% higher than that in DCAR, respectively. Supported by VOH, network coding at node 1 between f1 and f3 improves the throughput of both f1 and f3 in DCAR-VOH. However, DCAR cannot find any coding opportunity, resulting in a lower throughput. At a low flow rate (< 80 kbps), DCAR and DCAR-VOH perform at the same level for the throughput of f1 and f3 because both can handle the light traffic. Second, as expected, the throughput of f2 in DCAR-VOH remains essentially the same as in DCAR, which shows that virtual overhearing does not affect the throughput of f2. Consequently, as shown in Figure 6.8, DCAR-VOH improves the overall throughput of DCAR at a high flow rate by about 12%.

Figure 6.7: Flow throughput with a fixed rate of f2.

Figure 6.9 shows the comparison of the average packet delivery delay in DCAR-VOH and DCAR. It can be observed that at a high flow rate ($> 80\text{kbps}$), the average delay in DCAR-VOH is approximately 0.5 seconds shorter than that in DCAR because of the network coding at node 1 between f1 and f3 to allow node 1 to send two packets in one transmission.

Figure 6.8: Overall throughput with a fixed rate of f2.

Case 2: Different flow rates of f2. Case 2 is used to study how the flow rate of f2 affects the performance of f1 and f3, mainly f3. The flow rates of f1 and f3 are fixed at a

saturated rate of 110kbps and the flow rate of f2 varies from 20kbps to 100kbps. Figure 6.10 shows the throughput of these three flows under DCAR-VOH and DCAR. First, it can be seen that regardless of the flow rate of f2, the throughput of f1 in DCAR-VOH is higher than that in DCAR by about 16%. This is because with the network coding at node 1 in DCAR-VOH, whenever the channel is occupied by node 1, the packets from f1 can be transmitted immediately, whereas in DCAR those packets from f1 cannot be sent immediately because they need to compete with packets from f3 at node 1. Second, VOH similarly has no impact on the throughput of f2.

Figure 6.9: Average packet delivery delay.

Figure 6.10: Flow throughput at different flow rates of f2.

Figure 6.11: Decoded packet ratio of f3 at different flow rates of f2.

We now compare the throughputs of f3 in DCAR and DCAR-VOH. Interestingly, it can be seen that (1) at a low flow rate of f2 the throughput of f3 in DCAR is higher than that in DCAR-VOH; (2) but at a high flow rate of f2 the situation is just the opposite. At a low flow rate of f2, the competition to use the channel between f2 and f3 is small at node 6. Therefore, the throughput of f3 in DCAR is high due to low channel competition between f2 and f3 at node 6. However, in DCAR-VOH, since most packets from f3 received at node 6 are encoded packets from node 1, they need the packets from f1 to be virtually overheard for their decoding at node 6. But, as shown in Figure 6.11, at a low flow rate of f2 many encoded packets cannot be decoded at node 6 because fewer packets from f1 are virtually overheard by node 6 due to the lower flow rate of f2. Thus, the decoding failure decreases the throughput of f3 in DCAR-VOH, and again, it shows the importance to follow the *critical condition* presented in Chapter 5 when using VOH for network coding. At a high flow rate of f2, as shown in Figure 6.11, node 6 can decode most of the encoded packets sent by f3 because node 6 can virtually overhear the packets from f1 quickly. Hence, at a high flow rate of f2, the throughput of f3 in DCAR-VOH will benefit from the coding opportunities at node 1. By contrast, as the flow rate of f2

increases, the f3's throughput in DCAR decreases because there is more competition for the use of the channel between f2 and f3 at node 6. As a result, the throughput of f3 in DCAR-VOH is higher than that in DCAR at a high flow rate of f2.

Hereafter, for all simulations below, we assume an equal flow rate for all flows to avoid the detrimental effect of low virtual overhearing rate at the decoding node to evaluate the performances of DCAR-VOH and DCAR-VOH+.

6.6.1.2 DCAR-VOH under Inter-cluster Traffic in a Random Network

In this study, we evaluate the performance of DCAR-VOH in a 400m x 400m random network topology with inter-cluster traffic. Four cluster-regions are formed at four corners of the network area, as shown in Figure 6.12. We assume that each cluster-region has a relatively small region, a 150m x 150m square, to represent some traffic hotspot, and that any two cluster-regions are beyond one hop. A total of 30 nodes are randomly distributed in the network and the nodes located in one cluster-region form a cluster. Then, six inter-cluster flows are generated in the network at different times. Here, the inter-cluster flow means that the source and the destination of the flow are randomly selected from two cluster-regions. Both DCAR and COPE are simulated for comparison.

Figure 6.13 compares the overall throughput of the six flows in these three protocols. As expected, DCAR outperforms COPE by 3% and DCAR-VOH outperforms DCAR by 5%. Obviously, the reason for this is that VOH enables DCAR-VOH to identify more coding opportunities than DCAR in the inter-cluster traffic pattern. To explain how DCAR-VOH can find more coding opportunities, we refer to one representative example of the inter-cluster traffic pattern. Suppose that three inter-cluster flows are generated in three different cluster-regions, as shown in Figure 6.12. Two

features exist in this traffic pattern. First, for two nodes within one cluster-region on two flows, the possibility that one can directly overhear the other is high due to the small area. Second, for two nodes in different cluster-regions on two flows, the possibility that one can directly overhear the other becomes low. By referring to the network coding conditions in DCAR mentioned in Chapter 2, it is difficult in this scenario to meet these conditions due to this second feature, as shown in Figure 6.12. By contrast, with VOH existing between any two intersecting flows in Figure 6.12, it is highly possible that DCAR-VOH can find coding opportunities under this inter-cluster traffic pattern. For example, referring to Figure 6.12, if f_1 and f_2 are the first two established flows, then, f_3 , during its establishment, will find coding opportunities at node 3 with f_1 . Therefore, the inter-cluster traffic favors DCAR-VOH in identifying coding opportunities, and thus DCAR-VOH outperforms DCAR.

Figure 6.12: Simulation topology in the random network with clusters.

Figure 6.13: Overall throughput in the random network with clusters.

6.6.2 Performance Evaluation of DCAR-VOH+

6.6.2.1 DCAR-VOH+ in an Illustrative Topology

In this simulation, we study how DCAR-VOH+ can further improve DCAR-VOH based on the scenario defined in Figure 6.12 where three flows, f1, f2, and f3, are established. The three flows are randomly generated at different times with the same flow rate.

Figure 6.14: Overall throughput of the three flows in the topology of Figure 6.12.

Figure 6.14 shows the overall throughput of DCAR-VOH+, DCAR-VOH, and DCAR. Not surprisingly, DCAR-VOH+ is the best performer among them. At a high

flow rate (i.e., > 80 kbps), DCAR-VOH+ could improve the overall throughput over DCAR and DCAR-VOH by 14% and 9%, respectively. The main reason for this is that all three intersection nodes in the figure, i.e., nodes 1, 2, and 3, are identified as the encoding nodes in DCAR-VOH+, whereas only ONE node (node 1, 2 or 3, depending on the establishment order of the three flows) can be the encoding node in DCAR-VOH and no encoding node is identified in DCAR. Hence, DCAR-VOH+ benefits from the higher number of coding opportunities that are identified.

Figure 6.15: The average transmission times for each delivered packet.

In fact, VOH increases coding opportunities without incurring additional costs, such as energy consumption. Hence, similar to traditional network coding, the improved performance in DCAR-VOH+ is achieved even at less cost. Figure 6.15 reflects the energy consumption of these three protocols in terms of the average needed physical transmission times for each successfully delivered packet. The average number, N_t , of transmission times is defined as the ratio of the total number of data packet transmissions over the total number of received data packets at the destinations. From Figure 6.15, it can be seen that the average number of transmission times for DCAR, DCAR-VOH, and DCAR-VOH+ is about 4.3, 4.1, and 3.8, respectively. With three encoding nodes, 1, 2,

and 3, DCAR-VOH+ can achieve average transmission times of less than 4, which is the same as the number of hops for each flow in Figure 6.12. The reason why DCAR requires more than 4 transmission times is that no coding opportunity is identified there, and there are some wasted transmissions because some packets are dropped at some intermediate nodes due to an overflow at the interface queue.

6.6.2.2 DCAR-VOH+ under Random Traffic in a Random Network

We here study the performance of DCAR-VOH+ under random traffic in a random topology. Thirty nodes are randomly placed in a 400m x 400m network, and six flows are randomly generated at different times. All flows are injected at an equal flow rate.

Figure 6.16: Overall throughput in the random network.

The average overall throughput of DCAR-VOH+, DCAR-VOH, and DCAR is shown in Figure 6.16. Still, DCAR-VOH+ outperforms the other two schemes and it improves the performance of DCAR by about 7%. Due to more relaxed coding conditions in DCAR-VOH+, DCAR-VOH+ can find the most coding opportunities, thus improving network performance. By contrast, due to much less relaxed coding conditions in DCAR-VOH, DCAR-VOH performs at essentially the same level as DCAR in a random network with random traffic flows.

6.6.2.3 DCAR-VOH+ under Inter-cluster Traffic in a Random Network

Apart from the simulation in the random topology above, we are interested in knowing how DCAR-VOH+ will perform under inter-cluster traffic in a random network. We keep the same configuration as that introduced above. The comparison of the performance of these three protocols is shown in Figure 6.13. It can be seen that DCAR-VOH can improve the performance of DCAR by about 5% and that DCAR-VOH+ can improve the performance over DCAR by 10%. Together with the simulation in the random topology above, it can be seen that DCAR-VOH+ can make better use of virtual overhearing to benefit network performance in different network conditions.

6.7 Conclusion

In this Chapter, we present two ways of utilizing VOH for increasing network coding opportunities and design two network coding-aware routing protocols. The first routing protocol, DCAR-VOH, uses VOH to identify coding opportunities for a new incoming flow. The second protocol, DCAR-VOH+, makes further use of VOH by *creating* coding opportunities for other flows. It allows a new incoming flow to make two previously unmixable existing flows mixable. As a result, there are many more coding opportunities identified in DCAR-VOH+ as compared to DCAR-VOH.

Simulation results show that under many different network scenarios both DCAR-VOH and DCAR-VOH+ can discover or create coding opportunities in addition to those in the two well-known network coding schemes, DCAR and COPE. This results in improved network performance. In particular, the network with inter-cluster traffic favors both DCAR-VOH+ and DCAR-VOH, and with more relaxed network coding conditions, DCAR-VOH+ outperforms DCAR-VOH, especially in random networks

with random traffic.

Nonetheless, it is critical and challenging to ensure timely packet decoding with virtual overhearing in both DCAR-VOH and DCAR-VOH+. Specifically, in the simulations we find that there are two main problems when using virtual overhearing.

(1) *Low rate of virtual overhearing.* We find that if a node virtually overhears packets for packet decoding slowly, this is indeed detrimental to network performance. As observed in the simulation, one possible reason for this is a low rate of the flow, e.g., f2 in Figure 6.1, on which VOH is performed. In fact, we find that some other factors could also lower the rate of virtual overhearing. In Figure 6.1, for example, if a new flow is generated at node 8 and this causes node 8 to become congested, then the rate of virtual overhearing at node 6 will decrease sharply. Without being aware of the low rate of virtual overhearing, simply applying network coding at some node will present a risk.

(2) *Useless virtually overheard packet at the decoding node.* We also find that in both DCAR-VOH and DCAR-VOH+ there are always a number of *useless* packets virtually overheard at the decoding node such as node 6 in Figure 6.1. This is because it is possible that the virtually overheard packet may occasionally not be encoded at the encoding node such as node 1 in Figure 6.1. For example, P1 in Figure 6.1 may occasionally not be encoded with any packet from f3 at node 1, but it is virtually overheard by node 6. Then, virtually overhearing those *useless* packets will decrease the rate of virtual overhearing, and thus degrade network performance. Hence, a more intelligent packet selection in virtual overhearing may be needed.

The two problems above need to be solved to benefit G-ER or LBGR more from VOH, and we will discuss two optional solutions to these two problems in the next Chapter.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

This Thesis studies the routing strategies in MANETs under group mobility. We first adopt the store-carry-forward approach used in the delay/disruption-tolerant network (DTN) to achieve the communications in MANETs in case that any two groups are isolated for long time. By making full use of the group mobility, we propose three effective routing schemes, group-epidemic routing (G-ER), leader based group routing (LBGR), and purposeful movement assisted routing (PMAR). In addition, this Thesis also studies the inter-flow network coding (XNC) in wireless ad hoc networks, with the focus on the study of the conditions of XNC. The work that has been done in this Thesis can be summarized as follows.

In order to have a better understanding of the MANET routing, we first conduct a brief review on the basics of MANETs, including the mobility models, the traditional MANET routing, and the DTN routing. In particular, we focus on presenting some features of group mobility and illustrating various proposed DTN routing strategies. From this review, we find that the design of both the traditional MANET routing and the DTN routing largely depends on the node movement patterns in the network, i.e., the mobility models, and that each proposed MANET routing scheme in the literature only performs well in some certain scenarios. All these in the review are essential to the

design of the routing in MANETs under group mobility.

The first routing scheme aiming for group mobility is called G-ER. G-ER is a two-layer routing protocol, including the intra-group routing and the inter-group routing. The table-driven routing is employed in the former, and the principle of one inter-group packet copy for each group is adopted in the latter. Particularly, the principle adopted in the inter-group routing greatly lowers the resource requirement for the packet delivery in G-ER. Moreover, the buffer sharing within a group and the group based packet exchange are further integrated into G-ER, which finally make G-ER perform effectively in different scenarios. The simulation results show that G-ER can guarantee the packet delivery with limited node buffer space, but it will be impacted by group dynamics.

To overcome the weakness of G-ER under group dynamics, LBGR is proposed. LBGR does not require the group members to store the inter-group packets and thus, the leave of the members will not incur packet unloading and packet dropping. Meanwhile, LBGR allocates most of the group buffer resources to the leader in each group and thus, the leave of the members will not impact the “carry” capacity of that group. By inheriting most strategies in G-ER and incorporating the above two features in the design, LBGR, as compared to G-ER, is much less impacted by group dynamics.

Subsequently, we consider the node mobility control in group mobility to increase contact opportunities among groups and expedite the packet delivery. Specifically, we propose two simple but effective mobility control techniques; the group formation enforcement and the group purposeful movement. In the former technique, some members in one group are positioned strategically to extend the reachability of that group; and in the latter technique, some members are commanded by its leader to move purposely towards some other group to create a new connection. Our stochastic analysis

shows that both mobility control techniques can effectively reduce the packet delivery delay of G-ER. In particular, we design a routing scheme, PMAR, based on the second technique, and add it onto G-ER. The simulation results show that PMAR improves G-ER effectively in the scenario where node buffer space is small and the node speed is low.

The second part of this Thesis studies XNC conditions in wireless ad hoc network, aiming to effectively apply XNC in MANETs under group mobility. To have a better understanding of XNC, we also conduct a detailed survey on XNC in wireless ad hoc networks. Various situations in which XNC can be applied to an intermediate node of multiple flows are enumerated in the survey. Moreover, various design factors or strategies, e.g., the scheduling and the network coding-aware routing, are introduced in the survey to benefit network performance more from XNC. In addition, some drawbacks of applying XNC are also presented in the survey.

Towards our aim mentioned above, we first solve the re-encoding problem in most of the current proposed XNC-based routing schemes. The re-encoding problem refers to the misidentification problem of the network coding opportunity. That is, if a packet is re-encoded or encoded for twice or more in most existing XNC-based routing schemes, then it might be un-decodable at the decoding node, which is detrimental to network performance. To solve this problem, we propose a principle called consistency of encoding and overhearing (CEO), and we also propose a distributed mechanism for the encoding node to adhere to CEO. Both our analysis and simulation in some topologies show the importance of CEO to XNC or to network coding-aware routing.

Then, we propose a new technique called virtual overhearing (VOH) to discover additional coding opportunities. With VOH, a node can obtain the packet sent by another node which can be two or more hops away at no cost; whereas with traditional

overhearing, a node can only obtain the packet sent by its one-hop neighbors. Thus, with VOH, a node can acquire packets from many other nodes. As a result, VOH can be used to increase network coding opportunities. Our analysis in a simple but typical network topology shows that the additional coding opportunities brought by VOH can effectively be translated into performance gain.

Lastly, we present two ways to utilize VOH. First, it increases the chance of mixing a new incoming flow with existing flows. Second, and more importantly, it allows a new incoming flow to *create* coding opportunities between two existing flows that were previously unmixable. Then, based on these two features, we design two routing protocols: distributed coding-aware routing with VOH (DCAR-VOH), and its enhanced version DCAR-VOH+. DCAR-VOH is designed based only on the first feature, whereas DCAR-VOH+ incorporates both features in its design. The simulation results show that both DCAR-VOH and DCAR-VOH+ can improve network performance because of the additional coding opportunities found by them. In particular, since DCAR-VOH+ implements the two features above, the XNC conditions are much more relaxed in DCAR-VOH+. Thus, DCAR-VOH+ finds the most coding opportunities and performs the best in many situations.

7.2 Recommendations for Further Work

7.2.1 Problem of Low Virtual Overhearing Rate

In Chapter 5, we have shown by analysis and simulation in two simple networks that network coding opportunities can be increased with VOH to improve network performance. Moreover, in Chapter 6 we design two network coding-aware routing

schemes, DCAR-VOH and DCAR-VOH+, to make effective use of VOH, and the simulation results show that the two routing schemes can find additional coding opportunities to benefit network performance in various network topologies. Nonetheless, the studies in these two Chapters also show that the above performance gain diminishes as the virtual overhearing rate decreases, because many encoded packets cannot be decoded at low virtual overhearing rate.

To solve this problem, we may have the following two options.

(1) *Letting the encoding node in DCAR-VOH (or DCAR-VOH+) encode packets adaptively.* At low virtual overhearing rate, the encoded packet is received by the decoding node in DCAR-VOH faster than the packet to be virtually overheard, which causes the decoding failure of the encoded packet. Thus, based on a low virtual overhearing rate, it is possible to ensure the packet decoding by letting the encoding node like node 1 in Figure 6.1 encode packets adaptively. Specifically, the encoding node can increase the encoding rate if it is informed by a decoding node that only few undecoded packets are accumulated at it; otherwise, it can decrease the encoding rate. This simple mechanism can be incorporated into DCAR-VOH to effectively ensure the packet decoding.

(2) *Letting a virtual encoding node additionally forward some overheard packets to the decoding node to ensure the decoding of the encoded packets.* As shown in Chapter 6, the low virtual overhearing rate is possibly caused by the low rate of the flow on which the virtual encoding node lies. To increase the probability of decoding the encoded packet in this situation, one aggressive way is to let the virtual encoding node additionally forward the overheard packet to the decoding node. For example, if the flow rate of f2 in Figure 6.1 is low, the decoding node of f3, node 6, will virtually overhear the

packets from f1 slowly, and many coded packets of f3 at node 6 will fail to be decoded. However, node 8, the virtual encoding node for node 6, can be requested to additionally forward f1's packets to node 6, besides mixing f1's packets with f2's packets. Obviously, this is achieved at the expense of *additional* traffic injected to the network, but this tentative solution can work effectively in the situation where the *additional* traffic does not compete with existing flows for the use of network resources such as the bandwidth. For example, we have shown in Figure 5.8 that NC-VOH can support (r_1, r_2, r_3) up to $(\frac{1}{4}, \frac{1}{4}, \frac{1}{4})$. Thus, in the case of $(r_1, r_2, r_3) = (\frac{1}{4}, r_2, \frac{1}{4})$ with $r_2 < \frac{1}{4}$, we can use the residual bandwidth, which is $\frac{1}{4} - r_2$, for node 5 to additionally send f1's packets to node 4 to ensure the packet decoding.

Obviously, option (1) is conservative whereas option (2) is aggressive, because option (2) could bring more benefits in some situations. However, more investigations for option (2) are needed, e.g., on how to know the residual network resources. We leave them as our future work.

7.2.2 Extension of XNC to MANETs

Our studies on XNC in this Thesis target static wireless ad hoc networks. But this Thesis considers the routing strategy in MANETs under group mobility. Hence, how to extend the proposed XNC solutions to our concerned scenario or how to add XNC onto G-ER (or LBGR) is briefly discussed here.

As mentioned earlier, XNC is applicable to the intra-group routing in G-ER, because nodes in a group stay close to each other and thus the group is a connected sub-network. To apply XNC to G-ER, we can add following modules to G-ER; the virtual neighbor maintenance, the coding opportunity detection, and the packet encoding

deactivation. First, due to node mobility in MANETs, the one-hop neighbors of a node and the flow path may vary from time to time. Hence, the virtual neighbors of a node may also vary from time to time, because the existence of the virtual neighbor relies on the existing flows. To make use of VOH in XNC, the virtual neighbors of a node need to be maintained in a dynamic manner. Second, similarly, due to the changes of both the network topology and the flow path in MANETs, a certain coding structure may also appear or disappear from time to time. Hence, a common intermediate node of two or more flows needs to dynamically detect the appearance of the coding opportunity or some certain coding structures. Upon discovering the coding opportunity, the intermediate node or the encoding node can adhere to CEO to make a decision on the packet encoding. Third, due to the node mobility, a formed coding structure may disappear after some time, which means the coding opportunity disappears accordingly from then on. Hence, once the disappearance of the coding opportunity is detected, the encoding node needs to deactivate the packet encoding at it.

With these three modules, the packet encoding can be done in G-ER in a dynamic manner and furthermore, our proposed XNC solutions can also be applied to connected MANETs under either entity mobility or group mobility. However, we can see that the execution of these three modules will incur some amount of overhead. Thus, how to control this overhead will be the focus of applying XNC to G-ER. We leave this as our future work.

7.2.3 Further Improvement of G-ER

Here, we point out one direction to further improve G-ER: network coding, both XNC and the intra-flow network coding (INC), can be applied to the inter-group routing

of G-ER for performance improvement. Many previous studies [124-127] show that network performance of a DTN based on entity mobility can benefit greatly from both XNC and INC when the bandwidth or the node buffer space is limited. More importantly, they further show that these benefits can be achieved by simply encoding packets from the same flow or different flows into a random linear combination in each transmission of a node. For example, if node A stores a number of native packets (say $P_1, \dots, P_N$) possibly originated from different nodes but destined to the same destination node, and meets an intermediate node B, it forwards to node B some random linear combinations of these packets (each combination expressed as $\sum_i c_i P_i$ where c_i is the random coefficient) instead of some native packets. This can finally boost network performance. Obviously, by treating each group as a single unit, the linear combinations of packets can be exchanged between groups in G-ER. However, there exists some challenge(s). For example, if the abovementioned packets $P_1, \dots, P_N$ are stored at multiple nodes in a group, how can they form a linear combination to be forwarded to another group to benefit network performance? Some cooperative mechanism inside a group may be needed. We leave the work of applying network coding to the inter-group routing of G-ER as the future work.

Author's Publications

Journal Papers:

- [1] **Ling Fu Xie**, Peter Han Joo Chong, and Yong Liang Guan, "Performance analysis of network coding with virtual overhearing in wireless networks," *IEEE Transactions on Vehicular Technology*, accepted for publication, 2014
- [2] **L. F. Xie**, P. H. J. Chong, S. C. Liew, and Y. L. Guan, "CEO: consistency of encoding and overhearing in network coding-aware routing," *IEEE Wireless Communications Letters*, vol. 2, no. 2, pp. 187-190, 2013.
- [3] **Ling Fu Xie**, Peter Han Joo Chong and Yong Liang Guan, "Routing strategy in disconnected mobile ad hoc networks with group mobility," *EURASIP Journal on Wireless Communications and Networking*, vol. 2013, no. 1, 2013.
- [4] **Ling Fu Xie**, Peter H. J. Chong and Yong Liang Guan, "Leader based group routing in disconnected mobile ad hoc networks with group mobility," *Wireless Personal Communications*, vol. 71, no. 3, pp. 2003-2021, 2013.
- [5] Jane Yang Yu, **Ling Fu Xie**, Mingyang Zhang, and Peter H. J. Chong, "A performance comparison of flat and cluster based routing in mobile ad hoc networks," *International Journal of Wireless Information Networks*, vol. 19, no. 2, pp. 122-137, 2012.

Conference Papers:

- [1] **Ling Fu Xie**, H. J. Chong, Y. L. Guan, and B. C. Ng, "Routing for Group Mobility in Mobile Ad Hoc Networks with Delay-Tolerant Approach", in *Proc. of IFIP WMNC*, Hungary, 2010.

- [2] **Ling Fu Xie**, H. J. Chong, Y. L. Guan, and B. C. Ng, "G-ER: Group-Epidemic Routing for Mobile Ad Hoc Networks with Buffer Sharing Mechanism", in *Proc. of IEEE IWCMC*, Istanbul, 2011.

- [3] **Ling Fu Xie**, Peter H. J. Chong, and Yong Liang Guan, "Purpose-Movement Assisted Routing for Group Mobility in Disconnected Mobile Ad Hoc Networks," in *Proc. of IEEE WCNC*, Paris, France, Apr 2012.

Bibliography

- [1] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," *Wireless Communications and Mobile Computing*, vol. 2, pp. 483-502, 2002.
- [2] K. Blakely and B. Lowekamp, "A structured group mobility model for the simulation of mobile ad hoc networks," in *Proc. of the ACM MobiWac*, pp. 111-118, Philadelphia, Pennsylvania, USA, 2004.
- [3] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. Peh, and D. Rubensteinl, "Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with ZebraNet," in *Proc. of 10th International Conference on Architectural Support for Programming Languages and Operating Systems*, pp. 96-107, San Jose, CA, United states, 2002.
- [4] T. Small and Z. J. Haas, "The shared wireless infostation model - A new ad hoc networking paradigm (or where there is a whale, there is a way)," in *Proc. of the ACM MOBIHOC*, pp. 233-244, Annapolis, MD, United states, 2003.
- [5] K. Fall, "A delay-tolerant network architecture for challenged internets," in *Proc. of ACM SIGCOMM*, pp. 27-34, Karlsruhe, Germany, 2003.
- [6] 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, pp. 215-233, 2003.
- [7] A. Vahdat and D. Becker, "Epidemic routing for partially-connected ad hoc networks," in *TR CS-2000-06*, Duke University, 2000.
- [8] M. Zhang and P. Chong, "Performance comparison of flat and cluster-based hierarchical ad hoc routing with entity and group mobility," in *Proc. of IEEE Wireless Communications and Networking Conference*, pp. 1-6, Budapest, Hungary, 2009.
- [9] X. Hong, K. Xu, and M. Gerla, "Scalable routing protocols for mobile ad hoc networks," *IEEE Network*, vol. 16, pp. 11-21, 2002.
- [10] R. R. Ahlswede, N. Cai, S. Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, pp. 1204-1216, 2000.
- [11] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and wait: An efficient routing scheme for intermittently connected mobile networks," in *Proc. of ACM SIGCOMM Workshops: Conference on Computer Communications*, pp. 252-259, Philadelphia, PA, United states, 2005.
- [12] Q. Li and D. Rus, "Communication in disconnected ad hoc networks using message relay," *Journal of Parallel and Distributed Computing*, vol. 63, pp. 75-86, 2003.
- [13] W. Zhao, M. Ammar, and E. Zegura, "A message ferrying approach for data delivery in sparse mobile Ad Hoc Networks," in *Proc. of the Fifth ACM MoBiHoc*, pp. 187-198, Tokyo, Japan, 2004.
- [14] L. F. Xie, P. H. J. Chong, and Y. L. Guan, "Routing strategy in disconnected mobile ad hoc networks with group mobility," *EURASIP Journal on Wireless Communications and Networking*, in Press, 2013.

- [15] L. Pelusi, A. Passarella, and M. Conti, "Opportunistic networking: data forwarding in disconnected mobile ad hoc networks," *IEEE Communications Magazine*, vol. 44, pp. 134-141, 2006.
- [16] Z. Zhang, "Routing in intermittently connected mobile ad hoc networks and delay tolerant networks: overview and challenges," *IEEE Communications Surveys & Tutorials*, vol. 8, pp. 24-37, 2006.
- [17] L. F. Xie, P. H. J. Chong, and Y. L. Guan, "Leader based group routing in disconnected mobile ad hoc networks with group mobility," *Wireless Personal Communications*, in Press, 2012.
- [18] L. F. Xie, P. H. J. Chong, S. C. Liew, and Y. L. Guan, "CEO: consistency of encoding and overhearing in network coding-aware routing," *IEEE Wireless Communications Letters*, in Press, 2012.
- [19] N. Aschenbruck, E. Gerhards-Padilla, and P. Martini, "A survey on mobility models for performance analysis in tactical mobile networks," *Journal of Telecommunication and Information Technology*, pp. 54-61, 2008.
- [20] X. Hong, M. Gerla, G. Pei, and C. Chiang, "A group mobility model for ad hoc wireless networks," in *Proc. of the ACM International Workshop on MSWiM*, pp. 53-60, Seattle, WA, USA, 1999.
- [21] B. Zhou, K. Xu, and M. Gerla, "Group and swarm mobility models for ad hoc network scenarios using virtual tracks," in *Proc. of IEEE Military Communications*, pp. 289-294, Monterey, CA, USA, 2004.
- [22] M. Musolesi and C. Mascolo, "A community based mobility model for ad hoc network research," in *Proc. of 2nd International Workshop on Multi-hop Ad Hoc Networks: from Theory to Reality*, pp. 31-38, Florence, Italy, 2006.
- [23] S. A. Williams and D. Huang, "A group force mobility model," in *proc. of 9th Communications and Networking Simulation Symposium*, pp. 333-340, Huntsville, AL, USA, 2006.
- [24] D. Kim and S. Hwang, "Swarm group mobility model for ad hoc wireless networks," *Journal of Ubiquitous Convergence Technology*, vol. 1, pp. 200-206, 2007.
- [25] G. Pei, M. Gerla, and X. Hong, "LANMAR: landmark routing for large scale wireless ad hoc networks with group mobility," in *Proc. of Mobile Ad Hoc Networking and Computing*, pp. 11-18, Boston, MA, USA, 2000.
- [26] C. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance vector routing (DSDV) for mobile computers," in *Proc. of the ACM SIGCOMM*, pp. 234-244, London, UK, 1994.
- [27] G. Pei, M. Gerla, and T. Chen, "Fisheye state routing: a routing scheme for ad hoc wireless networks," in *Proc. of IEEE ICC*, pp. 70-74, New Orleans, LA, USA, 2000.
- [28] P. Jacquet, P. Muhlethaler, and A. Qayyum, "Optimized link state routing protocol for ad hoc networks," in *Proc. of IEEE INMIC*, pp. 62-68, Lahore, Pakistan, 2001.
- [29] C. E. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing," in *Proc. of 2nd IEEE Workshop on Mobile Computing Systems and Applications*, pp. 90-100, Los Alamitos, CA, USA, 1999.
- [30] D. B. Johnson, D. A. Maltz, and Y. C. Hu, "The dynamic source routing for mobile ad hoc networks," *Internet-Draft*, 2004.

- [31] B. Karp and H. Kung, "GPSR: Greedy Perimeter Stateless Routing for wireless networks," in *Proc. of MOBICOM*, pp. 243-254, Boston, MA, USA, 2000.
- [32] F. Granelli, G. Boato, D. Kliazovich, and G. Vernazza, "Enhanced GPSR routing in multi-hop vehicular communications through movement awareness," *IEEE Communications Letters*, vol. 11, pp. 781-783, 10/ 2007.
- [33] J. Tao, G. Bai, H. Shen, and L. Cao, "ECBRP: An efficient cluster-based routing protocol for real-time multimedia streaming in MANETs," *Wireless Personal Communications*, vol. 61, pp. 283-302, 2011.
- [34] Y. Lee, K. Xu, X. Hong, and M. Gerla, "Hybrid landmark routing in ad hoc networks with heterogeneous group mobility," in *2nd International Symposium on Wireless Communications Systems*, pp. 125-129, Siena, Italy, 2005.
- [35] B. Zhou, Y. Z. Lee, M. Gerla, and F. D. Rango, "Geo-LANMAR: a scalable routing protocol for ad hoc networks with group motion," *Wireless Communications and Mobile Computing*, vol. 6, pp. 989-1002, 2006.
- [36] L. L. Dai and V. W. S. Chan, "Proactive topology reinforcement of wireless networks," in *Proc. of 8th IEEE Wireless Communications and Networking*, pp. 1-6, Kowloon, China, 2007.
- [37] L. L. Dai and V. W. S. Chan, "Helper node trajectory control for connection assurance in proactive mobile wireless networks," in *Proc. of 16th International Conference on Computer Communications and Networks*, pp. 882-887, Honolulu, HI, USA, 2007.
- [38] Y. Mei, C. Xian, S. Das, Y. C. Hu, and Y. Lu, "Sensor replacement using mobile robots," *Computer Communications*, vol. 30, pp. 2615-2626, 2007.
- [39] R. Handorean, C. Gill, and G. C. Roman, "Accommodating transient connectivity in ad hoc and mobile settings," in *Lecture Notes in Computer Science*, pp. 305-322, Berlin, Germany, 2004.
- [40] S. Jain, K. Fall, and R. Patra, "Routing in a delay tolerant network," in *Proc. of the ACM SIGCOMM*, pp. 145-157, Portland, OR, USA, 2004.
- [41] A. Lindgren, A. Doria, and O. Scheln, "Probabilistic routing in intermittently connected networks," in *Proc. of Workshop on Service Assurance with Partial and Intermittent Resources*, pp. 239-254, Fortaleza, Brazil, 2004.
- [42] J. Burgess, B. Gallagher, D. Jensen, and N. Levine, "MaxProp: routing for vehicle-based disruption-tolerant networks," in *Proc. of 25th IEEE INFOCOM*, pp. 1688-1698, Barcelona, Spain, 2006.
- [43] E. P. C. Jones, L. Li, J. K. Schmidtke, and P. A. S. Ward, "Practical routing in delay-tolerant networks," *IEEE Transactions on Mobile Computing*, vol. 6, pp. 943-959, 2007.
- [44] Y. Wang and H. Wu, "Delay/fault-tolerant mobile sensor network (DFT-MSN): a new paradigm for pervasive information gathering," *IEEE Transactions on Mobile Computing*, vol. 6, pp. 1021-1034, 2007.
- [45] V. Conan, J. Leguay, and T. Friedman, "Fixed point opportunistic routing in delay tolerant networks," *IEEE Journal on Selected Areas in Communications*, vol. 26, pp. 773-782, 2008.
- [46] P. Costa, C. Mascolo, M. Musolesi, and G. P. Picco, "Socially-aware routing for publish-subscribe in delay-tolerant mobile ad hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 26, pp. 748-760, 2008.

- [47] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Efficient routing in intermittently connected mobile networks: the single-copy case," *IEEE/ACM Transactions on Networking*, vol. 16, pp. 63-76, 2008.
- [48] M. K. Bromage, J. T. Koshimoto, and K. Obraczka, "TAROT: Trajectory-Assisted Routing for intermittently connected networks," in *Proc. of 4th ACM Workshop on Challenged Networks*, pp. 9-17, Beijing, China, 2009.
- [49] X. Chen, J. Shen, and J. Wu, "A novel information model for efficient routing protocols in delay tolerant networks," in *Proc. of IEEE International Symposium on Parallel and Distributed Processing*, pp. 1-8, Rome, Italy, 2009.
- [50] M. Musolesi and C. Mascolo, "CAR: context-aware adaptive routing for delay-tolerant mobile networks," *IEEE Transactions on Mobile Computing*, vol. 8, pp. 246-260, 2009.
- [51] P. U. Tournoux, J. Leguay, F. Benbadis, V. Conan, M. D. Amorim, and J. Whitbeck, "The accordion phenomenon: analysis, characterization, and impact on DTN routing," in *Proc. of IEEE INFOCOM*, pp. 1116-1124, Rio de Janeiro, Brazil, 2009.
- [52] H. Chen and W. Lou, "On Using Contact Expectation for Routing in Delay Tolerant Networks," in *Proc. of International Conference on Parallel Processing*, pp. 683-692, Taipei City, Taiwan, 2011.
- [53] J. Wu and Y. Wang, "Social feature-based multi-path routing in delay tolerant networks," in *Proc. of IEEE INFOCOM*, pp. 1368-1376, Orlando, FL, USA, 2012.
- [54] P. Yuan and H. Ma, "Hug: Human gathering point based routing for opportunistic networks," in *Proc. of IEEE Wireless Communications and Networking*, pp. 3024-3029, Shanghai, China, 2012.
- [55] M. M. B. Tariq, M. Ammar, and E. Zegura, "Message ferry route design for sparse ad hoc networks with mobile nodes," in *Proc. of MOBIHOC*, pp. 37-48, New York, NY, USA, 2006.
- [56] Z. Zhang and Z. Fei, "Route design for multiple ferries in delay tolerant networks," in *Proc. of 8th IEEE Wireless Communications and Networking*, pp. 1-6, Kowloon, China, 2007.
- [57] B. Burns, O. Brock, and B. N. Levine, "MORA routing and capacity building in disruption-tolerant networks," *Ad Hoc Networks*, vol. 6, pp. 600-620, 2008.
- [58] X. Yi, C. Huang, and J. Cobb, "Look-ahead routing and message scheduling in delay-tolerant networks," *Computer Communications*, vol. 34, pp. 2184-2194, 2011.
- [59] M. Thomas, S. Phand, and A. Gupta, "Using group structures for efficient routing in delay tolerant networks," *Ad Hoc Networks*, vol. 7, pp. 344-362, 2009.
- [60] H. Dang and H. Wu, "Clustering and Cluster-Based Routing Protocol for Delay-Tolerant Mobile Networks," *IEEE Transactions on Wireless Communications*, vol. 9, pp. 1874-1881, 2010.
- [61] J. Ott, D. Kutscher, and C. Dwertmann, "Integrating DTN and MANET routing," in *Proc. of the ACM SIGCOMM Workshop on Challenged Networks* vol. 2006, ed. Pisa, Italy: Association for Computing Machinery, 2006, pp. 221-228.
- [62] C. Kretschmer, S. Ruhrop, and C. Schindelhauer, "DT-DYMO: delay-tolerant dynamic MANET on-demand routing," in *Proc. of 29th IEEE ICDCS Workshops*, pp. 493-498, Montreal, QC, Canada, 2009.

- [63] J. Whitbeck and V. Conan, "HYMAD: Hybrid DTN-MANET routing for dense and highly dynamic wireless networks," *Computer Communications*, vol. 33, pp. 1483-1492, 2010.
- [64] A. Balasubramanian, B. N. Levine, and A. Venkataramani, "DTN routing as a resource allocation problem," in *Proc. of the ACM SIGCOMM*, pp. 373-384, Kyoto, Japan, 2007.
- [65] V. N. G. J. Soares, J. J. P. C. Rodrigues, F. Farahmand, and M. Denko, "Exploiting Node Localization for Performance Improvement of Vehicular Delay-Tolerant Networks," in *Proc. of IEEE ICC*, pp. 1-5, Cape Town, South Africa, 2010.
- [66] Y. Wang, J. Wu, Z. Jiang, and F. Li, "A joint replication-migration-based routing in delay tolerant networks," in *Proc. of IEEE ICC*, pp. 73-77, Ottawa, ON, Canada, 2012.
- [67] B. Lee, D. Park, and H. Seo, *Wireless Communications Resource Management*, John Wiley & Sons, 2009.
- [68] C. Fragouli, J. Y. L. Boudec, and J. Widmer, "Network coding: an instant primer," *Computer Communication Review*, vol. 36, pp. 63-68, 2006.
- [69] P. A. Chou and Y. Wu, "Network coding for the Internet and wireless networks," *IEEE Signal Processing Magazine*, vol. 24, pp. 77-85, 2007.
- [70] C. Fragouli, D. Katabi, A. Markopoulou, M. Medard, and H. Rahul, "Wireless network coding: opportunities & challenges," in *Proc. of IEEE MILCOM*, pp. 1-8, Orlando, FL, United states, 2007.
- [71] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, "Trading structure for randomness in wireless opportunistic routing," *Computer Communication Review*, vol. 37, pp. 169-180, 2007.
- [72] Y. Lin, B. Li, and B. Liang, "CodeOR: opportunistic routing in wireless mesh networks with segmented network coding," in *Proc. of IEEE International Conference on Network Protocols*, pp. 13-22, Orlando, FL, USA, 2008.
- [73] D. Koutsonikolas, C. C. Wang, and Y. C. Hu, "CCACK: Efficient Network Coding Based Opportunistic Routing Through Cumulative Coded Acknowledgments," in *Proc. of IEEE INFOCOM*, pp. 1-9, Piscataway, NJ, USA, 2010.
- [74] C. Qin, Y. Xian, C. Gray, N. Santhapuri, and S. Nelakuditi, "I2MIX: integration of intra-flow and inter-flow wireless network coding," in *Proc. of IEEE SECON*, pp. 1-6, San Francisco, CA, USA, 2008.
- [75] Y. Yan, B. Zhang, J. Zheng, and J. Ma, "CORE: a coding-aware opportunistic routing mechanism for wireless mesh networks," *IEEE Wireless Communications*, vol. 17, pp. 96-103, 2010.
- [76] H. Seferoglu, A. Markopoulou, and K. K. Ramakrishnan, "I2NC: Intra- and inter-session network coding for unicast flows in wireless networks," in *Proc. of IEEE INFOCOM*, pp. 1035-1043, Shanghai, China, 2011.
- [77] B. Ni, N. Santhapuri, Z. Zhong, and S. Nelakuditi, "Routing with opportunistically coded exchanges in wireless mesh networks," in *Proc. of 2nd IEEE Workshop on Wireless Mesh Networks*, pp. 157-159, Reston, VA, USA, 2006.

- [78] M. A. M. A. Iqbal, B. Dai, B. Huang, A. Hassan, and S. Yu, "Survey of network coding-aware routing protocols in wireless networks," *Journal of Network and Computer Applications*, vol. 34, pp. 1956-1970, 2011.
- [79] T. Matsuda, T. Noguchi, and T. Takine, "Survey of Network Coding and Its Applications," *IEICE Transactions on Communications*, vol. E94-B, pp. 698-717, 2011.
- [80] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft, "XORs in the air: Practical wireless network coding," *IEEE/ACM Transactions on Networking*, vol. 16, pp. 497-510, 2008.
- [81] A. P. Gheibi, A. Ozdaglar, M. Effros, and M. Medard, "Optimal reverse carpooling over wireless networks - A distributed optimization approach," in *Proc. of 44th Annual Conference on Information Sciences and Systems*, pp. 1-6, Princeton, NJ, United states, 2010.
- [82] S. Omiwade, R. Zheng, and C. Hua, "Practical localized network coding in wireless mesh networks," in *Proc. of 5th Annual IEEE SECON*, pp. 332-340, San Francisco, CA, United states, 2008.
- [83] J. Le, J. C. S. Lui, and D. M. Chiu, "DCAR: distributed coding-aware routing in wireless networks," *IEEE Transactions on Mobile Computing*, vol. 9, pp. 596-608, 2010.
- [84] Z. Zhou and L. Zhou, "Network Joint Coding-Aware Routing for Wireless Ad Hoc Networks," in *Proc. of IEEE International Conference on Wireless Communications, Networking and Information Security*, pp. 17-21, Beijing, China, 2010.
- [85] N. M. Jones, B. Shrader, and E. Modiano, "Optimal routing and scheduling for a simple network coding scheme," in *Proc. of IEEE INFOCOM*, pp. 352-360, Orlando, FL, United states, 2012.
- [86] J. Le, J. C. S. Lui, and D. M. Chiu, "How many packets can we encode? -An analysis of practical wireless network coding," in *Proc. of IEEE INFOCOM*, pp. 1040-1048, Phoenix, AZ, United states, 2008.
- [87] S. Sengupta, S. Rayanchu, and S. Banerjee, "Network coding-aware routing in wireless networks," *IEEE/ACM Transactions on Networking*, vol. 18, pp. 1158-1170, 2010.
- [88] L. Huang and C. W. Sung, "An Iterative Routing Algorithm for Energy Minimization in Coded Wireless Networks," in *Proc. of IEEE PIMRC*, pp. 1124-1128, Toronto, ON, Canada, 2011.
- [89] S. Shabdanov, C. Rosenberg, and P. Mitran, "Joint routing, scheduling, and network coding for wireless multihop networks," in *Proc. of IEEE WiOpt*, pp. 33-40, Princeton, NJ, USA, 2011.
- [90] P. S. Mogre, N. Heuresse, M. Hollick, and R. Steinmetz, "A case for joint routing, scheduling, and network coding in TDMA-based wireless mesh networks: a cross-layer approach," in *Proc. of IEEE International Conference on Mobile Adhoc and Sensor Systems*, pp. 658-660, Pisa, Italy, 2007.
- [91] B. Scheuermann, W. Hu, and J. Crowcroft, "Near-Optimal Co-ordinated Coding in Wireless Multihop Networks," in *Proc. of the ACM CoNEXT*, 2007.
- [92] D. Koutsonikolas, Y. C. Hu, and C. C. Wang, "An empirical study of performance benefits of network coding in multihop wireless networks," in *Proc. of IEEE INFOCOM*, pp. 2981-2985, Rio de Janeiro, Brazil, 2009.

- [93] D. Umehara, T. Hirano, S. Denno, M. Morikura, and T. Sugiyama, "Wireless network coding in slotted ALOHA with two-hop unbalanced traffic," *IEEE Journal on Selected Areas in Communications*, vol. 27, pp. 647-661, 2009.
- [94] D. Umehara, S. Denno, M. Morikura, and T. Sugiyama, "Throughput Analysis of Two-Hop Wireless CSMA Network Coding," in *Proc. of IEEE ICC*, pp. 1-6, Cape Town, South Africa, 2010.
- [95] F. Zhao and M. Medard, "On analyzing and improving COPE performance," in *Proc. of IEEE Workshop on Information Theory and Applications*, pp. 317-322, San Diego, CA, USA, 2010.
- [96] H. Guo, Y. Qian, K. Lu, and N. Moayeri, "Backbone routing over multihop wireless networks: Increased network coding opportunity," in *Proc. of IEEE ICC*, Cape Town, South africa, 2010.
- [97] S. Wang, A. Vasilakos, H. Jiang, X. Ma, W. Liu, K. Peng, *et al.*, "Energy Efficient Broadcasting Using Network Coding Aware Protocol in Wireless Ad Hoc Network," in *Proc. of IEEE ICC*, pp. 1-5, Kyoto, Japan, 2011.
- [98] X. Jiao and X. Wang, "Active network coding based high-throughput optimizing routing for wireless ad hoc networks," in *Proc. of 4th International Conference on Wireless Communications, Networking and Mobile Computing*, Dalian, China, 2008.
- [99] Y. Wu, S. M. Das, and R. Chandra, "Routing with a Markovian metric to promote local mixing," in *Proc. of IEEE INFOCOM*, pp. 2380-2384, Anchorage, AK, USA, 2007.
- [100] Y. Yan, Z. Zhao, B. Zhang, H. T. Mouftah, and J. Ma, "Rate-adaptive coding-aware multiple path routing for wireless mesh networks," in *Proc. of IEEE GLOBECOM*, New Orleans, LO, USA, 2008.
- [101] M. Jhang, S. W. Lin, and W. Liao, "C2AR: Coding and Capacity Aware Routing for Wireless Ad Hoc Networks," in *Proc. of IEEE ICC*, Cape Town, South Africa, 2010.
- [102] B. Guo, H. Li, C. Zhou, and Y. Cheng, "Analysis of General Network Coding Conditions and Design of a Free-Ride-Oriented Routing Metric," *IEEE Transactions on Vehicular Technology*, vol. 60, pp. 1714-1727, 2011.
- [103] N. Gaddam, S. A. Gathala, D. Lastine, and A. Somani, "Energy minimization through network coding for lifetime constrained wireless networks," in *Proc. of 17th International Conference on Computer Communications and Networks*, pp. 471-476, St. Thomas, US Virgin Islands, USA, 2008.
- [104] A. Eryilmaz and D. S. Lun, "Control for Inter-Session Network Coding," presented at the Proc. of IEEE International Symposium on Network Coding, San Diego, California, USA, 2007.
- [105] J. Price and T. Javidi, "Network coding games with unicast flows," *IEEE Journal on Selected Areas in Communications*, vol. 26, pp. 1302-1316, 2008.
- [106] Y. Peng, Y. Yang, X. Lu, and X. Ding, "Coding-aware Routing for Unicast Sessions in Multi-hop Wireless Networks," in *Proc. of IEEE GLOBECOM*, Miami, FL, USA, 2010.
- [107] Y. Benfattoum, S. Martin, and K. A. Agha, "IROCX: Interference-aware routing with opportunistically coded exchanges in wireless mesh networks," in *Proc. of IEEE Wireless Communications and Networking Conference*, pp. 1113-1118, Cancun, Quintana Roo, Mexico, 2011.

- [108] T. Kim, S. Vural, I. Broustis, D. Syrivelis, S. V. Krishnamurthy, and T. F. L. Porta, "A Framework for Joint Network Coding and Transmission Rate Control in Wireless Networks," in *Proc. of IEEE INFOCOM*, pp. 1-9, San Diego, CA, USA, 2010.
- [109] B. Ni, N. Santhapuri, C. GraY, and S. Nelakuditi, "Selection of bit-rate for wireless network coding," in *Proc. of 5th Annual IEEE SECON*, pp. 1-6, San Francisco, CA, USA, 2008.
- [110] J. Yomo and P. Popovski, "Opportunistic scheduling for wireless network coding," *IEEE Transactions on Wireless Communications*, vol. 8, pp. 2766-2770, 2009.
- [111] Y. Kim and G. D. Veciana, "Is rate adaptation beneficial for inter-session network coding?," *IEEE Journal on Selected Areas in Communications*, vol. 27, pp. 635-646, 2009.
- [112] J. El-Najjar, H. M. K. AlAzemi, and C. Assi, "On the Interplay Between Spatial Reuse and Network Coding in Wireless Networks," *IEEE Transactions on Wireless Communications*, vol. 10, pp. 560-569, 2011.
- [113] P. Chaporkar and A. Proutiere, "Adaptive network coding and scheduling for maximizing throughput in wireless networks," in *Proc. of the ACM MobiCom*, pp. 135-146, Montreal, QC, Canada, 2007.
- [114] C. H. Liu and F. Xue, "Network coding for two-way relaying: rate region, sum rate and opportunistic scheduling," in *Proc. of IEEE ICC*, pp. 1960-1965, Beijing, China, 2008.
- [115] J. Zhang and Q. Zhang, "Cooperative network coding-aware routing for multi-rate wireless networks," in *Proc. of IEEE INFOCOM*, pp. 181-189, Rio de Janeiro, Brazil, 2009.
- [116] R. Groenevelt, P. Nain, and G. Koole, "The message delay in mobile ad hoc networks," *Performance Evaluation*, vol. 62, pp. 210-228, 2005.
- [117] X. Zhang, G. Neglia, J. Kurose, and D. Towsley, "Performance modeling of epidemic routing," *Computer Networks*, vol. 51, pp. 2867-2891, 2007.
- [118] Y. Lin, B. Li, and B. Liang, "Stochastic analysis of network coding in epidemic routing," *IEEE Journal on Selected Areas in Communications*, vol. 26, pp. 794-808, 2008.
- [119] S. Y. R. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE Transactions on Information Theory*, vol. 49, pp. 371-381, 2003.
- [120] R. Dougherty, C. Freiling, and K. Zeger, "Insufficiency of linear coding in network information flow," *IEEE Transactions on Information Theory*, vol. 51, pp. 2745-2759, 2005.
- [121] Y. D. Lin and Y. C. Hsu, "Multihop cellular: a new architecture for wireless communications," in *Proc. of IEEE INFOCOM*, pp. 1273-1282, Tel Aviv, Israel, 2000.
- [122] X. Li, B. Seet, and P. H. J. Chong, "Multihop cellular networks: Technology and economics," *Computer Networks*, vol. 52, pp. 1825-1837, 2008.
- [123] K. Jain, J. Padhye, V. N. Padmanabhan, and L. Qiu, "Impact Of Interference On Multi-hop Wireless Network Performance," in *Proc. of the ACM MobiCom 2003*, pp. 66-80, San Diego, CA., United states, 2003.

- [124] J. J. Widmer and J. L. Boudec, "Network coding for efficient communication in extreme networks," in *Proc. of the ACM SIGCOMM Workshops on Computer Communications*, pp. 284-291, Philadelphia, PA, United states, 2005.
- [125] S. Ahmed and S. S. Kanhere, "HUBCODE: Message forwarding using hub-based network coding in delay tolerant networks," in *Proc. of 12th ACM MSWiM*, pp. 288-296, Tenerife, Canary Islands, Spain, 2009.
- [126] Z. Li, D. Zeng, S. Guo, S. Lu, D. Chen, and W. Zhuang, "On the throughput of feedbackless segmented network coding in delay tolerant networks," *IEEE Wireless Communications Letters*, vol. 1, pp. 93-96, 2012.
- [127] X. Zhang, G. Neglia, J. Kurose, D. Towsley, and H. Wang, "Benefits of network coding for unicast application in disruption-tolerant networks," *IEEE/ACM Transactions on Networking*, 2012.
- [128] E. M. Royer and C. K. Toh, "A review of current routing protocols for ad hoc mobile wireless networks," *IEEE Personal Communications*, vol. 6, 1999.