IMPROVING THE PERFORMANCE AND SECURITY OF MULTI-HOP WIRELESS NETWORKS

By

CHI ZHANG

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

2011

To all who nurtured my intellectual curiosity, academic interests, and sense of scholarship throughout my lifetime, making this milestone possible

ACKNOWLEDGMENTS

First and foremost, I would like to express my sincere gratitude to my advisor, Prof. Yuguang Fang, for his invaluable guidance, encouragement and support with my years in Wireless Networks Laboratory (WINET). He convinced me to join the Ph.D. program at UFL seven years ago, and encouraged me in my pursuit of an academic career after graduation. I am looking forward to our collaboration in the future.

I also would like to thank Professor Pramod Khargonekar, Professor Dapeng Wu, and Professor Shigang Chen for serving on my supervisory committee and for their great help in various stages of my work and career.

I would not be a sane graduate student without a group of great friends. I would like to extend my thanks to all my colleagues in WINET for providing me a warm, family-like environment and for their collaboration and insightful advice. I specially thank Dr. Wenjing Lou, Dr. Wenchao Ma, Dr. Byung-Seo Kim, Dr. Wei Liu, Dr. Xiang Chen, Dr. Jing Zhao, Dr. Hongqiang Zhai, Dr. Yanchao Zhang, Dr. Shushan Wen, Dr. Jianfeng Wang, Dr. Yun Zhou, Dr. Xiaoxia Huang, Dr. Pan Li, Dr. Feng Chen, Dr. Jinyuan Sun, Dr. Yang Song, Dr. Rongsheng Huang, Frank Goergen, Miao Pan, Hao Yue, Linke Guo, Huang Lin, Yuanxiong Guo, Zongrui Ding, Xinxin Liu, Dr. Sunmyeng Kim, Dr. Nicola Scalabrino, Dr. Roberto Riggio, Dr. Qiang Shen, Dr. Zhiqiang Shi, Dr. Xiaoyan Yin, Dr. Xiaobin Tan, Dr. Guoliang Yao and Dr. Xihua Dong for many valuable discussions and all the good memories.

Special thanks are due to the Fang family: my advisor Yuguang Fang and his wife Jennifer Lu, and to the Huang family: Rongsheng and Haiyan, who not only constantly encouraged me and helped me in many ways, but also shared their view of life with me.

Finally, I owe a special debt of gratitude to my beloved parents and sister. Without their love and unwavering support, I would never imagine what I have achieved.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Abstract of Dissertation Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy

IMPROVING THE PERFORMANCE AND SECURITY
OF MULTI-HOP WIRELESS NETWORKS

By

Chi Zhang

August 2011

Chair: Yuguang Fang
Major: Electrical and Computer Engineering

Multi-hop wireless networks (or wireless ad hoc networks) have been widely accepted as an indispensable component of next-generation communication systems to facilitate ubiquitous network access from anywhere at any time. Although offering significant benefits, they also provide unique research challenges over their wired counterparts. Of note are the issues associated with the design of efficient routing protocols and network security, etc.

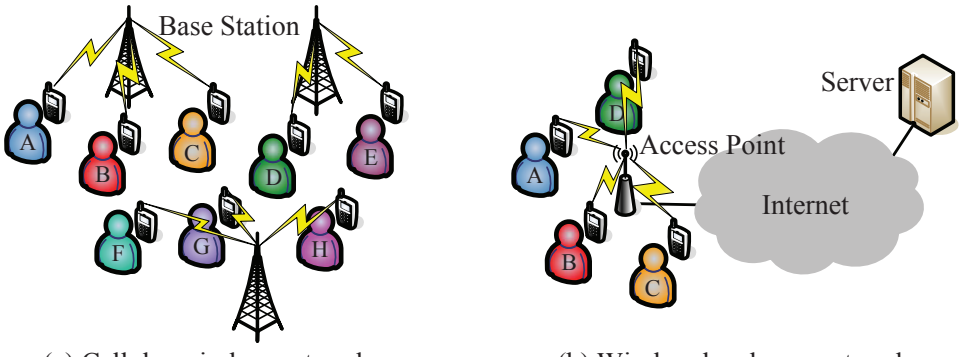
In this dissertation, we aim to address these challenging and fundamental issues in heterogeneous, large-scale multi-hop wireless networks, spanning mobile ad hoc networks, wireless sensor networks, and multi-hop cellular networks. Our contributions are mainly sixfold. First, for a wireless sensor network, we propose a coverage inference protocol which can provide the base station an accurate and in-time measurement of connected coverage with minimized overhead. Second, we consider decentralized routing problems in heterogeneous ad hoc networks, wherein each node is connected to all its neighbors within some fixed radius, as well as possessing random ling-range links to more distant nodes. We characterize the necessary and sufficient condition for greedy geographic routing to be efficient with nonhomogeneous node distribution. Third, we study the throughput-delay tradeoffs in mobile ad hoc networks with network coding, and compare results with the situation where only replication and forwarding are allowed in each node. Forth, we propose a novel and promising incentive paradigm called C4 to

induce cooperative behaviors in multi-hop cellular networks with minimized overhead. Fifth, we study the relationships between varying trust metrics and trust-based routing protocols. By developing a formal model to describe different trust environment, we identify the basic algebraic properties that a trust metric must have in order to guarantee the correctness and optimality of different wireless routing protocols. Last, we studies the relationships between network security requirement and network performance degradation. We characterize the asymptotic behaviors of achievable secure throughput and delay when the network size is sufficiently large.

CHAPTER 1
INTRODUCTION

1.1 Wireless Ad Hoc Networks: An Overview

Recent years have witnessed a surge of research and development for wireless ad hoc networks (or wireless multi-hop networks) as they have tremendous military and commercial potential. A wireless ad hoc network (WANET) [119, 131] is a wireless network, comprised of mobile computing devices that use wireless transmission for communication, having no fixed infrastructure (a central administration such as a base station in a cellular wireless network or an access point in a wireless local area network, cf. Figure 1-1). The mobile devices also serve as routers due to the limited range of wireless transmission of these devices, that is, several devices may need to route or relay a packet before it reaches its final destination (cf. Figure 1-2). Ad hoc wireless networks can be deployed quickly anywhere and anytime as they eliminate the complexity of infrastructure setup. These networks find applications in several areas. Some of these include: military communications (establishing communication among a group of soldiers for tactical operations when setting up a fixed wireless communication infrastructure in enemy territories or in inhospitable terrains may not be possible), emergency systems (for example, establishing communication among rescue personnel in disaster-affected areas) that need quick deployment of a network, collaborative and distributed computing, and hybrid (integrated cellular and ad hoc) wireless networks.



(a) Cellular wireless network (b) Wireless local area network

Figure 1-1. Examples of wireless networks with infrastructures.

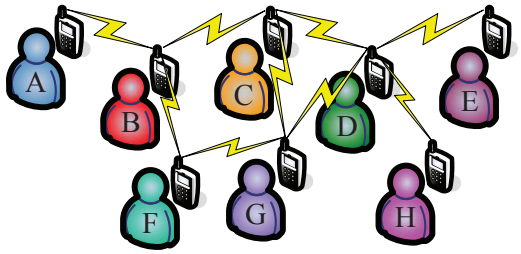


Figure 1-2. Example of a wireless ad hoc network (wireless network without infrastructure).

In general, wireless ad hoc networks can be classified into two categories, mobile ad hoc networks (MANETs) and static ad hoc networks. The former comprise network nodes that are free to move about randomly and organize themselves arbitrarily. Exemplary application scenarios of MANETs include tactical military operations, homeland security, emergency disaster relief and rescue, and so on. Most recently, MANETs have been extended to general civilian contexts and are often referred to as wireless mesh networks [5], where mobile users can access the network either through a direct wireless link to a wireless access point (AP), or through a sequence of intermediate users to an AP that is too far away to reach. By contrast, static ad hoc networks mainly consist of stationary nodes, that is, fixed at where they were deployed. The most significant example of this later type is wireless sensor networks [4], which have attracted extensive attention in both academia and industry for their broad potential in not only military and homeland security scenarios but also in general civilian settings.

The most fundamental functionality of any wireless ad hoc network is to provide end-to-end communication in a peer-to-peer (P2P) fashion, i.e., without any infrastructure (or with partial infrastructure). Figure 1-2 depicts the peer-level multi-hop representation of such a network. Mobile node A communicates with another such node B directly (single-hop) whenever a radio channel with adequate propagation characteristics is available between them. Otherwise, multi-hop communication is necessary where one or more intermediate nodes must act as a relay (router) between the communicating nodes. For example, there is no direct radio channel (shown by the lines) between A

and C or A and G in Figure 1-2. Nodes B and F must, therefore, serve as intermediate routers for communication between A and C, and A and G. Indeed, a distinguishing feature of ad hoc networks is that all nodes must be able to function as routers on demand. To prevent packets from traversing infinitely long paths, an obvious essential requirement for choosing a path is that the path must be loop-free. A loop-free path between a pair of nodes is called a route.

1.2 Research Challenges

While offering significant benefits, wireless ad hoc networks also provide unique research challenges over their wired counterparts. This subsection outlines the major problems that ought to be addressed. The protocol dependent development possibilities are mostly omitted and the focus is on the "big picture", on the problems that stand in a way of having peer-to-peer connectivity everywhere in the future.

1.2.1 Scalability

Most of the visionaries depicting applications which are anticipated to benefit from the ad hoc and sensor networking technology take scalability as granted. Imagine, for example, the vision of ubiquitous computing where networks can be of "any size". However, it is unclear how such large networks can actually grow.

Ad hoc networks suffer, by nature, from the scalability problems in capacity. To exemplify this, we may look into simple interference studies. In a non-cooperative network, where omni-directionaLantennas are being used, the throughput per node decreases at a rate $1/\sqrt{n}$, where n is the number of nodes [60]. That is, in a network with 100 nodes, a single device gets, at most, approximately one tenth of the theoretical network data rate. This problem, however, cannot be fixed except by physical layer improvements, such as directional antennas.

If the available capacity sets some limits for communications, so do the protocols. Route acquisition, service location and encryption key exchanges are just few examples of tasks that will require considerable overhead as the network size grows. If the scarce

resources are wasted with profuse control traffic, these networks may see never the day dawn. Therefore, scalability is a crucial research topic and has to be taken into account in the design of solutions for ad hoc and sensor networks.

1.2.2 Heterogeneity

Network heterogeneity is a certainty for today's wireless networks. There are two kinds of heterogeneity: first, the distribution of wireless users/devices in the physical space is non-homogeneous; second, wireless devices are likely to have widely varying radio ranges (e.g., cellular/WiMax, WiFi, Zigbee). For a heterogeneous ad hoc network, where short-range wireless links and long-range wireless links (shortcuts) coexist, how to design efficient decentralized routing protocols with local information is an open problem in the literature.

1.2.3 Client-Server Model Shift

In the Internet, a network client is typically configured to use a server as its partner for network transactions. These servers can be found automatically or by static configuration. In ad hoc networks, however, the network structure cannot be defined by collecting IP addresses into subnets. There may not be servers, but the demand for basic services still exists. Address allocation, name resolution, authentication and the service location itself are just examples of the very basic services which are needed but their location in the network is unknown and possibly even changing over time. Due to the infrastructureless nature of these networks and node mobility, a different addressing approach may be required. In addition, it is still not clear who will be responsible for managing various network services. Therefore, while there has been vast research initiatives in this area, the issue of shift from the traditional client-sever model remains to be appropriately addressed.

1.2.4 Security

Wireless ad hoc networks are particularly prone to malicious behavior. Lack of any centralized network management or certification authority makes these dynamically

changing wireless structures very vulnerable to infiltration, eavesdropping, interference, and so on. Security is often considered to be the major “roadblock” in the commercial application this technology. Security is indeed one of the most difficult problems to be solved. Some of the major security challenges that a wireless ad hoc network faces include the following:

- All old threats to a conventional wired network apply to a wireless ad hoc network.
- The shared wireless medium facilitates passive eavesdropping on data communications and/or active bogus message injection into the network by attackers.
- Early protocol design for wireless ad hoc networks all assumed a friendly and cooperative environment. As such, many wireless protocols have inherent security flaws.
- Mobile devices are subject to physical theft or loss, leading to insider attacks launched by attackers harnessing confidential information extracted from stolen devices.
- Intrusion detection is far more difficult, mainly because it is hard to differentiate anomalies caused by characteristics of wireless channels and those caused by attacks.
- There is often lack of an on-line centralized authority or administration.
- Mobile devices usually have stringent resource constraints and thus cannot afford resource-hungry security protocols.

1.3 Scope and Organization of the Dissertation

This dissertation contributes to the development of novel solutions to a number of challenging and fundamental issues in wireless multi-hop networks (or wireless ad hoc networks), which are either ignored or not well addressed in previous research. The rest of the dissertation can be divided into two main parts as follows.

1.3.1 Chapters on Network Performance

From Chapter 2 to Chapter 4, we will discuss several techniques to improve network performances, like coverage, throughput, and delay.

Connected coverage, which reflects how well a target field is monitored under the base station (BS), is the most important performance metrics used to measure the quality of surveillance that wireless sensor networks can provide. In Chapter 2, we propose a coverage inference protocol (CIP) which can provide the BS an accurate and in-time measurement

Chapter 3 considers decentralized routing problems in heterogeneous ad hoc networks, wherein each node is connected to all its neighbors within some fixed radius, as well as possessing random shortcuts to more distant nodes. We show that with nonhomogeneous node distribution, the necessary and sufficient condition for greedy geographic routing to be efficient is that the probability of a shortcut being present from node u to v should be inversely proportional to the number of nodes which are closer to u than v is. To the best of our knowledge, this is the first work to prove this result in the nonhomogeneous continuum setting.

Chapter 4 characterizes the throughput-delay tradeoffs in mobile ad hoc networks (MANETs) with network coding, and compares results with the situation where only replication and forwarding are allowed in each node. The schemes/protocols achieving those tradeoffs in an effective and decentralized way are proposed and the optimality of those tradeoffs is established.

1.3.2 Chapters on Network Security

We will concentrate network security problems in the following three chapters.

For a multi-hop wireless network (MWN) consisting of mobile nodes controlled by independent self-interested users, incentive mechanism is essential for motivating mobile nodes to cooperate and forward packets for each other. In Chapter 5, we propose a novel and promising incentive paradigm, Controlled Coded packets as virtual Commodity Currency (**C4**), to induce cooperative behaviors in MWNs. In our **C4**, through introducing several techniques from network coding, coded information packets are utilized as a new kind of virtual currency to facilitate packet/service exchanges

among self-interested nodes in a MWN. Since the virtual currency implemented in this way also carries useful data information, it is the counterpart of the so-called commodity currency in the physical world, and the overhead brought by **C4** is extremely small compared to traditional schemes.

Chapter 6 studies the relationships between trust metrics and trust-based routing protocols. By developing a formal models to describe different trust environment, we identify the basic algebraic properties that a trust metric must have in order to guarantee the correctness and optimality of different generalized distance-vector or link-state routing protocols in wireless ad hoc networks. The proposed research provides a new methodology for the formal analysis of wireless network security and accelerates the evaluation, design and real deployment of trust-based routing protocols.

Security always comes with a price in terms of performance degradation, which should be carefully quantified. This is especially the case for wireless ad hoc networks which offer communications over a shared wireless channel without any pre-existing infrastructure. In Chapter 7, based on a general random network model, the asymptotic behaviors of secure throughput and delay with the common transmission range r_n and the probability p_f of neighboring nodes having a primary security association are quantified when the network size n is sufficiently large.

Finally, Chapter 8 summarizes this dissertation and points out some future research directions.

CHAPTER 2
DESIGNING COVERAGE INFERENCE PROTOCOLS FOR WSNS

2.1 Chapter Overview

A wireless sensor network (WSN) is a large collection of densely deployed, spatially distributed, autonomous devices (or nodes) that communicate via wireless and cooperatively monitor physical or environmental conditions [4, 79]. In such network, sensor nodes are deployed over a geographic area (called the region of interest or ROI) by aerial scattering or other means. Each sensor node can only detect events within some very limited distance from itself, called the sensing range. In addition, sensor nodes normally have fairly limited transmission and reception capabilities so that sensing data have to be relayed via a multi-hop path to a distant base station (BS), which is a data collection center with sufficiently powerful processing capabilities and resources. After being deployed, sensor nodes are usually left unattended for a very long period of time so that they may fail over time due to various reasons such as battery exhaustion and physical destructions by attackers. They may also be moved away from where they were deployed by animals, winds, or other environmental means. As a consequence of node failures, node movements and other unpredictable factors, the network topology may change with time. It is, therefore, critical that the BS learn in real time how well the WSN performs the given sensing task under dynamically changing network topology.

From the BS's (or user's) point of view, a position in the ROI is really under the surveillance of the WSN if and only if this position is within the sensing range of at least one sensor node connected to the BS. We define the collection of all these positions in the ROI as the connected coverage (or coverage in short). Obviously, it is one of the most important performance metric measuring the quality of surveillance a WSN can provide. The BS also should have the ability to monitor the coverage status in real time.

Although much research [10, 40, 45, 59, 69, 92, 93] has been conducted to ensure high network coverage and connectivity for the WSN, none of them addresses how to help the BS infer the coverage boundary when coverage holes emerge. Possible causes leading to coverage holes include energy depletion of sensor nodes, intended attacks on sensor nodes, and so on. In many WSN applications, especially security-sensitive applications, it is a must to accurately detect the coverage boundary. The protocol developed in this chapter can affirmatively answer this open challenging issue.

On the other hand, problems related to the self-monitoring of a WSN have been studied in the literature for various applications and purposes. For example, Chessa and Santi [137] propose a single time-out scheme to monitor the system-level fault diagnosis. In [179], a residual energy scan is designed to approximately depict the remaining energy distribution within a WSN. In addition, a self-monitoring mechanism for detecting node failures is proposed in [68]. However, All these schemes cannot be directly used for the coverage inference, as they are either centralized schemes or assume that each individual sensor in the WSN needs to be monitored. This is not true for our case because the BS only needs to ensure that a ceratin percentage of the sensors are functioning, especially when the WSN is densely deployed.

Generally, we can distinguish two basic types of WSNs: proactive and reactive. Proactive WSNs involve a periodic data delivery between sensor nodes and the BSs. By comparison, in reactive WSNs, packets are sent only when some event of interest occurs and is sensed. Although for proactive WSNs, each node can simply help the BS infer the connected coverage by piggybacking its status information on data traffic, it is well known that proactive WSNs are energy inefficient and not scalable [21, 79]. Therefore, in this chapter, we focus on providing coverage inference for the BS in reactive WSNs. However, due to the very nature of WSNs, this task is far from triviality. The most significant challenge is due to the strict resource limitation of sensor nodes (battery power, memory, computational capability, etc.) which highlights the need for

a localized solution. Although there is plenty of work [40, 53, 62, 69, 157, 174] on the localized coverage boundary detection, none of them is adequate because some of them such as polygon-based approaches [40, 53, 157] are not truly localized solutions and suffer from possibly significant communication overhead, while others such as perimeter-based schemes [62, 69, 174] are inefficient when the node density is high (cf. Section 2.5). Generalization of such schemes to all the situations we are interested in is not trivial.

This chapter makes the following contributions. First, we present a Coverage Inference Protocol (CIP) which can provide the BS an accurate and in-time measurement of the current connected coverage in an energy-efficient way. Second, we show that the major component of our CIP - BOundary Node Detection (BOND) scheme - can be reused to provide many other functionalities for WSNs, such as topology control, efficient routing, sleeping scheduling, and spatial aggregation. Therefore, our schemes can be exploited to seamlessly integrate multiple functionalities with low overhead. The performance of our BOND-based CIP compared with other possible CIPs is also investigated. Moreover, we devise extensions to CIP which can tolerate location errors and actively predict the change of the connected coverage by utilizing the information of residual energy information of sensor nodes.

The rest of this chapter is organized as follows. Section 2.2 introduces the network model and the design goals of our CIP. Next we detail the core component of our solution called BOND in Section 2.3. This is followed by presenting the complete CIP in Section 2.4 and its comparison with other possible alternatives in the design space in Section 2.5. We then present several extensions to CIP in Section 2.6 and end with conclusions and future work.

2.2 Preliminaries

2.2.1 Network Model

Throughout this chapter, we assume that any two sensor nodes can directly communicate via bi-directional wireless links if their Euclidean distance is not greater than r_c, the communication range; and a position in the plane can be perfectly monitored (or covered) by a sensor node if their Euclidean distance is not greater than r_s, the sensing range. Similar to [10, 93, 159], we also assume that sensor nodes are homogeneous in the sense that r_c and r_s are the same for all nodes, and keep constant during each node's lifetime.

Instead of considering all the possible combinations of r_c and r_s, we focus on the case of $r_c = 2r_s$ in this chapter. There are two reasons for doing so. First, as pointed out in [174], the specification of $r_c \geq 2r_s$ holds for most commercially available sensors such as Berkeley Motes and Pyroelectric infrared sensors. Second, as shown in Section 2.3.6, for arbitrary spatial distributions of sensor nodes, $r_c \geq 2r_s$ is the sufficient and necessary condition for the existence of local boundary node detection algorithms[1] . Therefore, we set $r_c = 2r_s$ to reduce communication energy consumption and interference. However, it should be noted that our algorithms are still applicable to the scenarios of $r_c > 2r_s$ without any changes. We also present a scheme called SAB to deal with the case of $r_c < 2r_s$ (cf. Section 2.5.2.2).

For simplicity, we assume that the ROI is a 2-D square planar field hereafter. Our results, however, can be easily extended to 2-D or 3-D ROIs of arbitrary shapes. For $l > 0$, let A_l denote the square ROI of side length l centered at the origin, i.e., $A_l = [-l/2, l/2]^2$, and ∂A_l be the border of A_l. We examine a large-scale WSN consisting of

[1] The formal definition of "boundary nodes" and "local algorithms" will be given in Sections 2.3.1 and 2.3.6, respectively.

hundreds or even thousands of stationary sensor nodes[2] , and denote the sensor nodes deployed in the ROI to be $V = \{s_1, \cdots, s_i, \cdots, s_n\}$ ($s_i \in A_I$, for $1 \leq i \leq n, i \in \mathbb{N}$), where s_i represents the position of node i and n is the total number of sensor nodes (or network size).

In general, no assumption should be made about the distribution of the sensor nodes in the environment. Our schemes are designed to work correctly under arbitrary node distributions. However, in the performance evaluation of the schemes proposed in this chapter, we utilize homogeneous Spatial Poisson Point Process (SPPP) as the node distribution model to facilitate the theoretical analysis and simulations. It is well known that this model is a good approximation of the distribution of sensor nodes massively or randomly deployed (e.g, via aerial scattering or artillery launching) and can be easily extended to characterize the process that nodes fail dynamically.

2.2.2 Design Goals

In this chapter, we intend to design a coverage inference protocol which can provide the BS an accurate and timely measurement of the current connected coverage in an energy-efficient way. Specifically, our design goals include:

Effectiveness and robustness: The BS should be able to have a timely and accurate view of the connected coverage, regardless of arbitrary network topologies, location errors of sensor nodes and error-prone wireless channels such that it can instruct necessary and quick actions, e.g., adding new sensor nodes to enlarge the connected coverage.

[2] Stationary nodes here do not imply that the topology of the WSN is static. Instead, the WSN may have highly dynamic topology changes due to node failures, new node additions or nodes switching their states between active and sleeping modes to save energy. One advantage of our schemes lies in the efficiency to handle topology changes in WSNs (cf. Section 2.3.5).

Truly localized and distributed properties: As compared to previous approaches, the coverage inference protocol is intended to be a truly localized and distributed solution, in which each sensor node can self-determine whether it is on the area-coverage boundary by a few simple computations on information only from one-hop neighbors. These nice properties will enable the protocol to have low computational and communication overhead, high energy efficiency, and excellent network scalability.

Universal applicability: Although WSNs are often said to be highly application -dependent [4, 79], the coverage inference protocol is designed to work with arbitrary applications and network topologies and to be independent of all the other components of the network protocol stack.

Versatility: For resource-constrained sensor networks, it is highly desirable that some basic protocol operations for implementing one functionality can be reused in providing other necessary functionalities. Otherwise the protocol stack of sensor nodes will be too complicated to have high operational efficiency [79]. The design of coverage inference protocol will take this requirement into consideration so that many of its basic operations can be highly reused in realizing important functionalities other than network health diagnosis, as we will show soon.

2.3 BOND: Boundary Node Detection Scheme

This section presents BOND which enables each node to locally self-detect whether it is a boundary node. We begin with several important definitions, followed by the illustration of BOND.

2.3.1 Boundary Node and Its Detection Algorithm

We say that nodes s_i and s_j ($i \neq j$ and $s_i, s_j \in V$) are neighbors or there exists a direct wireless link between them if the Euclidean distance between them is no larger than r_c, i.e., $\|s_i - s_j\| \leq r_c$. We also denote by $Neig(s_i)$ the neighbors of node s_i (not including s_i). In addition, two nodes s_i and s_j are said to be connected if there is at least one path consisting of direct wireless links between them, similarly a set of nodes is

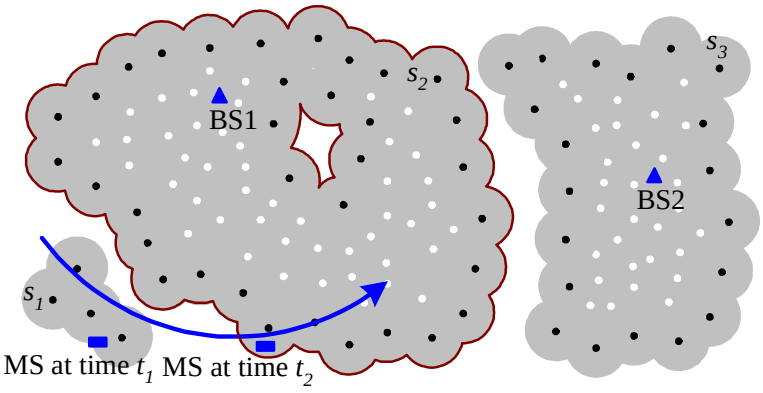


Figure 2-1. An exemplary WSNs. There are three sensor clusters: $Clust\,(s_1)$, $Clust\,(s_2)$, and $Clust\,(s_3)$, two static BSs: $BS1$ and $BS2$ and one mobile BS: MS. Shaded area, solid dots and open dots represent the coverage of sensors, boundary nodes and interior nodes, respectively. The boundary of $Cover\,(s_2)$ is depicted by red curves.

called connected if at least one path exists between each pair of nodes in the set. The fundamental connected unit of WSNs is called a cluster:

Definition 2.1. ***[cluster]*** *A connected set of nodes is said to be a* ***cluster*** *if the inclusion of any other node not in this set will break the connectedness property.*

We write $Clust\,(s_i)$ for the cluster containing node s_i. Based on the sensing model, the sensing disk (or coverage) of node s_i can be given by

$$Disk_i = Disk(s_i, r_s) = \{u \in \mathbb{R}^2 : \|u - s_i\| \leq r_s\}. \tag{2–1}$$

Specifically, let $\mathbf{0}$ indicate the origin, we have $Disk_0 = Disk(\mathbf{0}, r_s)$. Then the coverage corresponding to a cluster can be defined as follows:

Definition 2.2. ***[coverage of a cluster]*** *Given $Clust\,(s_i)$, we refer to the set of all points in the monitored field that are within radius r_s from any node of $Clust\,(s_i)$ as the set* ***covered*** *by cluster $Clust\,(s_i)$. Denoting this set by $Cover\,(s_i)$, we have*

$$Cover\,(s_i) = \left(\bigcup_{u \in Clust(s_i)} (u + Disk_0) \right) \bigcap A_I. \tag{2–2}$$

Obviously, $Cover(s_i)$ is uniquely determined by its boundary $\partial Cover(s_i)$.

Definition 2.3. ***[boundary and interior node]*** *We define* ***boundary nodes*** *of* $Clust(s_i)$ *as those whose minimum distances to* $\partial Cover(s_i)$ *are equal to* r_s*, i.e.,*

$$BN(s_i) = \{u \in Clust(s_i) : \min \|u - v\| = r_s \text{ for } v \in \partial Cover(s_i)\}. \tag{2–3}$$

Accordingly, we call all the other nodes in $Clust(s_i)$ *as* ***interior nodes****, i.e.,*

$$IN(s_i) = \{u \in Clust(s_i) : u \notin BN(s_i)\}. \tag{2–4}$$

The minimum information needed to describe $\partial Cover(s_i)$ is r_c and $BN(s_i)$. We denote the position of the base station as BS, then the connected coverage of the BS is[3]

$$Cover(BS) = \left\{ \bigcup_{1 \le i \le n} Cover(s_i) : BS \bigcap \left(Clust(s_i) \oplus Disk(\mathbf{0}, r_c)\right) \neq \emptyset \right\}. \tag{2–5}$$

Obviously, the problem of finding the boundary of connected coverage, i.e., $\partial Cover(BS)$, is equivalent to detecting the boundary nodes of clusters with connections to the BS. Based on this observation, it is possible to design a distributed CIP if we can first find a localized way to detect boundary nodes.

Note that our definition of boundary nodes is uniquely based on the cluster they belong to and is unrelated to the position of the BS. In addition, our definition of connected coverage is applicable to WSNs with multiple or mobile BSs. For example, in the WSN given in Figure 2-1 where there are three BSs (two static and one mobile), from Eq. (2–5) we can directly obtain the connected coverage of the WSN at time instances t_1 and t_2 as $Cover(s_1) \bigcup Cover(s_2) \bigcup Cover(s_3)$ and $Cover(s_2) \bigcup Cover(s_3)$, respectively. Since for the multiple-BS cases, the connected coverage of the WSN is just

[3] Note that $A \oplus B = \{u + v : u \in A, v \in B\}$ for $A, B \subset \mathbb{R}^2$.

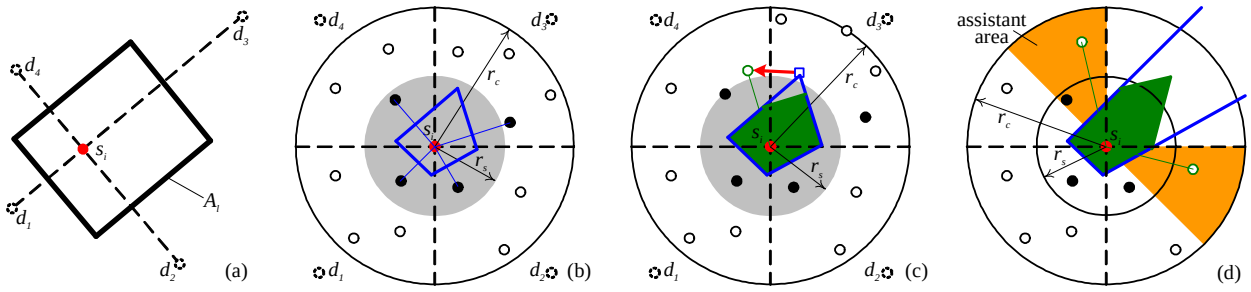


Figure 2-2. Illustration of the LVP-based boundary node detection algorithm.

the union of the connected coverage regarding each individual BS, hereafter we focus on the single-BS case for the ease of presentation.

2.3.2 Localized Voronoi Polygons

Our BOND scheme is based on two novel geometric concepts called Localized Voronoi Polygon (LVP) and Tentative LVP (TLVP) which are nontrivial generalization of Voronoi Polygons (VPs) [123] from computational geometry. We must point out that a similar concept called Localized Voronoi Diagrams (LVDs) is introduced as the dual of Localized Delaunay Triangulations (LDTs) in the literature [78, 98]. The edge complexity of LDT is analyzed in [78] and its applications in topology control and routing for wireless networks are discussed in [98]. However, there is no indication on how to relate this concept to the coverage problems in WSNs. Moreover, unlike our work there is no description on how to efficiently construct LVDs given in [78, 98]. Furthermore the idea of using TLVP to reduce the overhead of the detection algorithm in this chapter is completely new. Finally and most important, our scheme BOND only uses the local information to detect boundary instead of global information commonly used in either VP or DT.

We first define VPs, LVPs and TLVPs in terms of half planes. For two distinct points $s_i, s_j \in V$, the dominance region of s_i over s_j is defined as the set of points which are at least as close to s_i as to s_j, i.e.,

$$Dom(s_i, s_j) = \left\{ v \in \mathbb{R}^2 : \|v - s_i\| \leq \|v - s_j\| \right\}. \tag{2–6}$$

Obviously, $Dom(s_i, s_j)$ is a half plane bounded by the perpendicular bisector of s_i and s_j, which separates all points in the plane closer to s_i than those closer to s_j.

Definition 2.4. ***[VP, LVP and TLVP]*** *The **VP** associated with s_i denoted by $Vor\,(s_i)$, is the subset of the plane that lies in all the dominance regions of s_i over other points in V, namely,*

$$Vor\,(s_i) = \bigcap\nolimits_{s_j \in V - \{s_i\}} Dom\,(s_i, s_j). \tag{2–7}$$

*In the same way, the **LVP** denoted by $LVor(s_i)$, and the **TLVP** denoted by $TLVor(s_i)$, associated with s_i are defined as:*

$$LVor\,(s_i) = \bigcap\nolimits_{s_j \in Neig(s_i)} Dom\,(s_i, s_j); \tag{2–8}$$

$$TLVor\,(s_i) = \bigcap\nolimits_{s_j \in SubNeig(s_i)} Dom\,(s_i, s_j), \tag{2–9}$$

where $SubNeig\,(s_i)$ is a proper subset of $Neig\,(s_i)$, i.e., $SubNeig\,(s_i) \subset Neig\,(s_i)$.

The collection of LVPs given by

$$\mathfrak{LVor}\,(V) = \{LVor\,(s_i) : s_i \in V\} \tag{2–10}$$

is called the localized Voronoi diagram (LVD) generated by the node set V. The boundary of $LVor(s_i)$, i.e., $\partial LVor(s_i)$, may consist of line segments, half lines, or infinite lines, which are all called local Voronoi edges.

Lemma 1. *Properties of VPs, LVPs and TLVPs:*

(i) $LVor\,(s_i)$, $TLVor\,(s_i)$ and $Vor\,(s_i)$ are convex sets;

(ii) $Vor\,(s_i) \subseteq LVor\,(s_i) \subset TLVor\,(s_i)$;

(iii) Plane $\mathbb{R}^2$ is completely covered by $\mathfrak{LVor}\,(V)$.

Proof. (i) Since a half plane is a convex set and the intersection of convex sets is a convex set, a LVP (or a TLVP) as well as a VP is a convex set.

(ii) From Eqs. (2–7), (2–8) and (2–9) we have

$$Vor\,(s_i) = LVor\,(s_i) \bigcap \left(\bigcap\nolimits_{s_j \in V, s_j \notin Neig(s_j)} Dom\,(s_i, s_j) \right),$$

$$LVor\left(s_i\right) = TLVor\left(s_i\right) \bigcap \left(\bigcap_{s_j \in Neig, s_j \notin SubNeig\left(s_j\right)} Dom\left(s_i, s_j\right)\right),$$

which directly leads to Lemma 1 (ii).

(iii) It is well known in computational geometry that

$$\bigcup_{s_i \in V} Vor\left(s_i\right) = \mathbb{R}^2. \tag{2–11}$$

(cf. [123, Property V1, pp. 77]). Combining (2–11) with Lemma 1 (ii), we can directly obtain Lemma 1 (iii). □

Therefore, the set $\mathfrak{LVor}\left(V\right) \cap A$ can fully cover an arbitrary subset $A \subseteq \mathbb{R}^2$. Note that this result can be easily extended to any cluster $Clust\left(s_i\right)$ in V, that is,

$$\bigcup_{s_j \in Clust\left(s_i\right)} LVor\left(s_j\right) = \mathbb{R}^2. \tag{2–12}$$

2.3.3 LVP-Based Boundary Node Detection

In this subsection, we present our BOND scheme for each node to detect whether it is a boundary node based on its own LVP or TLVP by taking node s_i as an example.

2.3.3.1 Input

Our BOND is a distributed scheme in that we only need positions of node s_i's neighbors as the input of our algorithm. We temporally assume that there is no location error and will relax this assumption in Section 2.6.1. We need to consider two cases based on whether the information about the border of A_l, i.e., ∂A_l, is available. In the first case when ∂A_l is unavailable at node s_i, our detection scheme is based on the construction of $LVor\left(s_i\right)$ (or $TLVor\left(s_i\right)$); in the second case when ∂A_l is available, we need to exploit this information by calculating $LVor\left(s_i\right) \cap A_l$ (or $TLVor\left(s_i\right) \cap A_l$). It can be shown that $LVor\left(s_i\right) \cap A_l$ must be a finite convex polygon. Thus, the second case can be transformed into the first case by introducing dummy nodes into $Neig(s_i)$. See Figure 2-2 (a) for example, in which four dummy nodes, d_1 through d_4, are introduced such that

perpendicular bisectors between s_i and the dummy nodes generate the four border edges of ROI. Then we can calculate $LVor\,(s_i) \cap A_l$ by following the same procedure for calculating $LVor\,(s_i)$. Therefore, we will only discuss the first case in what follows.

We notice that dummy nodes cannot be directly applied to the generalized cases, i.e., the border of A_l consisting of curves. However in these cases, this just means that the information of border of A_l's border cannot be efficiently exploited, and the correctness of our scheme is not affected. There also exist two easy ways to remedy our BOND here. First, in general a curve can be approximated with straight line segments and thus the BOND is still applicable. Second, instead of checking whether the vertices of $LVor\,(s_i) \cap A_l$ are covered by $Disk(s_i, r_s)$ when A_l is a polygon, we still can correctly detect boundary nodes by checking every point on $\partial(LVor(s_i) \cap A_l)$ when A_l is not a polygon.

2.3.3.2 Algorithm

Our goal is to construct the $LVor(s_i)$ (or $TLVor(s_i)$) which is sufficient for the boundary node detection with the minimal requirement on the information about s_i's neighbors. We first divide $Disk\,(s_i, r_c)$ into four[4] quadrants. Then we construct the TLVP of s_i by using the nearest neighbors (solid nodes in Figure 2-2 (b)) in each of the four quadrants. Without lose of generality, we denote these four nearest neighbors as s_1, s_2, s_3, and s_4. The first TLVP is calculated by

$$TLVor\,(s_i) \leftarrow \bigcap_{j=1}^{4} Dom(s_i, s_j).$$

If all vertices of the TLVP is covered by $Disk\,(s_i, r_s)$, the procedure stops and this TLVP is saved. Otherwise, we need to find new neighbors which are the nearest to the uncovered vertices of the TLVP (cf. Figure 2-2 (c)), add those neighbors to $SubNeig(s_i)$,

[4] Other values will also work well.

and calculate the TLVP again:

$$TLVor\,(s_i) \leftarrow TLVor\,(s_i) \bigcap \left(\bigcap_{s_j \in SubNeig(s_i), j \neq 1,2,3,4} Dom\,(s_i, s_j)\right).$$

The new vertices of the new TLVP will be checked to see whether they are covered by $Disk\,(s_i, r_s)$. This procedure continues until all the vertices of the TLVP are covered by $Disk(s_i, r_s)$ or the LVP of s_i is calculated and saved.

Note that when ∂A_l is unavailable, $LVor(s_i)$ may be infinite, which means that it is possible that we cannot find any nodes in one or more quadrants in the first step. See Figure 2-2 (d) for an example. If a quadrant contains no neighbors, we define two sectors of angle $45°$ which are directly adjacent to the quadrant as the assistant area, and add the nodes in this area to $SubNeig(s_i)$ first. If all the nodes in the assistant area cannot make TLVP finite, we can conclude that LVP must be infinite without need to do further calculation.

2.3.3.3 Output

If $LVor\,(s_i)$ is infinite, s_i must be a boundary node. If $LVor\,(s_i)$ (or the final $TLVor\,(s_i)$) is finite with all the vertices are covered by s_i, then $s_i \in IN\,(s_i)$. Otherwise, $s_i \in BN\,(s_i)$.

2.3.4 Algorithm Validation

In the VD, the VPs of different nodes are mutually exclusive, but in the LVD, the LVPs of different nodes may overlap. This difference makes the validation of our algorithm totally different from that of existing VP-based ones.

Theorem 2.1. *If there is a point $v \in LVor\,(s_i)$ which is not covered by s_i, i.e., $v \notin Disk\,(s_i, r_s)$, there must exist a point $h \in LVor\,(s_i)$ that is not covered by any node, and s_i must be an area-coverage boundary node.*

Proof. Without loss of generality, we assume that the node nearest to s_i and outside $Disk\,(s_i, r_c)$ is s_m and $\|s_i - s_m\| = r_c + \delta$ for $\delta > 0$. Let $s`_m$ be the point on $\overline{s_i v}$ satisfying $\|s_i s`_m\| = \|s_i s_m\|$, and h be another point on $\overline{s_i v}$ such that $\|s_i h\| = r_s + \delta/2$ (Figure 2-3).

By the triangular inequality, we have $\|s_m h\| + \|s_i h\| \geq \|s_i s_m\| = \|s_i s`_m\| = \|s_i h\| + \|h s`_m\|$. Therefore, $\|s_m h\| \geq \|h s`_m\| = \|s_i s`_m\| - \|s_i h\| = r_s + \delta/2$, which means that s_m cannot cover h and nor does any other node in $Disk\,(s_i, r_c)^{\mathrm{C}}$. The reason is that, since $\|s_i s_l\| > \|s_i s_m\|$ holds for any node $s_l \in Disk\,(s_i, r_s)^{\mathrm{C}}$ and $s_l \neq s_m$, we have $\|s`_l h\| > \|s`_m h\|$ where point $s`_l$ is on the line $\overline{s_i v}$ and $\|s_i s`_l\| = \|s_i s_l\|$. Therefore, $\|s_l h\| \geq \|s`_l h\| > \|s`_m h\| = r_s + \delta/2$.

Since $v \in LVor\,(s_i)$, based on the convexity of $LVor\,(s_i)$ we have $\overline{s_i v} \in LVor\,(s_i)$. Therefore, $h \in LVor\,(s_i)$, which implies for any node $s_j \in Disk\,(s_i, r_c)$ and $s_i \neq s_j$, we have $\|s_j h\| \geq \|s_i h\| > r_s$, i.e., no nodes in $Disk\,(s_i, r_c)$ can cover h. Consequently, we can conclude that no node in the plane can cover h because $Disk\,(s_i, r_c) \cup Disk\,(s_i, r_c)^{\mathrm{C}} = \mathbb{R}^2$. Note that from the above proof process, we can see that h can be arbitrary close to $v`$, the intersection of circle $\partial Disk\,(s_i, r_s)$ and $\overline{s_i v}$. Therefore, s_i is a boundary node. □

Theorem 2.2. *If there is a point $v \in A_l$ not covered by any sensor node, for every cluster $Clust(s_i)$ there must exist at least one sensor $s_j \in Clust(s_i)$ whose $LVor\,(s_j)$ is not completely covered by $Disk\,(s_j, r_s)$.*

Proof. According to Lemma 1 (iii) or (2–12), we have

$$\bigcup\nolimits_{s_j \in Clust(s_i)} (LVor\,(s_j) \cap A_l) = A_l \tag{2–13}$$

Therefore, for any $v \in A_l$, it must lie in at least one $LVor\,(s_j) \cap A_l$ for $s_j \in Clust(s_i)$. □

Theorems 2.1 and 2.2 prove that the sufficient and necessary condition for $Clust\,(s_i)$ to completely cover A_l is that $LVor\,(s_j) \cap A_l$ is completely covered by s_j for all $s_j \in Clust\,(s_i)$. The following theorem shows that when $LVor\,(s_i)$ or $LVor\,(s_i) \cap A_l$ is finite, the coverage of vertices of $LVor\,(s_i)$ by s_i is equivalent to the coverage of the whole $LVor\,(s_i)$ by s_i, which guarantees the correctness of our LVP-based algorithm.

Theorem 2.3. *$LVor\,(s_i)$ is fully covered by s_i if and only if $LVor\,(s_i)$ is finite and all the vertices are covered by s_i.*

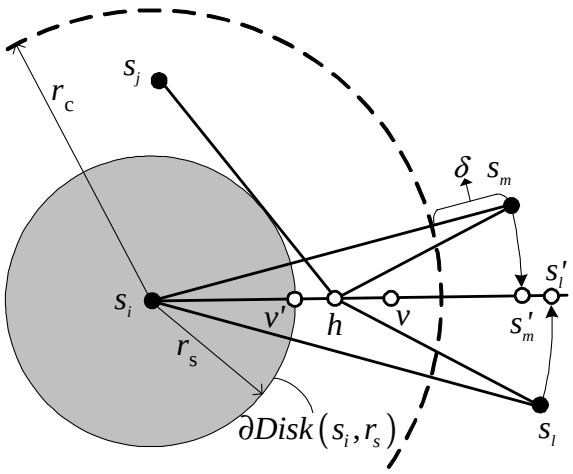


Figure 2-3. Illustration of the proof of Theorem 2.1.

Proof. Let $Ve(s_i)$ be the set of vertices of $LVor(s_i)$. Obviously, when $LVor(s_i)$ is completely covered by s_i, i.e., $LVor(s_i) \subset Disk(s_i, r_s)$, we have $v \in Disk(s_i, r_s)$ for all $v \in Ve(s_i)$ and $LVor(s_i)$ is finite. Since

$$\max_{u \in LVor(s_i)} \{\|s_i - u\|\} \leq \max_{v \in Ve(s_i)} \{\|s_i - v\|\},$$

when $v \in Disk(s_i, r_s)$ for all $v \in Ve(s_i)$, we have $u \in Disk(s_i, r_s)$ for all $u \in LVor(s_i)$. □

2.3.5 Discussions on BOND

Low Overhead. It has been shown in [123] that in general VPs cannot be computed locally. Therefore, the traditional VP-based schemes [40, 53, 157] are not distributed and are very expensive in terms of communication overhead. Our BOND scheme is a truly localized polygon-based solution because computing $LVor(s_i)$ (or $TLVor(s_i)$) only needs one-hop information (this can be directly obtained from the Eqs. 2–8 and 2–9). Assuming that the number of neighbors is k, each node can compute its own $LVor(s_i)$ with complexity smaller than $O(k)$. In addition, the computation of the $LVor(s_i)$ only involves some simple operations on polygons which can be efficiently implemented (e.g., PolyBoolean library [97]). We further simplify the detection process by constructing TLVPs first. For a densely deployed WSN, we have $LVor(s_i)$ or $TLVor(s_i) \rightarrow Vor(s_i)$, and it is well known in computational geometry that

under the homogeneous spatial Poisson point process, the average number of vertices of $Vor(s_i)$ is 6 [123]. Therefore, when the node density is high, BOND on average only needs 4 to 6 nearest neighbors' information to successfully detect the boundary nodes. Moreover, when a neighbor node dies, BOND needs do nothing unless the dead node is used to construct the final $TLVor(s_i)$ or $LVor(s_i)$ in the last turn of LVP or TLVP construction. This unique property will greatly simplify the update of detection results and save precious energy of each sensor node. All these advantages cannot be achieved by other localized boundary node detection schemes in the literature, such as the perimeter-coverage checking approach [69] and the crossing-coverage checking approach [62, 174]. We refer to Section 2.5.1 for a detailed comparison.

Other Applications. We are aware of the following applications of BOND or its basic operations, which are by no means a complete list.

- Topology control and routing. It has been shown in [78, 98] that the dual of LVP, called LDT, can be used to design distributed topology control and routing protocols for wireless ad hoc networks with energy efficiency and the guarantee of the delivery. From the property of duality, we can directly obtain LVPs if LDTs are determined, or vise versa.

- Spatial aggregation. In distributed data processing for WSNs, to reduce the sampling errors in the aggregated spatial data, it is proposed in [138] to first calculate the VP of each sensor. As mentioned before, since the VP cannot be computed locally, the LVP can be used as a good approximation of the VP in spatial aggregation.

- Coverage-preserving node sleeping scheduling. Since sensor nodes are usually deployed with redundancy, it is possible to prolong the network lifetime while preserving the connected coverage by scheduling some nodes into the sleeping state [153]. Each node can locally decide whether its own LVP is covered by its neighbors. If this is the case, a node declares itself eligible for sleeping, announces this fact to its neighbors, and then goes to sleep. To avoid the forming of coverage hole caused by eligible nodes switching into sleep simultaneously, a randomly delayed announcement scheme using common random backoff approaches is proposed in [153].

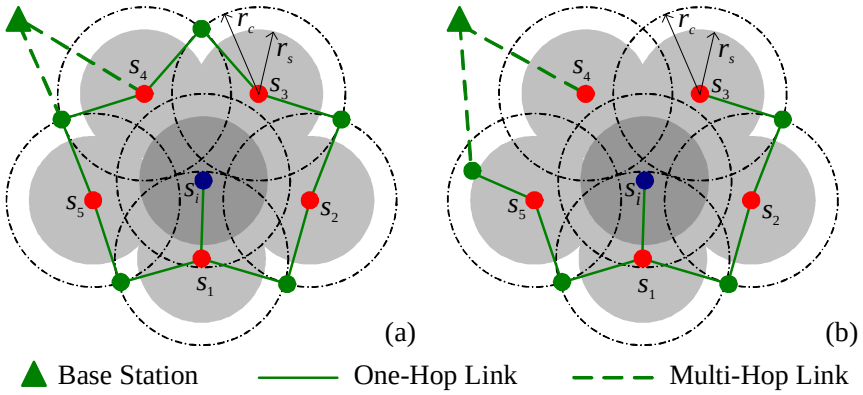


Figure 2-4. Non-locality of the boundary node detection when $r_c < 2r_s$.

2.3.6 Locality of Boundary Node Detection

In this subsection we investigate further to show that it is impossible to find local algorithms for boundary node detection with arbitrary node distributions when $r_c/r_s < 2$. We first define what we mean by local algorithms. This definition is based on a model proposed in [128].

Definition 2.5. ***[Local Algorithm]*** *Assume that each computation step takes one unit of time and so does every message to get from one node to its directly connected neighbors. With this model, an algorithm is called* ***local*** *if its computation time is $O(1)$, in terms of the number of nodes n in the system.*

Our BOND shows that when $r_c = 2r_s$, sensors can locally determine if it is a boundary node. When $r_c > 2r_s$, since the node will have more information about other nodes around itself, it can still locally detect whether it is a boundary node. However, in the case of $r_c < 2r_s$, individual nodes can neither locally say "yes" nor "no" to the question of whether a given node is a boundary node. To see this, consider sensors deployed as in Figure 2-4. Obviously, node s_i is an interior node. However, to confirm this, it needs to know the existence of nodes s_1 to s_5 with the help of some relay nodes (green nodes). In Figure 2-4 (a), node s_4 is already 5 hops away from node s_i. In fact the distance between these two nodes can be arbitrary long, which is shown in Figure 2-4 (b). Therefore, for arbitrary node distributions, it is impossible to find a localized

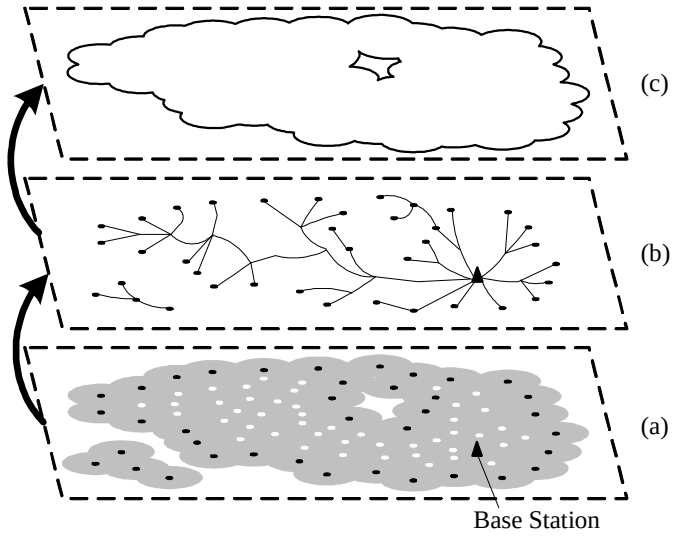


Figure 2-5. Basic operations of the BOND-based CIP. (a) Each sensor executes BOND individually. (b) Boundary nodes (black dots in (a)) report themselves to the base station. (c) The BS reconstructs the coverage boundary. Note that the shaded area in (a) represents the coverage of sensors, and that the shaded area at the left bottom corner in (a) is lost in (c) because it is not the connected coverage.

boundary node detection algorithm that always works. In [10], the authors considered general values of r_c/r_s with regular deployment patterns such as the hexagon, square grid, rhombus, and equilateral triangle. Unlike [10], in this chapter, we prefer to make the strict assumption on the value of r_c/r_s rather than on the node distribution pattern. The reason is that even in some scenarios, sensor nodes are originally distributed with regular patterns (e.g., hexagon, square grid, etc.), due to node failures, node movements caused by animals or winds, etc., the topology of the WSN will become irregular sooner or later. In contrast, even in the case of $r_c/r_s < 2$, we can still assume a smaller value of r_s whereby to get a conservative inference of the coverage, which is desirable for some security-critical applications.

2.4 CIP: Coverage Inference Protocol

In this section, we describe how to use BOND to build a practical coverage inference protocol (CIP).

Our design philosophy is that, since the minimum information required to describe the coverage is the positions of boundary nodes (cf. Section 2.3.1), we just need to detect boundary nodes. In other words, our scheme can ensure that, for the BS to reconstruct the "coverage image" without any distortion, the information transmitted from sensors to the BS is minimized. Also note that our BOND only involves local message exchanges. In a large-scale WSN, the overhead from local broadcast is very small as compared to that from the end-to-end communications from sensor nodes to the BS. Therefore, our approach can save the precious energy of sensor nodes. Figure 2-5 illustrates the basic operations of our BOND-based CIP, which consist of the following three steps:

2.4.1 Neighborhood Monitoring and Self-Detection

After the deployment of the WSN, we assume localization techniques are available for sensor nodes to decide their positions. Each node then collects the position information of its neighbors by broadcasting its own positions, and executes BOND to detect whether it is a boundary node. If so, it will report its position to the BS. We refer to the neighbors used to construct the LVP or TLVP in the last run of BOND as its consulting neighbors.

In our protocol, both interior and boundary nodes are required to broadcast an Existence Updating Packet (EUP) to their neighbors for a random period of time exponentially distributed with rate T_{EUP}. In addition, each interior node, say s_i, maintains a timer $C_0(j)$ of expiry value much larger than T_{EUP} for each of its non-consulting neighbors, say s_j. If s_i does not overhear any packet (either an EUP or data packet) from s_j before $C_0(j)$ expires, it will treat s_j as a dead neighbor, which can become alive if s_i overhears any packet from it later. Node s_i also maintains two timers for each of its consulting neighbors, say s_k: the neighbor-monitoring timer $C_1(k)$ and the neighbor query timer $C_2(k)$. If s_i does not overhear any packet from s_k before $C_1(k)$ expires, it unicasts a Neighbor Query Packet (NQP) to s_k and starts $C_2(k)$. If still alive, s_k is

required to send back an EUP immediately and wait for an ACK from s_i. If node s_i still does not overhear any packet from s_k before $C_2(k)$ expires, s_i will treat s_k as a dead neighbor and re-execute BOND with alive neighbors as input. In general, the expiry values of $C_0(j)$ and $C_1(k)$ should be in the same order of T_{EUP}, in order to guarantee that with high probability, each node will receive EUPs from all alive nodes in its neighborhood. The expiry values of $C_2(k)$ should be much smaller than T_{EUP}, because we require that the node which receives the NQP needs to send back an EUP immediately.

Unlike previous neighbor-monitoring schemes employing a single timer [137] or treating neighbors as the same [68], our scheme sets different timers for non-consulting and consulting neighbors. The major reason for doing so is that data packets and EUP-like broadcast packets are subject to loss due to wireless transmission errors or collisions. As a result, a node may falsely identify an alive neighbor as a dead one. Obviously, for non-consulting neighbors, we can decrease the false positive rate by setting a larger timer value. However, using a larger timer value for consulting neighbors will increase the response delay, i.e., the delay from when coverage holes emerge to when they are detected by boundary nodes. Therefore, we use two timers for consulting neighbors to ensure both a shorter response delay and a lower false positive rate: although the expiry value of $C_1(k)$ is small, we actively query the questionable neighbor before we treat it as the dead neighbor, which may significantly increase the accuracy of our scheme. As mentioned before, if the node distribution follows SPPP, each node only has 4 to 6 consulting neighbors on average, which means the high feasibility of our two-timer scheme. Therefore, by adopting one-timer (C_0) scheme for non-consulting neighbors and two-timer (C_1 and C_2) scheme for consulting neighbors, our design achieves a better balance among accuracy, delay and communication overhead. Note that this benefit stems from the fact that our BOND divides each node's neighbors in to two categories. For other boundary-node-based approaches in the literature (cf.

Section 2.5.1) where there is no division in neighbors, either one-timer or two-timer scheme should be adopted for all neighbors and the balance between communication overhead and the accuracy cannot be handled in this way.

Note that the communication overhead of our neighbor-monitoring scheme can be further reduced. The EUP packets can piggyback onto regular local broadcast packets used to learn link conditions, maintain the routing information, and facilitate other network operations. In addition, in the presence of data traffic, any packet overheard by a node should be regarded as an EUP packet, and any data packet sent from a node can cancel the next EUP packet it should broadcast.

2.4.2 Self-Reporting of Boundary Nodes

Whenever identifying itself as a boundary node, a sensor node should send its position information to the BS which can reconstruct the "image of the coverage" based on all the received position information of boundary nodes.

2.4.3 Explicit ACKs from the BS

Since the packet loss ratio due to collisions or noise is pretty high in the WSN [79], boundary nodes need some mechanisms to ensure that their reports have been received by the BS. Otherwise, they have to repeatedly resend their reports, which causes energy waste. The issue of reliable sensor-to-BS communication, thus far, has not been addressed thoroughly in WSN research community. The work on reliable communication in WSNs first appears in [155], and then in [149]. However, no guaranteed reliability semantics are provided in these work. In [136], the authors firstly propose the notion called "event-to-sink" reliability, and study the reliable communication from this perspective. Their solution is mainly based on adjusting the reporting frequency of source nodes, and is applicable for monitoring a continuously changing event or reporting a huge amount of data. For our case, each sensor node will be on the boundary or not for a relatively long period, and each boundary node only needs to report its location information to the BS. Therefore, the scheme proposed in [136] cannot

be applied here. In [127], the authors study the "BS-to-sinks reliability", and argue that the requirement and implementation of reliability in a WSN is firmly dependent upon the specific application, and there is no one-for-all solution. For our problem seating, an intuitive solution is to require the BS to send individual ACK to each boundary node from where the report has been received at the BS. In what follows, we will show that this intuitive solution can be improved by introducing the Bloom filter [15]. For the sake of clarity, we start with description of ACK format with the Bloom filter. Then we analyze its performance quantitatively.

The basic idea here is to use Bloom filters to design an energy-efficient approach for the BS to broadcast only one ACK to acknowledge multiple boundary nodes. Let $s_1, s_2, \cdots, s_a$ be the a boundary nodes which the BS wants to acknowledge explicitly. Let $h_1, h_2, \cdots, h_b$ be the b hash functions of the Bloom filter, each with range $\{1, 2, \cdots, l\}$. Let $ack(t) = (b_1, b_2, \cdots, b_l)$ be a bit vector of length l. $ack(t)$ is the t-th Bloom filter used to indicate boundary nodes the BS wants to acknowledge, and it is initially set to $(0, 0, \cdots, 0)$. For $\forall i$, $1 \leq i \leq b$ and $\forall j$, $1 \leq j \leq a$, the BS computes $h_{ij} = h_i(ID_{s_j})$ where ID_{s_j} indicates the unique ID of node s_j, and sets $b_{h_{ij}} = 1$. The BS can then use the Bloom filter $ack(t)$ as ACK and send it back using efficient broadcast or geographic multicast protocols for WSNs. When a boundary node s_j receives the $ack(t)$, it performs a membership test: it computes $h_i(ID_{s_j})$ for $\forall i$, $1 \leq i \leq b$; if all of these positions are 1 in the $ack(t)$, then boundary node s_j knows its report has been received and acknowledged by the BS.

Note that the Bloom filter may induce a small number of false positives, i.e., a few un-acknowledged boundary nodes may pass the membership test and therefore believe that their reports have been received by the BS. On the other hand, Bloom filters ensure that there are no false negatives, i.e., all acknowledged nodes are guaranteed to pass the membership test. In practice we can tune our b and l parameters to enable tradeoff between communication and computational overhead and false positive rate.

Now we analyze the benefit of utilizing ACKs with Bloom filters. First of all, we need quantify the communication overhead of individual-ACK scheme. Suppose there are n sensor nodes in the network, and the length of each node ID is $\log n$ bits. If there are a boundary nodes which the BS wants to acknowledge explicitly, the BS need send out a ACK packets, and each packet with the size of $O(\log n)$ bits. This is because the header of the ACK packet should include the destination node ID with $\log n$ bits. Next, we consider the multi-hop communication scheme that can be used to support the BS-to-sensor communication. Note that each sensor node only has a finite buffer, and the buffer size is relatively small compared to n for a large-scale WSN. Therefore, it is almost impossible to provide unicast communication from the BS to individual sensor node. Note that to support the unicast communication from the individual sensor to the BS is still possible (cf. Section 2.5.2.2). This is caused by the asymmetry of traffic pattern in WSNs [79]. For the sensor-to-BS unicast communication, we can construct a routing tree rooted at the BS (refer to Figure 2-11 for an example), and each node only needs to relay the packet from its child nodes to its parent node in this routing tree. To maintain this routing tree, each node only needs to add one entry into its routing table with the destination as the BS and the next-hop as its one-hop parent node. For the BS-to-sensor unicast communication, the situation will be totally different. For every child node in the subtree rooted at a particular node, this node needs to add one entry in its routing table for that child node. For the node near the BS, it may need $O(n)$ entries in the routing table, which is too large for the buffer-constraint sensors. So, in practice, there are no efficient scheme to support unicast downstream communication (from the BS to sensors) [79]. Especially for the small size packet, to establish a tentative route by broadcast a route discovery packet is invaluable. Therefore, even for the BS-to-sensor unicast communication, when the packet size is small, we still use global broadcast from the BS.

In individual-ACK scheme, the BS needs to perform a broadcasts, each with packet size $O(\log n)$ bits. For Bloom-filter-based scheme, the BS needs to perform one broadcast with packet size $O(l)$ bits. For the standard Bloom filter [17], in order to keep a low false positive rate, l is on the order of n. For example, in our simulation, $l = 9n$ bits, $b = 6$ and the false positive rate is 1.33%. Therefore, when $a = \Omega(n/\log n)$, it is beneficial to use Bloom-filter-based scheme. This is equivalent to the situation when the percentage of boundary nodes is $\Omega(1/\log n)$. For a large-scale WSN, $1/\log n$ is a very small number, which means the Bloom-filter-based ACk should be used in practice with high probability.

To sum up, in this section, we design some complementary components to BOND, to complete our BOND-based CIP protocol. We emphasize that these designs are intuitive and straightforward, because simplicity, i.e., minimal communication and computational overhead, is our principal design objective. However, the advantages of adopting these simple designs stem from the BOND algorithm. Specifically, BOND needs to check only minimal number of consulting neighbors, which enables us to utilize two-timer scheme without introducing too much communication overhead. Moreover, BOND is able to identify boundary nodes, which facilitates us to employ ACK-based scheme to provide reliable sensor-to-BS communications.

2.5 Comparison and Simulation

To the best of our knowledge, there is no other coverage inference protocol developed for WSNs which is as complete as our CIP. We, however, note that many interesting ideas proposed by researchers for WSN coverage and self-monitoring may be adapted to infer WSN coverage. In this section, we exploit these possible alternatives and compare them with our CIP via both theoretical analysis and simulations.

The most important metric used in the comparison is the energy consumption incurred by different coverage inference protocols. In addition, we assume that packet transmissions are subject to noise or collision. Under this assumption, we further

define two other performance metrics. The first one is the false alarm probability, defined as the probability that a non-boundary sensor node is falsely diagnosed as a boundary node. This may happen if consecutive EUPs from a sensor node get lost due to collision and/or noise. The other metric is the response delay, defined as the time from when a sensor node becomes a boundary node to when this event is locally detected. Apparently, the later two metrics are conflicting in essence for a timeout-based coverage inference protocol like our CIP. In particular, to achieve a smaller false alarm probability (i.e., more accurate boundary detection) desires a larger timeout value which, in turn, would result in a longer response delay, and vice versa. Therefore, we would compare the false alarm probabilities of different coverage inference protocols for a given response delay, or equivalently compare their response delays for a given false alarm probability.

The evaluation philosophy and simulation settings are given in Appendix-A. In order to mimic independent and random node failures, we simulate multiple sensor networks by varying the number k of neighbors for each node (or equivalently, the node density λ) through NS2 network simulator [113]. To facilitate the presentation, we also classify the possible coverage inference protocols into two categories, namely, boundary-node-based approaches and aggregation-based approaches, based on how the coverage information is gathered and reported to the BS.

2.5.1 Boundary-Node-Based Approaches

Approaches in this category are to find boundary nodes first and then transmit such information to the BS in a way similar to what was described in Section 2.4. In what follows, we further classify these approaches according to the boundary node identification methods they adopted.

2.5.1.1 Polygon-based schemes

In [40, 53, 157], Voronoi diagram (VD) is used for boundary node detection. Briefly speaking, the VD of a node set V, is the partition of the Euclidean space into polygons,

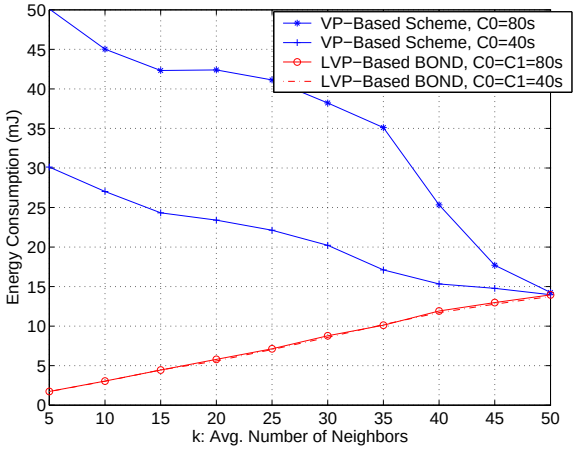


Figure 2-6. Energy consumption for the LVP- and VP- based schemes.

called Voronoi polygons (VPs) and denoted by $Vor(s_i)$ for $s_i \in V$ such that all the points in $Vor(s_i)$ are closer to s_i than to any other node in V. According to the closeness property of VPs, if some portion of a VP is not covered by nodes inside the VP, it will not be covered by any other node, which implies a coverage hole. Therefore, it is claimed in [40, 53, 157] that each node can locally check whether it is on the coverage boundary under the assumption that VPs can be derived locally. However, it has been shown that in general VPs cannot be locally computed [169]. Intuitively, our LVP-based BOND will have smaller communication overhead or equivalently, smaller energy consumption than the VP-based schemes since LVPs can be locally computed. In what follows, we prove this intuition in a formal way.

Theorem 2.4. *If there exist boundary nodes, the costs of the NEP-based and LVP-based algorithms are always smaller than the cost of the VP-based ones.*

The proof of the theorem depends on the following lemma:

Lemma 2. For any $s_i \in V$, the VP $Vor\,(s_i)$ can be locally computed if and only if $Clust\,(s_i)$ can completely cover the plane $\mathbb{R}^2$ (or A_I, when the information of ∂A_I is available), i.e., $Cover\,(s_i) = \mathbb{R}^2$ (or $Cover\,(s_i) \cap A_I = A_I$).

Proof. Let $d = \max \|v - s_i\|$ for any $v \in Vor(s_i)$. Since $Vor(s_i)$ is a convex set, then $d = \infty$ if $Vor(s_i)$ is infinite, otherwise d is the distance from a vertex of $Vor(s_i)$ to s_i.

$Vor(s_i)$ can also be computed in a similar way as LVP with set V as input. Specifically, we first compute $LVor(s_i)$ as the tentative VP of s_i and then refine the tentative VP iteratively. In each iteration, we add one-more-hop information about node positions. We can determine that the construction of $Vor(s_i)$ is completed when all the nodes in $Disk(s_i, 2d)$ have been counted. Therefore, $Vor(s_i)$ can be locally computed, which implies that $2d \leq r_c$ or $d \leq r_s$ and thus guarantees the complete coverage of $Vor(s_i)$. Since this holds for all $s_i \in V$, we can ensure the complete coverage of the plane.

From Theorems 2.1 and 2.2, a node set V can completely cover $\mathbb{R}^2$ if and only if $LVor(s_i)$ is fully covered by $Disk(s_i, r_s)$ for any $s_i \in V$. From Lemma 1, this implies that $Vor(s_i) = LVor(s_i)$ for any $s_i \in V$. Therefore, $Vor(s_i)$ can be locally computed by s_i just as $LVor(s_i)$. □

Therefore, when there are boundary nodes, it is impossible to compute all $Vor(s_i)$'s locally based on only one-hop information. Since multi-hop communications are unavoidable, the energy consumption of the VP-based approaches will be higher than our LVP-based BOND. Only when the node density is so high that the ROI is completely covered (not considering the ROI border), is the cost of the VP-based approaches equal to that of ours. However, in this case, there is no need for coverage boundary detection at all. So Theorem 2.4 guarantees that when boundary detection algorithms are helpful, the cost of our BOND is definitely smaller than the VP-based ones.

We answer the question of how significant the cost savings are by simulations. To focus on comparing the energy consumption of two schemes, we set $T_{EUP} = 1s$, and $C_0 = C_1 = 40s$ and $80s$, and $C_2 = 0$ for our BOND. For a fair comparison, the VP-based scheme uses the same timer C_0 for one-round boundary node detection. We set a very large timeout value to make sure that false alarm probability is very small and thus can be neglected here. The packet size of EUP is 64 bytes (no NQP here), and any control packet transmitted for the VP-based scheme to reconstruct VPs is of 64 bytes as well.

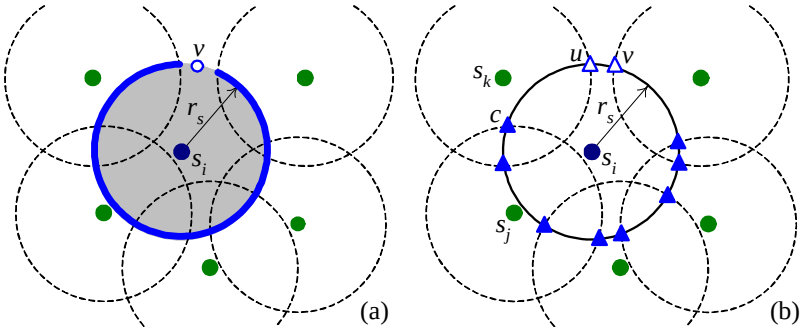


Figure 2-7. Perimeter-based boundary node detection approaches. (a) Perimeter-coverage checking approach proposed in [69]. The solid curve represents the portion of perimeter of sensing disk covered by neighbor nodes. (b) Crossing-coverage checking approach proposed in [62, 174]. Solid and open triangles represent covered and uncovered crossings, respectively.

Figure 2-6 shows the average node energy consumption for the VP-based and the LVP-based schemes as a function of k. We can see that energy consumption of our LVP-based BOND only slightly increase as k becomes larger, as the reception energy consumption becomes larger with the increasing number of neighbors. In addition, the energy savings of our LVP-based BOND are quite significant compared to the VP-based scheme. We can also observe that the difference between two schemes becomes smaller with the increase of k. The reason is that the number of hops needed to reach $2d$ for VP computing will become smaller with the increase of k. In particular, when $k = 50$ where the ROI is fully covered with probability 0.9999, VPs can be locally calculated as LVPs, so there will be no difference between these two schemes. Therefore, compared to the VP-based scheme, our BOND-based CIP can achieve remarkable energy savings when the functional nodes are sparsely deployed ($4.5 < k < 4\log n + 4\log\log n$). Also notice that a small C_1 has no effect on our BOND-based CIP because $C_1 = 40s$ is still enough for one-hop information gathering (note that $k < 50$). However, a small C_0 has a great effect on the VP-based scheme, because it significantly reduces the multi-hop communication, and thus only allows a node to calculate a tentative VP. This observation shows that calculating the final VP is not

necessary for most situations we are interested in, and LVP itself is enough for boundary node detection.

2.5.1.2 Perimeter-based schemes

The first realistic localized boundary node detection algorithm is proposed in [69], which is based on the information about the coverage of the perimeter of each node's sensing disk. It can be shown that node s_i is a boundary node if and only if there exists at least one point $v \in \partial Disk(s_i, r_s)$ which is not covered by any $s_j \in Neig(s_i)$ (cf. Figure 2-7 (a)). Based on this criterion, an algorithm with the complexity $O(k \log k)$ is designed in [69] to locally check whether one node is a boundary node, where k is the number of neighbors. A crossing-coverage checking approach proposed in [62, 174] simplifies the previous perimeter-coverage checking approach by just checking some special points called crossings on the perimeter. A crossing is defined as an intersection point of two perimeters of sensing disks. A node s_i is a boundary node if and only if there exists at least one crossing $v \in \partial Disk(s_i, r_s) \cap \partial Disk(s_j, r_s)$ which is not covered by any other $s_k \in Neig(s_i) - \{s_j\}$. Figure 2-7 (b) shows an example where c is a crossing determined by two perimeters $\partial Disk(s_i, r_s)$ and $\partial Disk(s_j, r_s)$, which is covered by the third sensing disk of node s_k. For simplicity, below we will just compare our BOND-based CIP with the crossing-coverage checking approach (denoted by CROSS).

Our BOND and CROSS can both provide truly localized boundary node detection with operational difference in the neighborhood monitoring phase. In particular, CROSS requires each node to check the positions and status of all it neighbors, which is quite inefficient when sensor nodes are densely deployed. This situation becomes worse every time when a node dies, as all its neighbors need recheck the coverage of their perimeters or crossings. In contrast, our BOND-based CIP only have consulting neighbors perform boundary node detection. When sensor nodes are densely deployed, from Lemma 1, we have $LVor(s_i)$ or $TLVor(s_i) \rightarrow Vor(s_i)$, and it is well known in computational geometry that under the homogeneous SPPP, the average number of

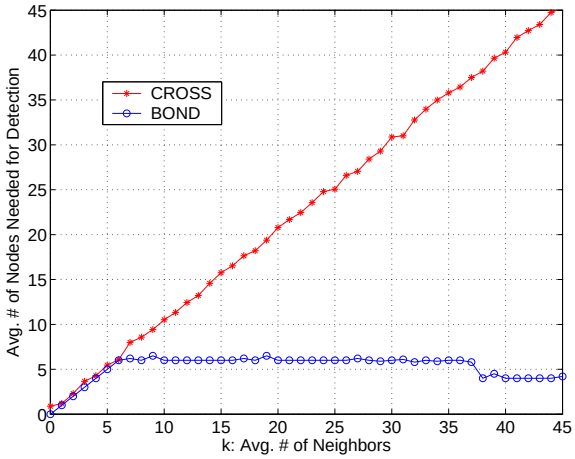


Figure 2-8. Average number of neighbor nodes needed for the crossing-coverage checking approach and our BOND.

vertices of $Vor(s_i)$ is 6 [123]. Therefore, when the node density is high, each node on average only has 4 to 6 consulting neighbors. Figure 2-8 shows the average number of neighbors needed for the CROSS and our BOND to detect boundary nodes as a function of k. It is of no surprise to see that when the node density increases, the number of nodes needed remains constant for BOND-based CIP while increasing dramatically for CROSS. This means that, in contrast to our BOND-based CIP, CROSS will incur a significant overhead at the initial stage of WSNs where sensor nodes are normally densely deployed to provide adequate redundancy and fault-tolerance.

Now we compare BOND-based CIP with CROSS regarding their tradeoffs among the updating time interval T_{EUP}, response delay and false alarm probability (denoted by $\Pr[FA]$) via simulations. Since CROSS needs the information of all neighbors, it can only use the same timer C_0 for all neighbors. Therefore, the response delay of BOND-based CIP and CROSS can be measured by $C_1 + C_2$ and C_0, respectively. The packet sizes of EUP and NQP are both 64 bytes. Our objective here is to select optimal system parameters to meet a certain performance requirement on either the response delay or $\Pr[FA]$.

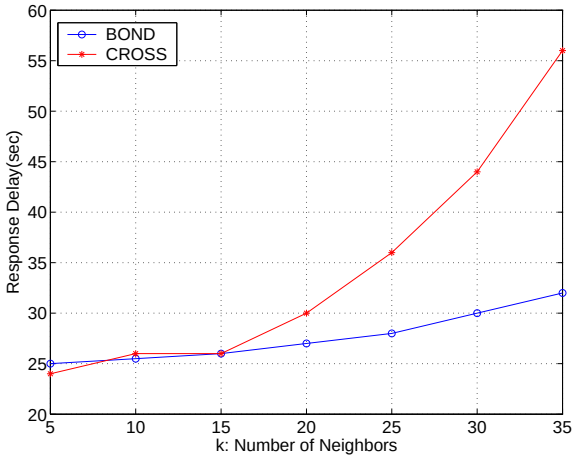


Figure 2-9. Simulation results with $T_{EUP} = 10s$, $Pr[FA] \leq 0.01$.

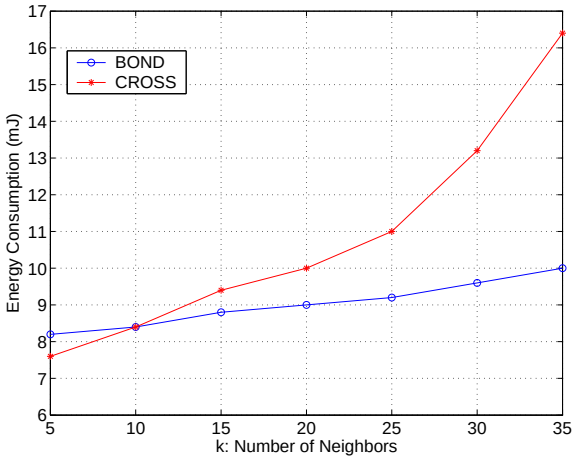


Figure 2-10. Simulation results with response delay $\leq 40s$, $Pr[FA] \leq 0.01$.

We first consider the smallest response delay can be achieved for different schemes when $T_{EUP} = 10s$ and $Pr[FA] \leq 0.01$. Simulation results in Figure 2-9 show that, when the average number of neighbors (k) increases, the response delay for CROSS dramatically increases due to more packet collisions in the shared wireless channel, but only slightly increases in our BOND-based CIP.

Figure 2-10 shows the power consumption for different schemes when $Pr[FA] \leq 0.01$ and the response delay ($C_1 + C_2$ or C_0) is bounded by $40s$. It can be seen that, in order to keep $Pr[FA] \leq 0.01$, CROSS has to adopt a small value of T_{EUP}, which would dramatically increase the power consumption. The energy consumption of our

BOND-based CIP slightly increases with k because the number of consulting neighbors will not increase with k and the related NQPs will be used with very small probability.

From Figure 2-9 and Figure 2-10, we can also observe that when functional nodes are sparsely deployed ($k \leq 15$), CROSS almost has the same performance as our BOND-based CIP. To sum up, the VP-based approaches only perform well when functional nodes are densely deployed; the perimeter-based approaches only work well when functional nodes are sparsely deployed; and only our BOND works equally well in both cases.

2.5.2 Aggregation-Based Approaches

Aggregation-based approaches are quite different from our BOND in that each node actively communicate with the BS no matter whether it is a boundary node or not. Therefore we need to compare those schemes to our BOND-based CIP as a whole in this subsection. In what follows, we classify these approaches into naive ones and spatial ones.

2.5.2.1 Naive schemes

The most straightforward scheme requires that each node periodically send EUPs ("keep-alive" messages) to inform the BS about its existence so that the BS can always learn the connected coverage of the WSN. If the BS does not receive the update information from a particular node in a pre-determined time interval, it can infer that this node is dead. This approach obviously suffers from significant communication overhead and thus energy consumption.

An alternative to the naive scheme is to let each node report the positions of its dead neighbors, in which nodes can locally cooperate to ensure that each dead node is reported once. However, this scheme is unlikely to be the optimal one because the death of some nodes does not necessarily imply the change of connected coverage especially when the WSN is densely deployed. In addition, there might be redundant

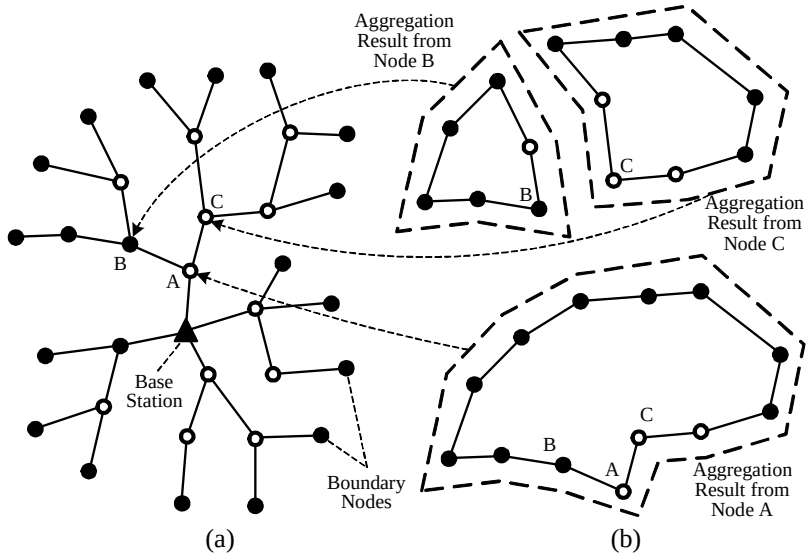


Figure 2-11. Illustration of SAB. (a) The routing tree for the spatial aggregation is constructed with the root as the BS. (b) Node *A* aggregates coverage information by combining polygons from its children *B* and *C*. Note that only the solid dots are real boundary nodes. If the open dot appears in the aggregation results, it represents redundant information and leads to energy waste.

information transmitted to the BS, which means that this scheme is also not energy efficient.

2.5.2.2 Spatial aggregation-based schemes (SAB)

Since the coverage information is highly spatially-correlated, a natural way to improve the energy efficiency of the above naive scheme or its alternative is to perform spatial aggregation at intermediate nodes to reduce redundant transmissions.

Almost all techniques for spatial aggregation require the construction of a routing tree for propagating data from source nodes to the BS [91]. Once the routing tree is established, each node utilizes the routing tree to find a path to the BS. A simple method for constructing the routing tree is as follows. The BS broadcasts an initialization message into the network, which contains hop count specifying the distance from the BS. All nodes that hear the initialization message will select the BS as their parent, increase the hop count by one, and then rebroadcast the message. The message

will propagate down the network until every node has established a sender as its parent node leading toward the BS (refer to Figure 2-11 (a) for an example). The coverage information in each node is represented by a polygon. Let V_{child} denote itself, its children, its children's children, $\cdots$ This polygon covers all nodes in V_{child}, and its vertices are the boundary nodes of V_{child}. To obtain this polygon, each node only needs to receive the polygons of its children and aggregates them into a new one which is sent along the routing tree towards the BS. Some boundary nodes reported from the children may be deleted in this step because the parent nodes have more information about the coverage and thus may find that those nodes are not real boundary nodes. Please refer to Figure 2-11 for an illustration of this process.

This kind of approach is used by e-Scan [179], which focuses on monitoring the residual energy of sensor nodes, and isobars [64], which focuses on advanced aggregation techniques for highly spatially-correlated sensing data. Basically speaking, the energy efficiency of SAB depends on the time interval between two successive active reports of individual nodes. In practice, it is difficult to adaptively tune this parameter to achieve the best tradeoff between information freshness and energy efficiency.

It is possible to adapt SAB to a more passive scheme. In particular, only when a node finds that its aggregation polygon changed, does it update the polygon to its parent. However, there is still room to further improve the energy efficiency. Note that, to reconstruct the connected coverage image at the BS, the only useful information is the positions of boundary nodes. As shown in Figure 2-11(b), in this layered aggregation scheme, aggregation results from the $(i+1)$-hop node (e.g., B or C) will contain more redundant information (non-boundary nodes) than those of the i-hop node (e.g., A), and only the polygon aggregated at the BS is redundant-free. In contrast, our CIP directly gets the real boundary nodes from the first step instead of removing the redundancy layer by layer.

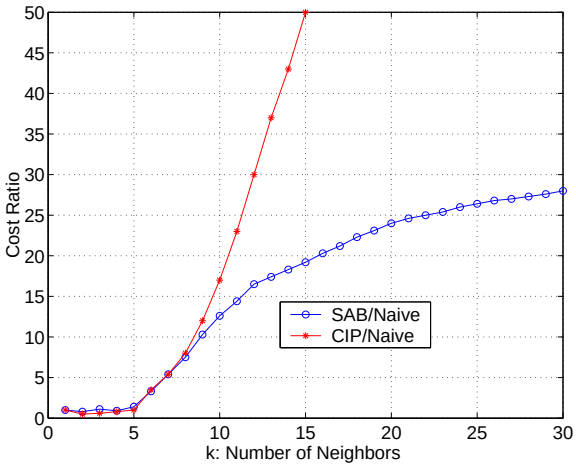


Figure 2-12. Energy cost ratio of SAB and CIP to naive scheme.

We compare the naive scheme in which periodic "keep-alive" messages need be sent, SAB, and CIP under the homogeneous SPPP model through simulations. We set $T_{EUP} = 1s$ and $C_0 = C_1 = 40s$ or $80s$, and $C_2 = 0$ for our BOND-based CIP. Since C_1 is much greater than T_{EUP}, the false alarm probability is very small and can be neglected here. We also assume that the packet size of EUP is 64 bytes (no NQP here), and that it requires 32 bits to represent a position. Therefore, in SAB, when a child sends a packet about its polygon to his parent node, and this polygon has m vertices, the packet size will be $64 + 4(m - 1)$ bytes. For a fair comparison, all the schemes use the same routing tree as shown in Figure 2-11. Let R denote the energy cost ratio which is defined as the ratio of the energy consumption of SAB or BOND-based CIP to that of the naive scheme. Figure 2-12 plots R versus different node density $\lambda = k/\pi r_c^2$, where k is the average number of neighbors per node. As we can see, when $k \leq 4$, almost every node connected to the BS is a boundary node, and the naive scheme is the best ($R < 1$ for both BOND-based CIP and SAB). However, this is the situation the WSN tries to avoid, under which the WSN has already collapsed. In addition, when $k \geq 5$, the R of BOND-based CIP grows exponentially, which shows the significant energy savings of our scheme over both the naive scheme and SAB. Also note that when the ROI is almost fully covered, the cost of local broadcasts is negligible as

compared to end-to-end communications from sensor nodes to the BS. Therefore, it is of no surprise to see that $R \to \infty$ for BOND-based CIP when $k \geq 25$. Recall that although BOND-based CIP has better performance compared to SAB, it is based on the assumption that $r_c \geq 2r_s$. If this assumption does not hold, boundary nodes cannot be locally identified (as we proved in Section 2.3.6), and all advantages will be lost. However, for SAB, it can be applied to all values of r_c/r_s, which gives another motivation for us to present SAB here.

2.6 Extensions to CIP

In this section, we extend our BOND-based CIP by considering location errors and energy depletion, which shows the flexibility of BOND and its ability to deal with some practical considerations

2.6.1 Location-Error-Tolerant CIP

So far we have assumed that each node knows its accurate location. Our CIP can also be extended to tolerate bounded location errors. In this subsection, we assume that the location error (defined as the distance between the actual location of a node and its estimated location) is upper-bounded by δ. We then have the following theorem.

Theorem 2.5. *If the location error is upper-bounded by δ, and a given node, e.g., s_i, is an interior node when all nodes are at their estimated locations and each node uses a sensing range $r_s - \delta$, node s_i must be an interior node when all nodes are at their actual locations and each node uses the actual sensing range r_s .*

Proof. We prove by contradiction. Let a and a‘ represent the actual and estimated locations of point a, respectively. Suppose that node s_i is a boundary node with the actual sensing range r_s when there is no location error. There must exist a point p in $LVor(s_i)$ that is not covered by $Disk(s_i, r_s)$. On the other hand, p must be covered by $Disk(s_i‘, r_s - \delta)$ when p is still in $LVor(s_i‘)$ or p is in $LVor(s_j‘)$ for $s_j‘ \in Neig(s_i‘)$. Note that here $LVor(s_i‘)$ and $LVor(s_j‘)$ are calculated with estimated locations and the sensing range $r_s - \delta$. We therefore have $\|s_i‘ - p\| \leq r_s - \delta$ or $\|s_j‘ - p\| < \|s_i‘ - p\|$ for

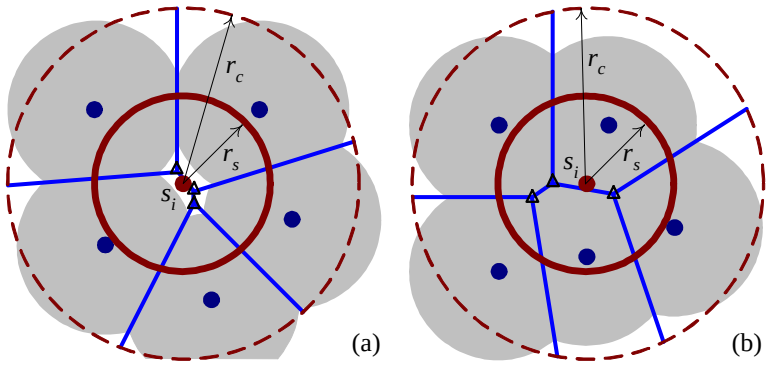


Figure 2-13. Voronoi-diagram based coverage hole prediction.

s_j‘ $\in$ *Neig*(s_i‘). Since $\|s_i - s_i\text{‘}\| \leq \delta$ and $\|s_j - s_j\text{‘}\| \leq \delta$, from the triangle inequality we have: $\|s_i - p\| \leq \|s_i\text{‘} - p\| + \|s_i - s_i\text{‘}\| \leq r_s$ or $\|s_j - p\| \leq \|s_j\text{‘} - p\| + \|s_j - s_j\text{‘}\| < \|s_i - p\|$. Hence, p is covered by *Disk*(s_i, r_s) or p is not in *LVor*(s_i), which contradicts our assumption that p is in *LVor*(s_i) and not covered by *Disk*(s_i, r_s). □

From Theorem 2.5, we can design our location-error-tolerant CIP as follows. When the location error is upper-bounded by δ, our CIP can assume the sensing range of $r_s\text{‘} = r_s - \delta$. Since $\delta > 0$, the condition that $r_c/r_s\text{‘} \geq 2$ still holds, and from the discussion in Section 2.3.6, we know that our BOND can correctly detect boundary nodes with r_s‘ and estimated locations. Based on Theorem 2.5, all the boundary nodes with the actual sensing range r_s and locations will be detected by BOND. It is worth noting that it is possible that some interior nodes in terms of their actual locations will be mislabeled as boundary nodes. Therefore, our location-error-tolerant CIP gives a conservative inference on the connected coverage, which is desirable for many WSN applications such as security-critical ones.

2.6.2 Prediction-Based CIP

Since node death caused by energy depletion is predictable, it is possible to design prediction-based CIP by exploiting the residual energy information of sensor nodes. The challenge here is that, the death of some nodes does not necessarily indicate the change of coverage especially when the WSN is densely deployed. Therefore, in order

to minimize the information relayed to the BS, we need to detect the nodes whose death caused by energy depletion will affect the coverage.

We first define healthy nodes as those with residual energy more than a threshold. Whenever a node finds that its own residual energy is less than the threshold and it is not a boundary node, it need locally broadcast this information. Let $HNeig(s_i)$ be the set of healthy neighbors of node s_i. When node s_i finds that its residual energy is less than the threshold, it first calculates the VP for each of its healthy neighbors, e.g., s_j, as follows:

$$Vor(s_j) = \bigcap_{s_k \in HNeig(s_i) - \{s_j\}} Dom(s_j, s_k).$$

Node s_i then check all the vertices of $Vor(s_j)$ ($s_j \in HNeig(s_i)$) in $Disk(s_i, r_s)$. If at least one of them is only covered by $Disk(s_i, r_s)$, the death of node s_i will cause the coverage hole (refer to Figure 2-13 (a) for an example), and node s_i will report the event that its residual energy is smaller than the threshold to the BS. Otherwise, node s_i can conclude that its death will not effect the coverage (refer to Figure 2-13 (b) for an example) and thus does not need to report itself to the BS. Based on the collected residue energy information, the BS can predict where the coverage hole will emerge with high probability. Note that, when the number of healthy neighbors of node s_i is k, there are elegant algorithms in computational geometry [123] to calculate the Voronoi diagram of $HNeig(s_i)$ with complexity $O(klogk)$, and there are at most $O(k)$ vertices need to be checked. The polygon operations in BOND can also be reused here. Since only local broadcasts are involved in our scheme, the computational and communication overhead introduced is rather small.

2.7 Chapter Summary

In this chapter, we propose a coverage inference protocol (CIP) which allows the BS to get an accurate and in-time measurement of the current connected coverage of the WSN. The CIP is built upon a novel lightweight distributed boundary node detection scheme (BOND). Detailed theoretical analysis and simulation studies show that both

BOND and CIP are highly effective and efficient. As the future research, we plan to evaluate the performance of BOND and CIP in real sensor platforms. We also intend to further investigate other potential applications of BOND in WSNs such as load balancing, topology control, distributed storage, and network health monitoring.

CHAPTER 3
LINK HETEROGENEITY AND DECENTRALIZED ROUTING

3.1 Chapter Overview

The modern view of the so-called small-world phenomenon[1] can be traced back to the famous experiments by Stanley Milgram in the 1960s [116]. His work showed that any two people in the world can be connected by a chain of (on the average) six acquaintances, and people can deliver messages efficiently to an unknown target via their acquaintances. The small-world phenomenon has also been shown to be pervasive in networks from nature and engineering systems, such as the nervous system of the nematode worm Caenorhabditis elegans [1], food webs [161], the World Wide Web [6, 160], P2P systems [8, 109, 173], "web of trust" for security systems [22], etc.

Graph models to explain why social networks develop a small diameter (maximum hopcount of the shortest paths), have been around for some time. While Erdös-Rényi random graphs possess the property of having a small diameter [16, 74], it is well-known that they are not good models for social networks because of the assumption of independence of links [160][2] . Watts and Strogatz [160] conduct a set of re-wiring experiments on graphs, and observe that by re-wiring a few random links in finite lattices, the average path length could be reduced drastically, which is smaller than logarithm of the number of vertices (nodes/points) of the graph. This leads them to propose the classic discrete small-world model which essentially consists of a lattice augmented with random links acting as shortcuts.

[1] A network is said to have the small-world property when the hopcount of the shortest path in that network is not strongly affected by an increase in the network size. Generally, the hopcount should be smaller than logarithmic in the network size.

[2] For the same reason, Erdös-Rényi random graph is also not a good model for the wireless networks, see [63] for a more detailed explanation.

The sociological experiments of Milgram demonstrated not only the existence of short chain of acquaintances between strangers but also the ability of people at finding such chains. Which graph models have this property? Specifically, when can decentralized routing algorithms (which we will formally define later) find a short path between arbitrary source and destination nodes? This question is first addressed by Kleinberg [83, 84] for the class of finite k-dimensional lattices augmented with long-range connections (shortcuts) chosen randomly from the α-harmonic distribution, that is, a long-range link between nodes u and v exists with probability proportional to $d(u, v)^{-\alpha}$, where $d(u, v)$ denotes the Manhattan distance between nodes u and v. Kleinberg shows that the simple geographic greedy routing algorithm by using only local information can route messages between any two nodes in $O(\log^2 n)$ expected number of hops if $\alpha = k$ and that there is no efficient decentralized routing algorithm if $\alpha \neq k$. Note that there is a fundamental difference between the existential discovery and the algorithmic discovery. It is quite possible that short paths exist, but that these cannot be found by any algorithm using only local knowledge of the network. For example, Kleinberg's results show that decentralized routing algorithms cannot find short routes when $\alpha \neq k$, even though such routes indeed exist for $\alpha < 2k$, as demonstrated in [30]. While it is a well recognized seminal contribution, Kleinberg's model is slightly unnatural since it is a discrete model that assumes all nodes to be located on a lattice, which is often not the case in the real world.

In this chapter, we first extend Kleinberg's result to a more realistic model constructed from a geometric network, wherein the nodes are distributed in a 2-dimensional or 3-dimensional space as a spatial Poisson point process (homogeneous or nonhomogeneous), and the probability of an edge (link/connection) between a pair of nodes u and v is given by a function $g(\cdot)$ of the distance $d(u, v)$, as well as the population between the nodes. Such spatial graph models and variants thereof arise, for instance, in the study of social networks or wireless communication networks. More importantly, we show that in

nonhomogeneous cases, the necessary and sufficient condition for greedy geographic routing to be efficient is that the probability of a shortcut being present from node u to v should be inversely proportional to the number of nodes which are closer to u than v is. Note that our model gives the same shortcut probabilities as models in previous work wherein the nodes are distributed uniformly. Therefore, our work can also be applied to homogeneous cases and gives more general condition on the navigability of any geometric network. Our result shows that it is the population-density based shortcut distribution which relates to the navigability of the geometric networks rather than the geographic-distance based shortcut distribution suggested in Kleinberg's work.

The rest of this chapter is organized as follows. In Section 3.2, we survey the related work. In Section 3.3 and Section 3.4, we provide the model of Poisson networks and the ways to determine the network parameters. In Section 3.5 we prove the main results in this chapter. Some implications and applications of our results to the wireless network engineering are discussed in Section 3.6. Finally, we conclude this chapter in Section 3.7.

3.2 Related Work

3.2.1 Related Work on Social Networks

Kleinberg's seminal work introduced a new theme in the network research literature: "navigable small-world networks". Most important advances along this theme have been summarized in Kleinberg's recent address [86] at the 2006 International Congress of Mathematicians. One interesting line of research is related to the design of decentralized routing algorithm. Recall that Kleinberg shows that the simple greedy routing algorithm by using only local information can route messages between any two nodes in $O(\log^2 n)$ expected number of hops if $\alpha = k$. This bound is tightened to $\Theta(\log^2 n)$ later by Barrière et al. [12] and Martel et al. [110]. Since the expected diameter of a k-dimensional Kleinberg network is $\Theta(\log n)$ [110], there is still some room for improving the routing performance. Further research [44, 95, 109, 110] shows that in

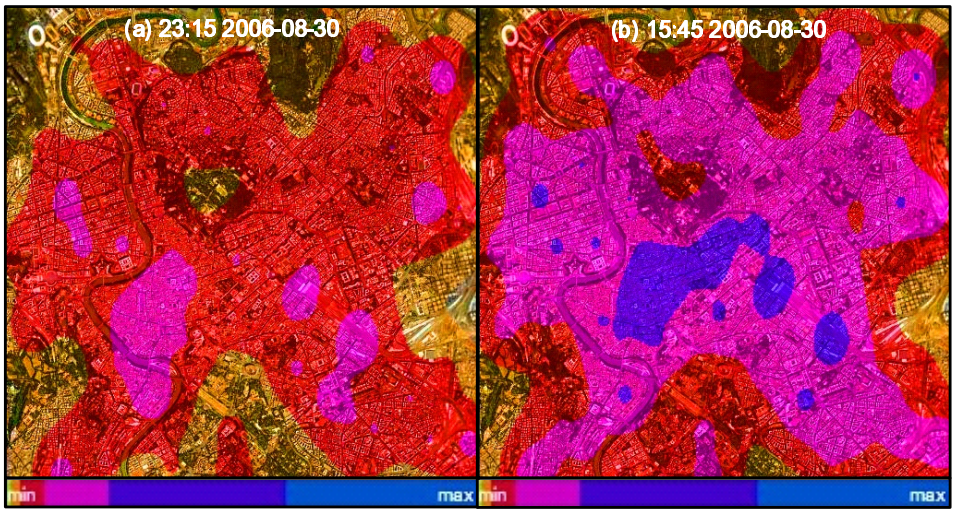


Figure 3-1. Snapshots on density of wireless users in Rome City on 30 August 2006. From Real Time Rome Project conducted by MIT SENSEable City Lab (http://senseable.mit.edu/realtimerome/).

fact the $O(\log^2 n)$ bound of the original greedy routing algorithm can be improved by putting some extra information in each message holder, which means there are some trade-off between the routing efficiency and memory space for different decentralized routing schemes. In this chapter, however, we focus on deriving the condition on the shortcut distribution which guarantees the navigability of a geometric network in a more general setting. Note that the research along this line is orthogonal and complementary to our work, in the sense that only when the geometric graph itself is navigable (which is guaranteed by our results), all those proposed more complicated geographic routing algorithms can be applied in order to further improve the routing performance.

Kleinberg's original discrete model is extended to continuum models recently, wherein the nodes are distributed as a homogeneous Poisson point process by Franceschetti and Meester [47], Draief and Ganesh [35]. The homogeneous Poisson location of people (for social networks) or the network devices (for wireless networks) reflects various irregularities of a real network architecture. This irregularity is, however, homogeneous in [35, 47], meaning that the respective mean densities are constant in the space. This assumption is often not very realistic. It is enough to take a look at a

map of the density of population of a given region (refer to Figure3-1 for an example) to realize that the social network and an optimal communication network that is supposed to reflect the traffic demand, should also be nonhomogeneous, which indicates the significance of our new model (cf. Section 3.3) which is based on a nonhomogeneous Poisson point process of node distribution.[3] Moreover, unlike the work of [35, 47] wherein the probability of shortcuts is solely determined by geographic distance, we adopt a different population-density based shortcut formation scheme.

The idea of adding shortcuts with probability inversely proportional to the number of closer candidates comes from Liben-Nowell et al.'s empirical investigation [99] of the real social network which comprises the 1,312,454 bloggers in the LiveJournal online community (www.livejournal.com), in February 2004. They find that Kleinberg's model cannot be used to explain the navigability of the LiveJournal network because of a large variance in population density across the space. They propose a new density-aware model of shortcut formation scheme to deal with the variance problem in population density. This idea is also implicit in Kleinberg's work on his group-structure model [85], which is based on people's membership in groups like organizations or neighborhoods. This model, a generalization of his lattice-based model [83, 84], introduces a long-range link between u and v with probability inversely proportional to the size of the smallest group containing both u and v. When the groups satisfy two key properties: a member of a group g must always belong to a subgroup of g that is not too much smaller than g, and every collection of small groups with a common member must have a relatively small union, Kleinberg has proven that the resulting network is a navigable small world. The main difference between the work in [85, 99] and that of this chapter is that (i) unlike

[3] Previous work shows that nonhomogeneous Poisson point process is a good approximation to the distribution of real-life mobile nodes, and has been widely used in the modeling of the spatial distribution of cellular phone users. See [9] and the references therein.

previous work based on discrete settings, to the best of our knowledge, this is the first work to prove the navigability of the geometric networks with the population-density based shortcut formation scheme in the general or nonhomogeneous continuum setting; (ii) unlike previous work with full scale invariance, in our model the scale invariance is cut off at very small and very large distances. Therefore in our model the long-range connections for each node is $O(1)$ (cf. Section 3.4.1) whereas in previous work it is $O(\log n)$. Note that our model is more realistic in that maintaining long-range connections is obviously more difficult for social or hybrid wireless communication networks and should be avoided as much as possible.

3.2.2 Related Work on Wireless Networks

Small world network model is also introduced in the wireless networking scenarios. For a wireless network, communication devices (or nodes) are distributed in a 2-dimensional or 3-dimensional space, and each node is connected to all its neighbors within some fixed radius r_c, which is called the transmission range. All those connections can be treated as local connections in Kleinberg's model [84]. Gupta and Kumar [60] show that when n nodes are distributed uniformly and randomly in the plane, the average number of hops along the shortest path between two randomly chosen nodes (source-destination pair) is $O(\sqrt{n})$. In mathematics, this kind of network is termed Random Geometric Graph (RGG) [130], which can be denoted as $G(n, r_c)$ and is obviously not a small world network. Since the forwarding burden is proportional to the average route hop-length, the hopcount on the order of n is one of the determining factors that cause the achievable throughput for each source-destination pair to approach 0 as the network size n grows to infinity [60].

In order to handle this vanishing throughput effect, many works [65, 89, 104, 133, 134, 141] suggest adding a wired infrastructure to an unstructured (ad-hoc) wireless network. Their results show that when a small number of wired or wireless (e.g., directional antenna) long-range connections are added in the ad hoc networks,

the average route hop-length can be significantly reduced, and better scaling law of throughput can be achieved.

Hybrid sensor networks with wired shortcuts have also been proposed in the literature [29, 139, 141]. A hybrid sensor network differs from a hybrid ad hoc network in that the communication in the hybrid sensor network is many-to-one (sensor-to-sink), rather than many-to-many. Based on the specific communication type of sensor networks, some wired shortcut placement schemes have been proposed. Furthermore, their corresponding improvement in energy efficiency, which is directly related to the hop-length, has been investigated.

The common problems of those works include: they do not address the problem of how to find those short paths by utilizing augmented shortcuts. Given the nodes are distributed as a homogeneous Poisson point process, the graph is connected only with the local connections, and the shortcuts are distributed in Kleinberg's fashion, Draief and Ganesh [35] show that no decentralized routing scheme can find short paths with hopcount smaller than $O(n^{\gamma})$ for any $\gamma < (2-\alpha)/6$ when $\alpha < 2$, even though there exists such short paths with $O(\log n)$ hops. Therefore, it is not necessary that all the improvement introduced by the infrastructure to the ad hoc networks can be achieved. In addition, most of these works assume that the nodes are distributed as a homogeneous Poisson point process, which is not realistic as mentioned before.

3.3 Network Model

3.3.1 Notation and Network Model

We use the following notation throughout this chapter:

- $f(n) = O(g(n))$ means that there exists a constant c and integer N such that $f(n) \leq cg(n)$ for $n > N$.
- $f(n) = \Theta(g(n))$ means that $f(n) = O(g(n))$ and $g(n) = O(f(n))$.
- $f(n) = \Omega(g(n))$ means that $g(n) = O(f(n))$.

- With high probability (w.h.p.) refers to a probability at least $1 - \epsilon(n)$, for a function $\epsilon(n)$ going to 0 with $n \to \infty$.
- **Pr** stands for probability of, and **E** is the corresponding expectation.
- log denotes the logarithm with base 2 while ln denotes the natural logarithm with base e.
- $d(u, v)$ means the Euclidean distance between two points u and v, where $u, v \in \mathbb{R}^2$ or $\mathbb{R}^3$.
- ball(u, r) means the closed ball of radius r and centered at point u, i.e. ball$(u, r) = \{w : d(w, u) \leq r\}$.
- L_{uv} is the lune of u and v where $u, v \in \mathbb{R}^2$ or $\mathbb{R}^3$, i.e., $L_{uv} =$ ball$(u, d(u, v)) \cap$ ball$(v, d(u, v))$.
- pop(A) means the number of nodes located in the region A where $A \subseteq \mathbb{R}^2$ or $\mathbb{R}^3$ is measurable.
- $|A|$ is the shorthand for 2-dimensional or 3-dimensional Lebesgue measure of a measurable set $A \subseteq \mathbb{R}^2$ or $\mathbb{R}^3$. All integrals considered will be Lebesgue integrals.

Definition 3.1. ***[Poisson Point Process]*** *A Poisson point process $\mathcal{P}$ with density measure Λ (a diffuse Radon measure) is a point process possessing the two following properties:*

- ***[Poisson distribution of point-counts]****: the number of points in a bounded Borel set B has a Poisson distribution with mean $\Lambda(B)$, i.e.,*

$$\textbf{\textit{Pr}}[\mathcal{P}(B) = m] = \frac{(\Lambda(B))^m}{m!} \cdot \exp(-\Lambda(B)) \textit{ for } m = 0, 1, 2, \ldots$$

- ***[Independent scattering]****: the number of points in k disjoint Borel sets form k independent random variables.*

We assume throughout this chapter that the Radon measure Λ has a density with respect to Lebesgue measure, and it can be written as [150]

$$\Lambda(B) = \int_B \lambda(x) dx \text{ for Borel sets } B.$$

The density $\lambda(x)$ is called the intensity function of the general Poisson point process.

In this work, we consider a network model constructed from a nonhomogeneous Poisson point process on a finite square $\mathcal{S}_n$, wherein each node is connected to all its geographic neighbors within some fixed radius, as well as possessing random shortcuts to more distant nodes. More precisely:

Definition 3.2. ***[Network Model]***

- ***[Node distribution]*** *We consider a sequence of graphs indexed by $n \in \mathbb{N}$. Nodes $\{x_i\}$ form a nonhomogeneous Poisson process defined by Def. 3.1, with density function λ having connected and compact support Ψ with smooth boundary $\partial\Psi$, and $\frac{\max_\Psi \lambda}{\min_\Psi \lambda} \leq c_\lambda$ where c_λ is a constant.*
- ***[Local link]*** *Each node x_i is connected to all nodes x_j that their Euclidean distance is not greater than $r_n = \sqrt{(c_g \cdot \ln n)/n}$ for 2-dimensional case or $r_n = \sqrt[3]{(c_g \cdot \ln n)/n}$ for 3-dimensional case. These links are referred to as local links and the corresponding nodes as the local neighbors of x_i. The graph with vertex set $V = \{x_i : 1 \leq i \leq n\}$ and edge set consisting solely of local links is a nonhomogeneous RGG denoted as $G(n, r_n)$. Parameter c_g is a constant called* GeoGreedy *parameter and will be discussed in Section 3.4.2.*
- ***[Shortcut]*** *For two nodes x_i and x_j such that $d(x_i, x_j) > r_n$, the link (x_i, x_j) is present with probability $\boldsymbol{Pr}[(x_i, x_j)] = a_n \cdot \Big(pop\big(ball(x_i, d(x_i, x_j))\big)\Big)^{-\alpha}$, where $a_n \geq 0$ and $\alpha \geq 0$ are universial constants. These links are referred to as shortcuts and the corresponding nodes as the long-range neighbors of x_i. The graph with vertex set V and edge set consisting of local links and shortcuts is called a nonhomogeneous Poisson Network denoted as $NPN(n, r_n, \alpha)$. Parameter a_n is a function of n, which is called normalization parameter and will be discussed in Section 3.4.1.*

Note that here we place some constraints on the probability density function λ of the nonhomogeneous Poisson point process, which greatly simplify the analysis. We recall that the support Ψ of a probability density function is the set of points in which it has nonzero value, and that the boundary $\partial\Psi$ is smooth if and only if it is twice differentiable. We also require that the ratio of the maximum and minimum value of λ on the support Ψ is upper-bounded by a constant number. The reason of doing so is that modeling inhomogeneity is not an easy task and more adequate, nonhomogeneous models rapidly become too difficult to analyze, and we think all those constraints are reasonable.

Here, we give some intuition behind our network model. This model has a simple "geographic" interpretation for social networks: the nodes of a social network are the people in it; two nodes are connected if these two persons know each other. Individuals know their neighbors within some fixed small radius r_n; they also have some number of acquaintances distributed more broadly across the space. Obviously, the probability that two persons u and v know each other should decrease as the geographical distance between them increases. In Kleinberg's model, this probability is solely determined by the geographical distance between them (Figure 3-2 (a)), while in our model, this probability is determined by the population in ball$(v, d(u, v))$ (Figure 3-2 (b), the shaded disk). Intuitively, one justification for this kind of shortcut distribution is the following: in order to befriend u, v will have to compete with all of the more "convenient" candidate friends for u, i.e., all people who live closer to u than v does. Therefore, in this chapter, when we are modeling the distribution of shortcuts, we consider the population densities as well as the geographical distances. Our consideration is more reasonable since the probability of two persons know each other should decrease more quickly with the geographical distance when the population density is high. It is a well-known observation that in the city with high population density, people know their geographical neighbors with a small probability while in the country, this probability will be much higher.

For wireless ad hoc networks, local links model the communication between nearby nodes through wireless links. This kind of disk communication model (with communication range r_n) is widely used in the theoretical study of wireless networks (e.g., [60]). Shortcuts can model the wired or wireless infrastructure added in this purely ad hoc network. Obviously, the probability of existence of the shortcut should decrease as the geographical distance between them increases since the cost of the shortcuts (wired infrastructure) is proportional to the total length of the wires deployed [141].

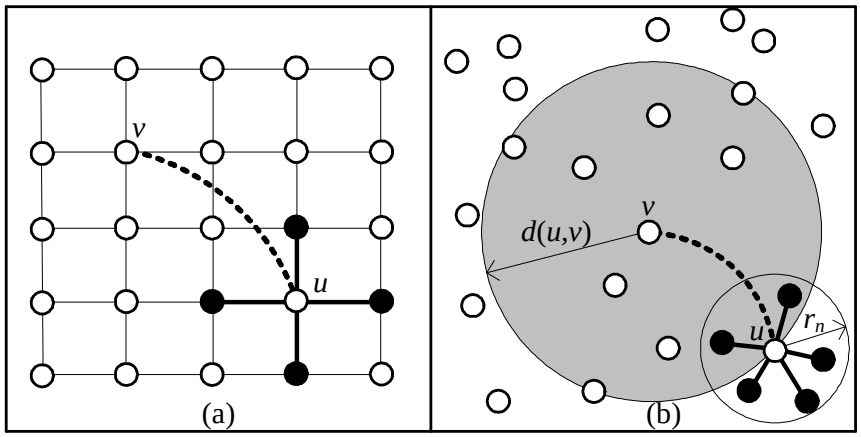


Figure 3-2. Navigable small-world network models. (a) Kleinberg's augmented lattice model in [84]; (b) Nonhomogeneous Poisson network model used in this chapter. Here stuffed nodes, bold solid lines and bold dashed curves represent the local neighbors of u, local links and shortcuts, respectively. Note that in (b), the probability of obtaining a shortcut from u to v is inversely proportional to the number of nodes within the shaded disk ball$(v, d(u, v))$.

The node density also need be included in the consideration in order to optimize the placement of the wired infrastructure.

3.3.2 Background

Here we present the formal definitions of the concepts used in this chapter.

Definition 3.3. ***[Small-World Network]*** *For a geometric network to be termed a small-world network (SWN), its diameter should be on the order of* $\log n$ *or at most polylogarithmic in* n*, where* n *is the network size.*

The objective is to route a packet from an arbitrary source node s to an arbitrary destination t using a small number of hops. We are interested in decentralized routing algorithms, which do not require global knowledge of the graph topology.

Definition 3.4. ***[Decentralized Routing]*** *A decentralized routing algorithm specifies a sequence of nodes* $s = x_1, x_2, \ldots, x_k = t$ *where the only requirement is that each node* x_i *(*$2 \leq i \leq k-1$*) should be chosen from the local or long-range neighbors of node* x_{i-1}*, and* x_i *only knows the topology of its neighbors (local information).*

One special kind of decentralized routing is called geographically greedy routing or GEOGREEDY in shorthand.

Definition 3.5. ***[Geographically Greedy Routing]*** *It is assumed that each node knows its location (coordinates) in the space, as well as the location of all its neighbors (both local and long-range), and of the destination t (the header of the packet carries the location of t). If there is no direct link from the source s to the destination t, the current packet-holder u will forward the packet to one of its local or long-range neighbors closest to the destination t. If none of these neighbors are closer to the destination of the packet than the packet-holder itself, the packet will be discarded.*

Definition 3.6. ***[Navigable Small-World Network]*** *A SWN is called navigable SWN if and only if there exists decentralized routing algorithm such that the number of hops for message delivery between any pair of nodes is of order at most polylogarithmic in n, where n is the network size.*

3.4 Characterization of the Parameters in the Network Model

There are two parameters c_g and a_n in our network model left to be determined in Section 3.3. In this section, we will show how to tune these two parameters in order to obtain the network model with the desired properties.

3.4.1 Normalization Parameter a_n and the Expected Number of Shortcuts for Each Node

Since the maintenance of shortcuts are costly both in social and wireless networks, in our model we upper-bound the expected number of shortcuts for each node as a constant number $\Theta(1)$.

For any node $v \in V$, we have:

$$\sum_{x_i \in V, x_i \neq v} \mathbf{Pr}[(v, x_i)] = a_n \cdot \left(\sum_{i=1}^{n-1} \frac{1}{i^\alpha} \right)$$
$$= \begin{cases} a_n \cdot \Theta(\frac{1}{1-\alpha} n^{1-\alpha}) & \text{if } \alpha < 1 \\ a_n \cdot \Theta(\ln n) & \text{if } \alpha = 1 \\ a_n \cdot \Theta(\frac{1}{\alpha - 1}) & \text{if } \alpha > 1 \end{cases}$$

Therefore, if we require the expected number of shortcuts for each node, i.e., $\sum_{x_i} \mathbf{Pr}[(v, x_i)]$, to be a constant number $\Theta(1)$, we need to have:

$$a_n = \begin{cases} \Theta((1-\alpha) \cdot n^{\alpha - 1}) & \text{if } \alpha < 1 \\ \Theta(1/\ln n) & \text{if } \alpha = 1 \\ \Theta(\alpha - 1) & \text{if } \alpha > 1 \end{cases} \tag{3–1}$$

3.4.2 GEOGREEDY Parameter c_g and the Expected Number of Local Neighbors for Each Node

We will implicitly assume that any node can find a local neighbor closer to the destination t than itself in the following discussion (cf. Section 3.5), and call it the successful GEOGREEDY assumption. This assumption implies that, GEOGREEDY algorithm can successfully route packets between any source-destination pairs from $V \times V$ on the network $G(n, r_n)$. We first show that this assumption holds w.h.p. if c_g is chosen to be sufficiently large.

Theorem 3.1. *Given the network model $NPN(n, r_n, \alpha)$ (cf. Def. 3.2) and the randomly chosen destination t, the sufficient condition for any node to find a local neighbor closer to t than itself w.h.p. is that*

$$c_g > \frac{6}{(4\pi - 3\sqrt{3}) \cdot \min_\psi \lambda} \approx \frac{0.82}{\min_\psi \lambda} \textit{ for 2-D case;}$$

$$\textit{or } c_g > \frac{12}{5\pi \cdot \min_\psi \lambda} \approx \frac{0.77}{\min_\psi \lambda} \textit{ for 3-D case.}$$

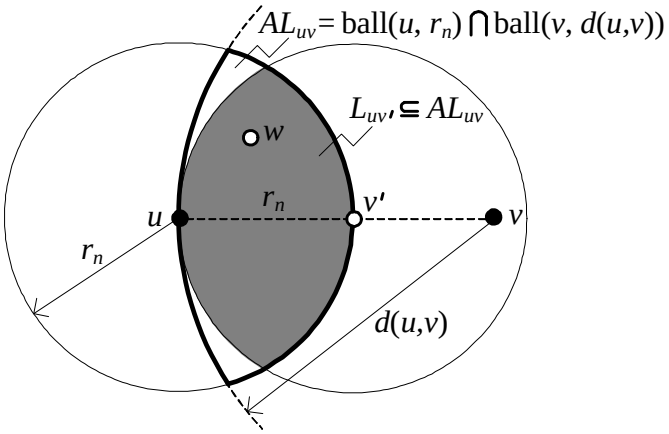


Figure 3-3. Sufficient condition for u always having a local neighbor w closer to the destination v with $d(u, v) > r_n$. Note that v' is the intersection point of the segment uv and the circulus of ball(u, r_n). The shaded area is $L_{uv'}$ which is contained in AL_{uv} (the area which is circulated by the bold curves).

Proof: For the 2-dimensional case, a straightforward calculation yields that

$$|L_{uv}| = \left(\frac{2\pi}{3} - \frac{\sqrt{3}}{2}\right) \cdot \left(d(u, v)\right)^2, \tag{3–2}$$

while for the 3-dimensional case, we obtain

$$|L_{uv}| = \frac{5\pi}{12} \cdot \left(d(u, v)\right)^3. \tag{3–3}$$

We set $\delta = \frac{2\pi}{3} - \frac{\sqrt{3}}{2}$ and focus on the 2-dimensional case in the following. The proof for the 3-dimensional case is almost the same except for setting $\delta = \frac{5\pi}{12}$.

Two key observations from Figure 3-3 are:

(i) The sufficient and necessary condition that node u can find a local neighbor closer to the destination v with $d(u, v) > r_n$ is that there exists at least one node $w \in V$ within AL_{uv}, where AL_{uv} is defined as $AL_{uv} = \text{ball}(u, r_n) \cap \text{ball}(v, d(u, v))$.

(ii) $L_{uv'} \subseteq AL_{uv}$ where v' is the intersection point of the segment uv and the circulus of ball(u, r_n).

Therefore, the sufficient condition that node u can find a local neighbor closer to the destination v is that there exists at least one node $w \in V$ within $L_{uv'}$. From Eq. (3–2), we

obtain

$$|L_{uv'}| = \delta \cdot r_n^2. \tag{3–4}$$

Let X_i be the event that node x_i does not have any local neighbors closer to the destination, and let $X = \bigcup_{i=1}^{n} X_i$ be the event that there is at least one node in V which does not have any local neighbors closer to the destination. From Eq. (3–4), we obtain

$$\mathbf{Pr}[X_i] = \left(1 - \int_{L_{x_i x_i'}} \lambda(x)dx\right)^{n-1} \leq \left(1 - \min_{\Psi} \lambda \cdot \delta \cdot r_n^2\right)^{n-1},$$

where x_i' is the intersection point of the segment $x_i't$ and the circulus of ball(x_i, r_n).

From the union bound, we can write[4]

$$\begin{aligned}
\mathbf{Pr}[X] &\leq n \cdot \left(1 - \min_{\Psi} \lambda \cdot \delta \cdot r_n^2\right)^{n-1} \\
&= e^{\ln n + (n-1)\ln(1 - \min_{\Psi} \lambda \cdot \delta \cdot r_n^2)} \\
&\leq e^{\ln n - (n-1)\min_{\Psi} \lambda \cdot \delta \cdot r_n^2} \\
&= e^{(\ln n)\cdot\left(1 - \frac{(n-1)}{\ln n}\min_{\Psi} \lambda \cdot \delta \cdot r_n^2\right)} \\
&= e^{(\ln n)\cdot\left(1 - \min_{\Psi} \lambda \cdot \delta \cdot \left(\frac{r_n}{\sqrt{\frac{\ln n}{n-1}}}\right)^2\right)}
\end{aligned}$$

where, to write the second inequality, we have used the fact that $\ln(1 + x) \leq x$. For successful GEOGREEDY assumption to be hold w.h.p., we want $\mathbf{Pr}[X] \to 0$ as $n \to 0$. From the final equation, this can be seen to happen if $\sqrt{\min_{\Psi} \lambda \cdot \delta} \cdot r_n / \sqrt{(\ln n)/n} \to \infty$, that is r_n is made to decrease strictly slower than $\sqrt{(\ln n)/(\min_{\Psi} \lambda \cdot \delta \cdot n)}$, with the ratio going to ∞ as $n \to \infty$. From our network model $NPN(n, r_n, \alpha)$, we have $r_n = \sqrt{(c_g . \ln n)/n}$. Therefore, the sufficient condition for successful GEOGREEDY assumption is $c_g > \frac{1}{\min_{\Psi} \lambda \cdot \delta}$. □

[4] For ease of presentation, we neglect edge effects in the following.

From Theorem 3.1, in order to satisfy the successful GEOGREEDY assumption, we need to set $c_g \geq \frac{0.82}{\min_\Psi \lambda}$ for the 2-dimensional case and $c_g \geq \frac{0.77}{\min_\Psi \lambda}$ for the 3-dimensional case. Obviously, the successful GEOGREEDY assumption is a stronger requirement than connectivity requirement on $G(n, r_n)$. This statement is also supported by Penrose's investigation [129] on the connectivity properties of RGG in the case of arbitrary node distribution (provided that certain technical conditions are satisfied), in which it is shown that $G(n, r_n)$ is connected w.h.p. if and only if $c_g > \frac{1}{\pi \cdot \min_\Psi \lambda} \approx \frac{0.32}{\min_\Psi \lambda}$ for the 2-dimensional case, which is smaller than our c_g for successful GEOGREEDY. Gupta and Kumar's classical work [60] shows that in order to maximize the throughput, r_n should be as small as possible when the connectivity condition is satisfied. Their conclusion is for the arbitrary routing algorithms. In our case, when GEOGREEDY is used, r_n should be as small as possible when the successful GEOGREEDY assumption is satisfied. Therefore we set $c_g = \frac{0.82}{\min_\Psi \lambda}$ in the following discussion. Note that the corresponding expected number of local neighbors $NeigLoc(u)$ is on the order of $\ln n$, which is the same as for the connectivity [129].

A recent result due to Wan and Yi et al. [156] shows that for the 2-dimensional homogeneous $G(n, r_n)$, where n nodes are distributed uniformly and randomly over a unit square, in order to satisfy the successful GEOGREEDY assumption, c_g should be greater than $1/\delta \approx 0.82$. Obviously, this is a special case of Theorem 3.1, when the normalized node density is a constant $\lambda(x) = n$, which implies that $\min_\Psi \lambda = 1$.

3.5 Navigability of Nonhomogeneous Poisson Networks

In this section we will demonstrate the main results of this chapter and the corresponding proofs on the navigability of nonhomogeneous Poisson networks. We first present a special routing algorithm used in this section.

An illustrative example is given in Figure 3-4. If there is no direct link from the source s to the destination t, the message (packet) is passed via intermediate nodes as follows. At each stage, the packet carries the address (co-ordinates) of

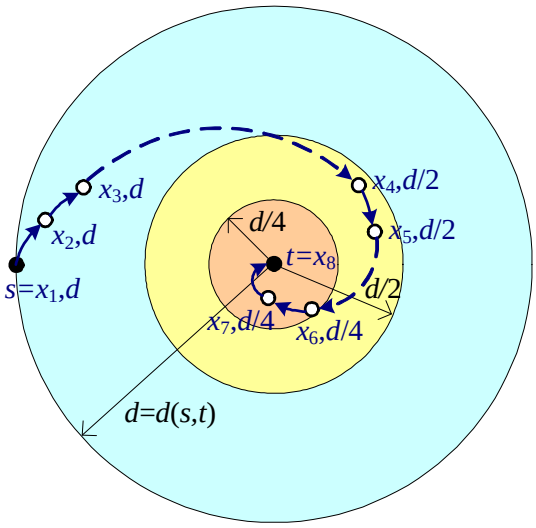


Figure 3-4. Approximate GEOGREEDY routing algorithm. Solid curves and dashed curves represent local links and shortcuts, respectively. Notation $x_4, d/2$ means it is the 4-th relay node, with the value of current indicator $d/2$.

the destination t, as well as an indicator in the packet header. The value of the indicator, i.e., d, is initialized to $d(s, t)$, the distance between s and t. Suppose that the packet is currently at node u and has the indicator value $d > r_n$.[5] If node u has a shortcut to some node $x_i \in A(t, d/2)$, where the annulus $A(t, d/2)$ is defined as $A(t, d/2) = \text{ball}(t, d/2) \backslash \text{ball}(t, d/4)$, then u forwards the message to x_i. If there is more than one such node, the choice can be arbitrary. Otherwise, it forwards the message to one of its local neighbors which is closer to t than itself. When a node x_i receives a packet, it updates d with $d/2$ if $d(x_i, t) \leq d/2$, and leaves d unchanged otherwise.

In other words, if u can find a shortcut which reduces the distance to the destination by at least a half but by no more than three-quarters, it adopts such a shortcut. Otherwise, it uses a local neighbor to reduce the distance to the destination. In that sense, we call the routing algorithm described above as approximate GEOGREEDY

[5] If $d \leq r_n$, then the node u will have contained t in its local neighbor list and will deliver the packet immediately.

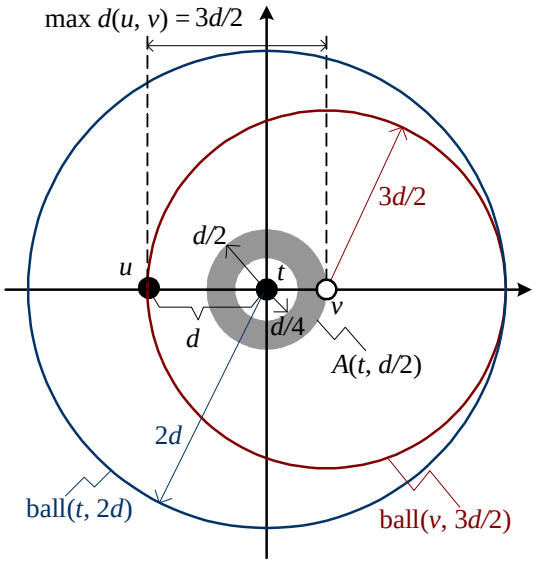


Figure 3-5. Calculating the probability of node u having a shortcut to one of the nodes in $A(t, d/2)$.

routing algorithm. The reason for considering such an algorithm rather than a GEOGREEDY defined in Def. 3.5 that minimizes the distance to the destination at each step is to preserve the independence, which greatly simplifies the analysis. Note that if a greedy step from u takes us to v (i.e., of all nodes to which u possesses a shortcut, v is closest to t), then the conditional law of the point process in the ball$(t, d(t, v))$ will be violated. The fact that there are no shortcuts from u to nodes within this ball biases the probability law and greatly complicates the analysis. Here approximate GEOGREEDY algorithm gets around this problem and has already been proved useful in [35].

3.5.1 Navigability of $NPN(n, r_n, 1)$

Theorem 3.2. *When $\alpha = 1$, $NPN(n, r_n, 1)$ is a navigable small-world network, i.e., there exists a decentralized routing algorithm, e.g., approximate* GEOGREEDY*, to route packets between any source-destination pairs chosen from $V \times V$ in $O(\log^2 n)$ hops.*

Proof: We concentrate on the 2-d case in the following. The 3-d case can be proven in a similar fashion. Before proceeding to the technical details of the proof, we begin with a brief high-level outline of the proof and some intuition of the analysis. First, we claim that the expected number of hops taken by approximate GEOGREEDY before we reach a

node halfway from the source to the destination by utilizing the shortcut is $O(\ln n)$; then we show that after at most $O(\log n)$ such halvings, we will reach the destination w.h.p.

To establish the claim, we first need to calculate the probability of finding a suitable shortcut at each step of the approximate GEOGREEDY. We think of the routing algorithm as proceeding in phases. The value of d is halved at the end of each phase when the packet reaches a node u satisfying the relation $d(u,t) \in (d/4, d/2]$ at the first time.

Figure 3-5 illustrates the following. Node u is the current packet-holder with the destination t. Denote by N_A the number of nodes in the annulus $A(t, d/2)$. Obviously

$$N_A = \text{pop}\Big(A\Big(t, \frac{d}{2}\Big)\Big) = \text{pop}\Big(\text{ball}\Big(t, \frac{d}{2}\Big) \backslash \text{ball}\Big(t, \frac{d}{4}\Big)\Big). \tag{3–5}$$

For any $v \in A(t, d/2)$ and $v \in V$, the distance $d(u,v)$ is bounded above by $3d/2$, and thus the probability that a shortcut from u is incident on a particular one of these nodes is bounded below by

$$\begin{aligned} \mathbf{Pr}[(v,u)] &\geq a_n \cdot \Big(\text{pop}\big(\text{ball}(v, 3d/2)\big)\Big)^{-1} \\ &\geq \frac{a_n}{\text{pop}\big(\text{ball}(t, 2d)\big)}. \end{aligned} \tag{3–6}$$

Thus, conditional on N_A, the probability that u has a shortcut to one of the N_A nodes in $A(t, d/2)$ is bounded below by

$$p(d, N_A) \geq 1 - \left(1 - \frac{a_n}{\text{pop}\big(\text{ball}(t, 2d)\big)}\right)^{N_A}. \tag{3–7}$$

From Eq. (3–5) and Inequality (3–7), we obtain

$$\begin{aligned} p(d, N_A) &\geq 1 - \exp\left(-\frac{N_A \cdot a_n}{\text{pop}\big(\text{ball}(t, 2d)\big)}\right) \\ &= 1 - \exp\left(-a_n \cdot \frac{\text{pop}\big(\text{ball}(t, d/2) \backslash \text{ball}(t, d/4)\big)}{\text{pop}\big(\text{ball}(t, 2d)\big)}\right) \\ &\geq a_n \cdot \frac{\text{pop}\big(\text{ball}(t, d/2) \backslash \text{ball}(t, d/4)\big)}{\text{pop}\big(\text{ball}(t, 2d)\big)} \geq a_n. \end{aligned} \tag{3–8}$$

If u does not have such a shortcut, the packet is passed via local neighbors which are successively closer to t, and hence the same lower bound on the probability of a shortcut to $A(t, d/2)$ is satisfied. Consequently, the number of local steps L_u until a shortcut is found is bounded above by a geometric random variable with conditional mean $1/p(d, N_A)$, i.e.,

$$\mathbf{Pr}[L_u = k] = (1 - p(d, N_A))^{k-1} \cdot p(d, N_A).$$

Therefore, we obtain $\mathbf{E}[L_u] \leq \frac{1}{p(d,N_A)} = \frac{1}{a_n}$. Since $a_n = \Theta(1/\ln n)$ (cf. Eq. (3–1) in Section 3.4.1), we obtain $\mathbf{E}[L_u] = O(\ln n)$. By using the Chernoff bound for a Poisson random variable, it can be shown that $L_u = O(\ln n)$ w.h.p.

Note that when $\lambda(x) = \min_\Psi \lambda$, the side length l of the smallest square $\mathcal{S}_n$ which contains n nodes is at most $O\left(\frac{\sqrt{2n}}{\min_\Psi \lambda}\right)$, which implies that the number of phases is $O(\log n)$ since the initial value of d is at most $O(\sqrt{n})$, and d is halved at the end of each phase.

Therefore, the total number of hops is $O(\log^2 n)$, which completes the proof of the theorem. □

Recall the results on homogeneous case, i.e., nodes are uniformly distributed in the space, from works of Kleinberg [84], Franceschetti and Meester [47], Draief and Ganesh [35]. They show that when the shortcut between nodes u and v exists with probability proportional to $d(u, v)^{-2}$ (for the 2-dimensional case), decentralized routing algorithm is efficient. Note that this is a special case of our Theorem 3.2. In our network model, when nodes are uniformly distributed, i.e., $\lambda(x) = \lambda$ for $\forall x \in \mathcal{S}$, we have

$$\begin{aligned} \mathbf{Pr}[(u, v)] &= a_n \cdot \Big(\mathrm{pop}\big(\mathrm{ball}(u, d(u, v))\big)\Big)^{-1} \\ &= \frac{a_n}{\lambda \pi d(u, v)^2} \propto d(u, v)^{-2}. \end{aligned}$$

Therefore, our work can also be applied to homogeneous cases and gives more general condition on the navigability of any geometric network. Our results show that it is the

population-density based shortcut distribution which determines the navigability of the geometric networks rather than the geographic-distance based shortcut distribution suggested in Kleinberg's work in a more general setting.

3.5.2 Innavigability of $NPN(n, r_n, \alpha)$ When $\alpha \neq 1$

In this subsection, we show that $\alpha = 1$ is also the necessary condition for greedy geographic routing to be efficient.

Theorem 3.3. *Suppose the source s and destination t are chosen uniformly at random from $V \times V$. **(a)** When $\alpha > 1$, the expected number of hops for routing packets between s and t is $\Omega(n^{(\alpha-1)/(2\alpha)})$; **(b)** When $0 \leq \alpha < 1$, the expected number of hops for routing packets between s and t is $\Omega(n^{(1-\alpha)/6})$.*

Proof: **(a)** For any node $u \in V$, we sort all the other nodes in V using the distance to u in the increasing order. Since nodes are distributed as a Poisson point process, no two nodes are of the same distance to u w.h.p. Therefore, we can obtain a sequence $x_1^u, x_2^u, \cdots, x_i^u, \cdots, x_{n-1}^u$, where $x_i^u \in V$ and $x_i^u \neq u$ for $1 \leq i \leq n-1$ with the property that $d(u, x_i^u) < d(u, x_{i+1}^u)$ for $1 \leq i \leq n-2$. Let v be a long-range neighbor of u, the probability that $d(u, v)$ is greater than $d(u, x_i^u)$ is bounded above by

$$\sum_{j=i+1}^{n-1} \frac{a_n}{j^\alpha} \leq a_n \cdot \int_i^\infty x^{-\alpha} dx \leq \frac{a_n}{\alpha - 1} i^{1-\alpha}.$$

Since $a_n = \Theta(\alpha - 1)$ (cf. Eq. (3–1) in Section 3.4.1), we obtain

$$\mathbf{Pr}[(u, v) | d(u, v) > d(u, x_i^u)] = O(i^{1-\alpha}). \tag{3–9}$$

For randomly chosen s and t, $d(s, t) > d(t, x_{\sqrt{n}}^t)$ w.h.p. Define $l' = n^{(1/2)-\beta} \cdot (\log n)^{-1}$. Therefore, the necessary condition for existing a path of length n^β hops between s and t is that at least one of the hops is a shortcut of length l' (in hops) or more. Let ε be the event that such a shortcut exists. By Eq. (3–9) and the union bound, we obtain:

$$\mathbf{Pr}[\varepsilon] = O\left(n^\beta \cdot \left(\frac{n^{(1/2)-\beta}}{\log n}\right)^{1-\alpha}\right). \tag{3–10}$$

It is obvious that this probability tends to 0 as $n \to \infty$ if $\beta < (\alpha - 1)/(2\alpha)$, which implies that if $\beta < (\alpha - 1)/(2\alpha)$, then the probability of finding a path with fewer than n^β hops between s and t tends to zero as $n \to \infty$.

(b) Define $\beta = (1 - \alpha)/6$. Let U denote the set of nodes within distance $d(t, x^t_{n^\beta})$ of t. Therefore, $|U| = n^\beta + 1$. For randomly chosen s and t, it is clear that s is not in U w.h.p., and for any $v \in U$, $d(s, v) > d(v, x^v_{\sqrt{n}})$ w.h.p.

Suppose now that there is a distributed routing algorithm which can find a path from s to t in fewer than n^β hops. Denote by $s = x_0, x_1, \cdots, x_m = t$, the sequence of nodes visited by the routing algorithm, with $m < n^\beta$. Then there must be a shortcut from at least one of the nodes $x_0, x1, \cdots, x_{m-1}$ to the node in U. We prove this claim by contradiction. Suppose that the current packet holder is x^t_i, then by only exploiting the local link, the next packet holder will at most be $x^t_{i-\Theta(\ln n)}$ (the expected number of local neighbors is on the order of $\ln n$, cf. Section 3.4.2). Therefore, in order to reach a node in U, we need at least $\Omega(\sqrt{n}/\ln n)$ local hops, which contradicts the assumption that the routing algorithm only needs $O(n^\beta)$ hops. Let ε be the event that within n^β hops, there exists at least one shortcut from $u \notin U$ to the node in U. For a node $u \notin U$, the probability that it has a long-range neighbor in U is upper-bounded by $O(|U| \cdot a_n)$. By Eq. (3–1) and the union bound we obtain

$$\mathbf{Pr}[\varepsilon] = O(|U| \cdot a_n \cdot n^\beta) = O(n^{2\beta} \cdot n^{\alpha-1}) = O\Big(n^{-\frac{2}{3}(1-\alpha)}\Big),$$

which goes to zero as $n \to \infty$ when $0 \leq \alpha < 1$. Therefore, we conclude that the probability of finding a path with fewer than n^β hops also tends to zero. □

Note that when $\alpha = 0$, the shortcuts are formed as an Erdös-Rényi random graph. More precisely, shortcuts are present between each pair of nodes with probability $p_n = \Theta(\ln n)$, independent of all other shortcuts. Theorem 3.3 (b) shows that $NPN(n, r_n, 0)$ is innavigable, even though the diameter of $NPN(n, r_n, 0)$ is $O(\log n)$ w.h.p.

3.6 Applications to Wireless Ad Hoc Networks

3.6.1 Does the Distribution of Shortcuts Count?

Theorem 3.2 produces a positive result, which shows that GEOGREEDY routing algorithm is efficient for our $NPN(n, r_n)$ in that it yields the short path with $O(\text{polylog } n)$ hops locally. From the proof of Theorem 3.2, we can see that the navigability of $NPN(n, r_n)$ comes from the special distribution pattern of shortcuts. In this and the next subsection, we will further investigate on this issue by providing some counterexamples with slightly different shortcut pattern, which yields short paths with $O(\text{polylog } n)$ hops while there exists no decentralized routing algorithms for finding them.

Example 1: In this example, the node distribution and the formulation of local links are the same as in the network model defined in Def. 3.2. The only difference is that the shortcuts are formed as an Erdös-Rényi random graph. More precisely, shortcuts are present between each pair of nodes with probability p_n, independent of all other shortcuts. The graph with vertex set V and edge set consisting solely of shortcuts is an Erdös-Rényi random graph (or Bernoulli random graph) denoted as $B(n, p_n)$ [16, 74]. We call this new network model the combined random network, denoted as $CG(n, r_n, p_n)$, which is the combination of nonhomogeneous RGG $G(n, r_n)$ and $B(n, p_n)$. In order to make this model comparable to our $NPN(n, r_n)$, we set $p_n = O(\frac{1}{n})$, therefore, the total number of shortcuts is on the same order of our $NPN(n, r_n)$.

In the following two theorems, we will show that, for $CG(n, r_n, p_n)$, w.h.p. there exists short paths between every pair of nodes whose lengths are bounded by a polynomial in $\log n$. However, there is no way for a decentralized routing algorithm to find these short paths.

Theorem 3.4. *The diameter of* $CG(n, r_n, p_n)$ *is w.h.p.* $\Theta(\log n)$.

Proof: For Erdös-Rényi random graph $B(n, p_n)$, we have the following well established results [16, 74]:

(i) The diameter of a connected $B(n, p_n)$ is of order $\frac{\log n}{\log np_n}$.

(ii) If $p_n = \frac{c}{n}$, $c > 1$, the unique giant component of size $O(n)$ emerges.

(iii) Let $I(n)$ be the set of nodes not in the giant component, then the expected size of $I(n)$ is $\mathbf{E}[|I(n)|] = \frac{x(\epsilon)}{\epsilon} n$ where $\frac{x(\epsilon)}{\epsilon}$ is the fraction of nodes not in the giant component, and $x(\epsilon) = \sum_{k=1}^{\infty} \frac{k^{k-1}}{k!}(2\epsilon e^{-2\epsilon})^k$, where $\epsilon = c/2 > 1/2$.

For $CG(n, r_n, p_n)$, the existence of giant component in $B(n, r_n)$ is ensured by setting $p_n = \frac{c}{n}, c > 1$. From (i), in order to prove the theorem, we only need to show that there is no shortest path longer than $\frac{\log n}{\log np_n}$ in $CG(n, r_n, p_n)$ from nodes not in the giant component of $B(n, p_n)$. A necessary condition to have a path longer than $\frac{\log n}{\log np_n}$ is to have $\frac{\log n}{\log np_n}$ nodes from $I(n)$ in an area of size less than or equal to $(\frac{\log n}{\log np_n})\pi r_n^2 = \frac{c_g \cdot \log^2 n}{n \cdot \log np_n}$. For $x_i \in I(n)$, let N_i be the number of nodes from $I(n)$ that are in an area of $\frac{c_g \cdot \log^2 n}{n \cdot \log np_n}$ which x_i belongs to. Let $N = \max\{N_i : x_i \in I(n)\}$. Since

$$\mathbf{E}[N_i] = \mathbf{E}[|I(n)|]\frac{c_g \cdot \log^2 n}{n \cdot \log np_n} = O\Big(\frac{\log^2 n}{n}\Big) \stackrel{n\to\infty}{\longrightarrow} 0,$$

it follows that $\mathbf{Pr}[N > \frac{\log n}{\log np_n}] \stackrel{n\to\infty}{\longrightarrow} 0$ and so the diameter of $CG(n, r_n, p_n)$ is of the same order as the diameter of the giant component of $B(n, p_n = c/n)$. □

Theorem 3.5. *For $CG(n, r_n, p_n)$, suppose that the source s and destination t are chosen uniformly at random from the node set V. Then, the number of hops for message delivery in any decentralized algorithm exceeds n^β w.h.p., for any $\beta < 1/3$.*

Proof: We prove by contradiction. Suppose there is a decentralized routing algorithm which can find a path from s to t in fewer than n^β hops. Denote the sequence of nodes on this path by $s = x_0, x_1, ..., x_m = t$, with $m \leq n^\beta$. Fix $\delta \in (\beta, 1/2)$ and define $C_\delta = C(t, n^\delta)$ to be the circle of radius n^δ centered at t. It is easy to show that, for any $\epsilon > 0$, the distance $d(s, C_\delta)$ from s to the circle C_δ is larger than $n^{(1/2)-\epsilon}$ w.h.p. (note that the side length l of square $\mathcal{S}$ scales as $\Theta(\sqrt{n})$). Then there must be a shortcut from at least one of the nodes $x_0, x_1, ..., x_{m-1}$ to the set C_δ. Indeed, if there is no such shortcut, then t must be reached starting from some node outside C_δ by using only local links. Since the length of each local link is at most $r_n = \sqrt{(c_g \cdot \log n)/n}$ and the number of hops

is at most n^β, the total distance traversed by local hops is strictly smaller than n^δ (for large enough n, by the assumption that $\delta > \beta$), which results in a contradiction.

We now estimate the probability that there is a shortcut from one of the nodes $x_0, x_1, \ldots, x_{m-1}$ to the set C_δ. The number of nodes in the circle C_δ, denoted by N_C, is upper-bounded by a Poisson random variable with mean $\max_\Psi \lambda \cdot \pi n^{2\delta}$, so $N_C < \max_\Psi \lambda \cdot 4n^{2\delta}$ w.h.p. By $p_n = c/n, c > 1$ and the union bound, we obtain

$$\begin{aligned} & \mathbf{Pr}\left[\exists \text{ shortcut between } u \text{ and } C_\delta \middle| N_C < \max_\Psi \lambda \cdot 4n^{2\delta}\right] \\ \leq \quad & \max_\Psi \lambda \cdot 4cn^{(2\delta-1)} \end{aligned}$$

for any node u. Applying this bound repeatedly for each of the nodes $x_0, x_1, \ldots, x_{m-1}$ on the path found by the decentralized algorithm, we obtain,

$$\begin{aligned} & \mathbf{Pr}\left[\exists \text{ shortcut to } C_\delta \text{ within } n^\beta \text{ hops} \middle| N_C < \max_\Psi \lambda \cdot 4n^{2\delta}\right] \\ \leq \quad & \max_\Psi \lambda \cdot 4cn^{(\beta+2\delta-1)}. \end{aligned}$$

By assumption, we have $\beta < 1/3$, and $\delta > \beta$ can be chosen arbitrarily. In particular, we can choose δ so that $\beta + 2\delta - 1$ is strictly negative, in which case the conditional probability of a shortcut to C_δ goes to zero as $n \to \infty$. Since $\mathbf{Pr}[N_C \geq \max_\Psi \lambda \cdot 4n^{2\delta}]$ also goes to zero, it is clear that the probability of finding an $s - t$ route with fewer than n^β hops also goes to zero, which results in a contradiction. □

Theorem 3.5 shows that the distribution of shortcuts does affect the navigability of geometric networks. The only difference between $CG(n, r_n, p_n)$ and $NPN(n, r_n)$ is that, in $CG(n, r_n, p_n)$, shortcuts are uniformly distributed over the node pairs from $V \times V$. This makes $CG(n, r_n, p_n)$ completely innavigable, even though $CG(n, r_n, p_n)$ has the same number of shortcuts and the order of diameters as $NPN(n, r_n)$.

The authors of [133, 134] suggest to uniformly and randomly add some wired shortcuts in wireless ad hoc networks in order to increase transport capacity. Even though their results show the significant improvement both theoretically and experimentally,

they fail to consider the algorithmic aspect of their scheme. Our results in this chapter show that there is no decentralized algorithm that can achieve those benefits from the short paths existing in the network.

3.6.2 Does Adding More Shortcuts Help?

Intuitively, when we add more shortcuts in the network, a smaller diameter of the network can be obtained, which implies that the shortest paths with smaller length will emerge. However, when we consider the navigability of networks, does adding more shortcuts really help?

Example 2: Given an $NPN(n, r_n)$, recall that the expected number of shortcuts for each node is $O(1)$. Now we add more shortcuts on this $NPN(n, r_n)$. The principle of this procedure is that each node will have the expected number of shortcuts of order $O(\log n)$, rendering the distribution of shortcuts between two nodes uniform. Therefore, original $NPN(n, r_n)$ will be transformed into $CG(n, r_n, p_n)$ with $p_n = O(\frac{\log n}{n})$. Following the proof procedure of Theorem 3.5, it can be shown that the new network becomes innavigable[6] , even though compared to original $NPN(n, r_n)$, no links have been deleted and only new shortcuts are added. We add a significant amount of new resource: in fact, the number of new shortcuts is $\log n$ times that of old ones, however, the performance of the network degrades greatly: the shortest path discovered by decentralized algorithm is on the order of n^{β} while it is $\log^2 n$ for the original network.

This example shows that it is the distribution pattern instead of total number of shortcuts that affects the navigability of geometric networks. In most cases, the number of shortcuts is proportional to the cost of the system. Therefore, we need to be very

[6] Theorem 3.5 shows the innavigability of $CG(n, r_n, p_n)$ with $p_n = O(\frac{1}{n})$. Here we have $p_n = O(\frac{\log n}{n})$, which is much larger but will not change the technical essence of the proof of Theorem 3.5, and hence $CG(n, r_n, p_n)$ is still innavigable.

careful when planning the network, due to the possibility that more shortcuts may lead to worse network performance.

3.7 Chapter Summary

In this chapter, we extend Kleinberg's model [84] to a more realistic model constructed from a nonhomogeneous Poisson point process, wherein each node is connected to all its neighbors within some fixed radius, as well as possessing random shortcuts to more distant nodes. More importantly, we show that in nonhomogeneous cases, the necessary and efficient condition for greedy geographic routing to be efficient is that the probability of a shortcut being present from node u to v should be inversely proportional to the number of nodes which are closer to u than v is. Note that our model gives the same shortcut probabilities as models in previous work wherein the nodes are distributed uniformly, therefore, our work can also be applied to homogeneous cases and gives more general condition on the navigability of any geometric network. To the best of our knowledge, this is the first work to prove this result for the general or nonhomogeneous continuum setting.

CHAPTER 4
SCALING LAWS FOR LARGE-SCALE MANETS WITH NETWORK CODING

4.1 Chapter Overview

One distinct characteristic of wireless mobile ad hoc networks (MANETs) is that, besides transporting data through multi-hop connected paths between the source and destination, packets can also be delivered through the physical mobility of relay nodes which is called store-carry-and-forward paradigm in the literature [41]. Grossglauser and Tse [58] have shown that significant gains in per-node throughput can be obtained by exploiting this paradigm. In particular, they proposed a 2-hop relaying scheme, and showed that it can achieve a constant per-node throughput. The scheme overcomes the throughput bound of $O(1/\sqrt{n \log n})$[1] originally established by Gupta and Kumar [60] for a static wireless network, where n is the number of nodes. Although heavy use of relaying through node mobility allows for higher throughput, it also bears two negative side-effects: increased delay and increased storage requirements. It has been shown in [48, 120] that the 2-hop relaying scheme in [58] yields an extremely large average delay of $\Omega(n)$, whereas the relay buffer size requirement on each node is at least $\Omega(n)$ [66].

Since both throughput and delay are important network performance metrics from the perspective of an application, significant effort in the last few years has been devoted to understand the throughput-delay relationship in MANETs (refer to Section 4.2.1 and the references therein) in the networking research community. An interesting work by Neely and Modiano [120] suggested to utilize redundant packets transmission through multiple opportunistic paths (which are composed of multiple opportunistic links) of a MANET to balance the conflicting requirements on throughput and delay. The basic idea

[1] Refer to Appendix-B for the standard asymptotic notation used throughout the Dissertation.

is that the time required for a packet to reach the destination (i.e., end-to-end delay) can be reduced by repeatedly transmitting this packet to many relay nodes of the network, and thus improving the chances that some user holding an original or duplicate version of the packet reaches the destination node. Clearly, the cost of this approach is the decreased throughput since duplicate packets waste scarce opportunities of wireless transmissions. In particular, with i.i.d. mobility, it was shown that for per-node throughput $T(n) = O(1)$, the relaying strategies with replication could yield end-to-end delay $D(n)$ scaling as $\Theta(n \cdot T(n))$ [120].

Buffer space of mobile nodes in MANETs is also an important and scarce network resource. Constraints on buffer space/storage reduce the throughput capacity and increase the network delay due to buffer overflow or packet losses. In practice, buffer space is always limited, and its effects on network performance should be quantified. In [66], Herdtner and Chong showed that, given the size of the relay buffer per node, say b_n, the per-node throughput is at most $O(\sqrt{b_n/n})$. The relationship of delay with storage and the impact of replication strategies as proposed in [120], however, were not addressed.

Previous studies on the scaling laws of MANETs, as discussed above, are all based on the implicit assumption that each node can only perform traditional operations on packets, such as storage, replication and forwarding. Recently, network coding, first introduced by Ahlswede et al. [2] in 2000, has been widely recognized as a promising primitive operation besides simple replicating and forwarding incoming packets [42]. Using the paradigm of network coding, when a node is scheduled to transmit, it may transmit a "mixed" packet as a result of algebraic operations on its incoming packets to maximize the usefulness of this transmission to all receivers in its transmission range. Moreover, when a node receives a new packet and its buffer is full, it will mix the new packet with stored ones in such a way that maximizes the information stored in its buffer. It is worth noting that a particular useful form of network coding called Random Linear

Coding (RLC) was proposed in the literature [67, 73] to independently and randomly mix incoming packets at each node with linear operations, which allows the nodes of the network to achieve the optimal performance while operating in a decentralized fashion.

Intuitively, when RLC instead of replication is used to minimize the end-to-end delay, network congestion can be alleviated and the requirement on buffer size can be relaxed. Therefore, a better throughput-delay-storage tradeoff is expected to obtained. Since network coding was not taken into consideration in Grossglauser and Tse's original work [58] and the related work [48, 66, 120] that followed, an interesting question raised naturally is how much benefit network coding can provide to the network performance of MANETs compared to when only simple replication and forwarding are allowed for relay nodes. Answering this question will help us better understand not only the benefits and limitations of network coding in wireless networks but also the fundamental tradeoffs determining MANET's performance.

In this chapter we study the scaling laws governing MANETs. We characterize the throughput-delay-storage tradeoffs with respect to different node mobility patterns. We identify scenarios in which network coding can provide significant improvement on network performance. We also provide insights on when and how information mixing is beneficial and propose algorithms to show that these benefits can be achieved in an effective and decentralized fashion.

The rest of the chapter is outlined as follows. Section 4.2 provides a review of related work. Section 4.3 presents models of MANETs and definitions of network performance metrics. Section 4.4 is included for comparison purposes, which summaries main results on throughput-delay-storage tradeoffs for MANETs using replication strategies, instead of network coding. Section 4.5 and Section 4.6 investigate throughput -delay-storage tradeoffs when network coding is used. Finally, Section 4.7 concludes the chapter.

4.2 Background and Related Work

There exist two main bodies of theoretical work related to the topic addressed in this chapter: the scaling laws in MANETs where network coding is not applied, and the network coding applications in MANETs where the scaling law is not studied in theoretical analysis. We review both veins of work and present recent results on scaling laws of throughput in static wireless networks with network coding as comparison. The contributions of our work are also discussed.

4.2.1 Scaling Laws of MANETs without Network Coding

Seminal work of Gupta and Kumar [60] initiated the investigation on how the throughput of wireless networks scales with n, the number of nodes. Under the assumption that nodes with common transmission range are randomly distributed, it is shown that per-node throughput for static wireless networks scales as $\Theta(1/\sqrt{n \log n})$. Note that [60] implicitly used a fluid model for establishing throughput scaling. Later work by Kulkarni and Viswannath [90] consolidated the result of [60] with an explicit constant packet size model. In [46], with percolation theory, Franceschetti et al. showed that the $\Theta(1/\sqrt{n})$ per-node throughput is achievable if each node can adjust its transmission range (through power control), however, the throughput vanishing problem for large-scale ($n \to \infty$) static wireless networks still remains. In [58], Grossglauser and Tse showed that the mobility of the nodes in a MANET can be exploited to overcome this problem. The 2-hop relaying scheme they proposed achieves a constant per-node throughput at the cost of a large delay on the order of n [48, 120]. This result reveals the possibility of trading larger delay for higher throughput or lower throughput for smaller delay in MANETs. Since then, a flurry of research activities have tried to characterize the throughput-delay relationship with respect to node mobility, e.g., [11, 48–50, 100, 101, 120, 140, 154, 165].

In general, there are two ways to trade throughput for delay in the literature. Kleinrock and Silvester [87] may be the first to find that delay of multi-hop routing

can be reduced by increasing the transmission radius of each relay node, at the expense of reducing the number of simultaneous transmissions the network can support, which leads to a lower throughput. Similar transmission radius scaling techniques have appeared in [48–50, 100, 101, 140, 154, 165]. Another approach, which improves delay via redundant packet transfers is considered in [120, 146]. In this chapter, we follow this approach, adopting replication strategy and comparing it with network coding for the following reasons:

- First of all, the assumption that transmission ranges can scale with n, the number of nodes, is impractical for large-scale MANETs. To obtain the scaling law of MANETs, we usually require n tending to infinity, which is equivalent to assuming $\sqrt{A_n} \to \infty$ for extended network model, where A_n is the area of the network (cf. Section 7.3.1). In general, wireless device is power limited, rendering it impossible to require the transmission range reaching the order of $\sqrt{A_n}$.
- Second, tradeoffs theoretically analyzed using the first means mentioned above are mainly based on fluid model, in which the packets are allowed to be arbitrarily small as $n \to \infty$ (e.g., [48, 49, 100, 101, 140, 154, 165]). On the other hand, tradeoffs obtained through the second approach assume constant packet size model, where the packet size remains constant, i.e., does not scale down with n (e.g., [120]). We prefer the constant packet size model since in reality, packet size does not change when more nodes are added to the network. Furthermore, fluid model cannot be applied to scenarios with network coding, since every coded packet includes a "code vector" of at least constant size (cf. Section 4.5.1) for successful decoding.
- Secondly, in this chapter we are interested in examining pure gains introduced by network coding in MANETs. Replication strategies can be replaced by network coding, which provides a good chance for comparison. Transmission radius scaling techniques, however, are orthogonal to network coding, and should be studied separately.

Buffer space (storage) is another important network resource, and its impact on the throughput scaling was investigated in [76] and [66]. For static wireless networks, Jelenkovic, Momcilovic and Squillante [76] showed that there was no protocol capable of carrying out the limiting per-node throughput of $\Theta(1/\sqrt{n})$ with nodes that have constant buffer space. On the other hand, they established the existence of a protocol which realizes the throughput of $\Theta(1/\sqrt{n \log n})$ when nodes have constant buffer space. For

mobile wireless networks, Herdtner and Chong [66] established that if relay buffer spaces are bounded above by a constant, mobility does not substantially increase the throughput of MANETs. In particular, they showed that the throughput is at most $O(\sqrt{b_n/n})$ per node, where b_n is the size of relay buffer spaces[2] . As a consequence, the throughput of mobile networks with finite buffer spaces is at most $O(1/\sqrt{n})$ per node, which is a large performance degradation compared to the case with infinite buffer spaces where $\Theta(1)$ throughput is achievable as established in [58]. Note that, issues related to buffer constraints can only be addressed with constant packet size model. For fluid model, buffers are not required [50] and related discussions are meaningless.

4.2.2 Network Coding Applications in Wireless Networks

The idea that, when RLC is allowed in intermediate nodes, compared to replication strategies [120, 146], larger throughput can be achieved with the same delay and smaller sizes of node buffers, was perhaps first explicitly developed in the work [175] by Zhang et al., where a simulation-based study of the benefit of RLC in one unicast communication is also presented. The recent work by Lin and Li [102] gives a rigorous analysis of this idea based on ordinary differential equations. To our knowledge, [175] and [102] are the closest to our work in terms of understanding the relationships between throughput, delay and storage with network coding. However, our work has the following advantages:

- First of all, instead of explicitly modeling nodes' spatial distributions as in this chapter, the mobility of nodes in [175] and [102] is modeled with meeting times of any pair of nodes, to simplify the analysis. The problem is that, the most important feature of wireless transmission, i.e., interference, is not included in their modeling. It is nevertheless still reasonable for [175] and [102], since the authors are mainly interested in delay tolerant networks, where nodes are assumed to be sparsely distributed and interference from simultaneous transmissions can be ignored. However, it is obviously not suitable for the study of general MANETs.

[2] We will show that this bound is not tight, i.e., it cannot be achieved by any real scheme, in Section 4.4.2.

- Second, the traffic pattern considered in our chapter is more practical. The number of unicast sessions supported in this chapter is $\Theta(n)$, while only one unicast or broadcast session is assumed in [175] and [102].

- Next, only epidemic routing and its replacement of network coding are considered in [175] and [102], while in our work, several alternatives are considered and different algorithms are developed which achieve throughput-delay-storage triples on different orders of n. Therefore, we obtain a complete characterization of tradeoffs in MANETs.

- Most importantly, explicit expressions of network performance or tradeoffs are obtained in our chapter for the first time, which provide insights on the degree of scalability of MANETs with network coding. This is the benefit brought by scaling law based approach adopted in this chapter.

4.2.3 Scaling Laws of Wireless Networks with Network Coding

Scaling laws governing wireless networks with network coding have only been investigated in the limited scenarios in the literature recently. The delay gains and reliability benefit (measured in the reduced number of transmissions) of network coding in unreliable wireless networks were characterized in [3, 37] and [52], respectively. However, these results are for one multicast session with one-hop transmission or stable network topology. For multiple unicast scenario, Liu et al. [105, 106] and Keshavarz-Haddadt et al. [81] showed that for static wireless networks, network coding and broadcasting at most provide a constant-factored improvement in the throughput, compared to Gupta and Kumar's $\Theta(1/\sqrt{n \log n})$ per-node throughput [60]. However in this chapter, our results show that, network coding can provide significant improvement on network performance when mobility is utilized, which is impossible in static wireless networks [81, 105, 106]. We believe it reveals the intrinsic difference between MANETs and static wireless networks.

Very recently, Ying et al. [165] proposed joint coding-scheduling schemes to improve the throughput-delay tradeoffs for MANETs using rate-less codes (e.g. Raptor codes). Note that their work [165] is different from ours in that: (1) They also considered adjusting transmission range (or cell size) to achieve a better tradeoff. Therefore it is

difficult to say that the gain in their scheme is purely due to the coding. (2) The results in [20] are restricted to cases when delay $D(n)$ is both $\omega(\sqrt[3]{n})$ and $o(n)$. In this chapter, we design the schemes with $D(n) = \Theta(\log n)$ and $D(n) = \Theta(n)$.

4.3 MANET Models and Definitions

In this section, we first present the mobile random network models along with the model for successful wireless transmission used in this chapter. The definitions of network performance metrics such as throughput and delay are also provided.

4.3.1 Network Models

Random network model for MANETs: Consider an ad hoc network where n nodes are initially uniformly distributed at random in a square torus of area A_n. We consider a multiple (n) unicast scenario in which each node $i \in \{1, 2, \cdots, n\}$ is a source node for one unicast session, and a destination node for another unicast session. Suppose that the source node i has data intended for destination node $d(i)$. We assume that each source node has an infinite stream of packets to send to its destination. The source-destination (S-D) association does not change with time, although the nodes themselves move.

Mobility models: The torus is divided into $m = \Theta(n)$ square cells of area A_n/m each, resulting in a two-dimensional $\sqrt{m} \times \sqrt{m}$ discrete torus, see Figure 4-1 for an illustration. The initial position of each node is equally likely to be any of the m possible cells independent of others. We further assume the time is slotted and we study the following mobility models in this chapter:

- Two-dimensional i.i.d. mobility model (fast mobility model): At each time slot, nodes randomly choose a new cell location independently and identically (i.i.d.) distributed over all cells in the network. This model captures the situation when mobile user moves so quickly that its position is almost independent from time to time. With this assumption, the network topology dramatically changes in every time slot, so that the network behavior cannot be predicted and fixed routing algorithms cannot be used. This mobility model is also used in [101, 120, 140, 154, 165].

- Two-dimensional random walk model (slow mobility model): Let a node be in cell $(i, j) \in \{1, \cdots \sqrt{m}\}^2$ at time slot t, then, at time slot $t + 1$, the node is equally

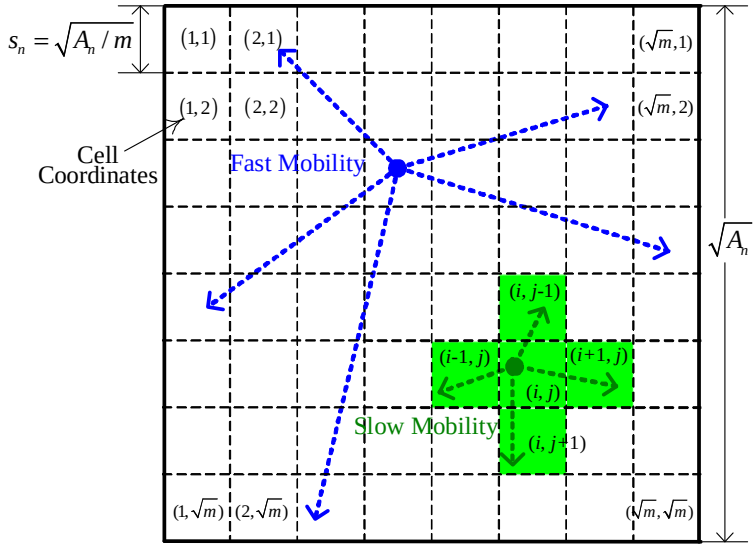


Figure 4-1. Fast and slow mobility models for MANETs.

likely to be in the same cell (i, j) or any of the four adjacent cells $\{(i-1, j), (i+1, j), (i, j-1), (i, j+1)\}$, where addition and subtraction are modulo $\sqrt{m}$. So each node in fact independently performs a simple random walk on the two-dimensional $\sqrt{m} \times \sqrt{m}$ discrete torus. Note that this model implicitly sets an upper-bound on the velocity of mobile nodes as $\sqrt{2A_n/m}$. Therefore, it is a suitable model for capturing real motion of nodes with slow mobility. Similar mobility model is also adopted in [48–50, 140, 165].

Model for successful transmission: For characterizing the condition for a successful transmission, we adopt the Protocol model as defined in [60]. We assume that all nodes use a common range r_c for their transmissions, and a transmission from node i to node j is successful if and only if $d_{ij} \leq r_c$ and $d_{kj} \geq (1+\Delta)r_c$ for any other simultaneous transmitter, say node k. Here, d_{ij} is the distance between nodes i and j, and Δ is a positive constant independent of n. During a successful transmission, nodes send data at a constant rate of W bits per second.

Concurrently transmitting cells: Now we define the transmission range and schedule. We choose r_n in such a way that any node in all cells can always directly transmit to any other node in the same cell using the smallest common range of transmission. Obviously, $r_c = \sqrt{2}s_n = \sqrt{2A_n/m} = \Theta(\sqrt{A_n/n})$.

Time is slotted for packetized transmission. We assume only $O(1)$ packets can be transmitted per cell per timeslot, i.e., our analysis is explicitly based on constant packet size model (refer to [50] and Section 4.2.1 for detailed discussions).

We say that a cell is active in a timeslot if any of its nodes transmits in that timeslot, and a cell (i, j) interferes with another cell (k, l) if a transmission by a node in cell (i, j) can affect the success of a simultaneous transmission by a node in cell (k, l). Consider in Protocol model, two interference-free cells vertically or horizontally $K - 1$ cells apart. We know that in order to guarantee the successful transmissions in these two cells, we need $(K - 1) \cdot s_n \geq (1 + \Delta) \cdot r_n$ (refer to nodes' positions at the top right of Figure 4-2 for an illustration). Let $K = \lceil 1 + (1 + \Delta)\sqrt{2} \rceil$, we divide all cells into K^2 groups. All cells belonging to the same group are at vertical or horizontal distance of exactly some multiples of K, and can transmit simultaneously as depicted in Figure 4-2. Now, we can design a finite length time-division scheduling scheme of K^2 timeslots, in which each cell group is assigned one slot to transmit. Figure 4-2 gives an example of such cell scheduling with $K = 4$. Based on the above discussion, we have the following Proposition.

Proposition 4.1. *Under the Protocol model, there exists an interference-free schedule such that each cell becomes active regularly once in K^2 timeslots and it does not interfere with any other simultaneously transmitting cells. Here K depends only on Δ, and is independent of n.*

Extended network model: We are particularly interested in asymptotic properties of MANETs, which hold with high probability[3] for large-scale MANETs. Therefore, we need often take limits as $n \to \infty$. When the region area A_n is fixed, it corresponds to a dense network model [60, 130], since the density of the network $d = n/A_n$ also tends

[3] We say that an event occurs with high probability (w.h.p.) if its probability tends to 1 as $n \to \infty$.

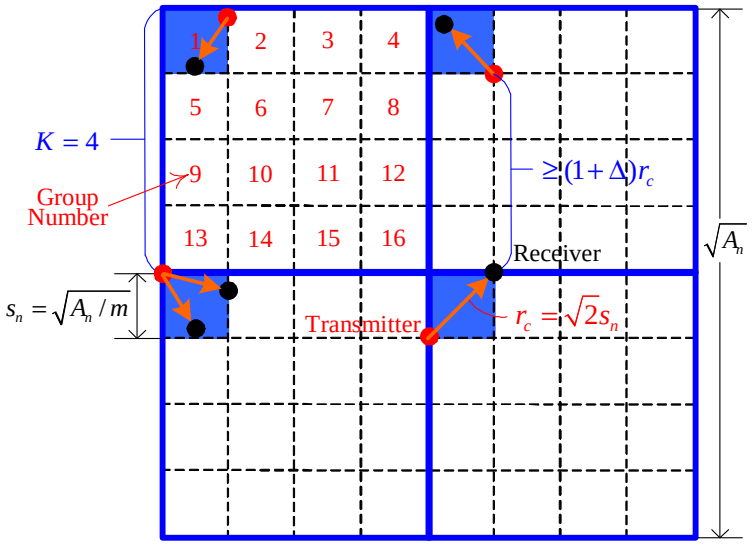


Figure 4-2. Cell transmission scheduling. Here is an illustration of the cells being divided into K^2 groups for the case of $K = 4$, i.e., 16 groups. All the blue cells which are in group 1 transmit in the same timeslot. In the next timeslot all the cells in group 2 transmit and so on.

to infinity as n. Another widely used model is the extended network model [34, 114], in which both the number of nodes and the area of the region A_n go to infinity while d is kept constant. Both models are widely used in the literature and we will focus on the latter one. In the extended network model, $A_n = n/d = \Theta(n)$, and correspondingly $r_c = \Theta(\sqrt{A_n/n}) = \Theta(1)$, which is independent of n. This is more practical, since power constraint on wireless devices does not change when more nodes are added to the network. We note that, however, results obtained in this chapter can be easily extended to dense network model.

4.3.2 Network Performance Metrics

Definition of throughput: A throughput $\lambda > 0$ is said to be feasible/achievable if every node can send at a rate of λ bits per second to its chosen destination. We denote by $T(n)$, the maximum feasible throughput w.h.p. Given a scheme Π, let $M_\Pi(i, t)$ be the number of packets from source node i that destination node $d(i)$ receives in t timeslots under scheme Π, for $1 \leq i \leq n$. Note that this could be a random quantity for a given realization of the network. Define the long term throughput of S-D pair i, denoted by

$\lambda_{\Pi}^{i}(n)$, to be

$$\lambda_{\Pi}^{i}(n) = \liminf_{t\to\infty} \frac{1}{t} M_{\Pi}(i,t).$$

Scheme Π is said to have throughput $T_{\Pi}(n)$ if

$$\lim_{n\to\infty} \mathbb{P}\left(\lambda_{\Pi}^{i}(n) \geq T_{\Pi}(n) \text{ for all } i\right) = 1.$$

We allow randomness in schemes and as a result random quantities above are in the joint probability space including both the random network of size n and the scheme Π. Note that when network coding is utilized in scheme Π, $M_{\Pi}(i,t)$ is the number of successfully decoded packets received by the destination $d(i)$ of S-D pair i in t timeslots under scheme Π.

Definition of delay : The delay of a packet is the time it takes the packet to reach the destination after it leaves the source. We do not take queueing delay at the source into account, since our interest is in the network delay. Let $D_{\Pi}^{i}(j)$ denote the delay of packet j of S-D pair i under scheme Π, then the sample mean of delay (over packets that reach their destinations) for S-D pair i is

$$\overline{D}_{\Pi}^{i} = \limsup_{k\to\infty} \frac{1}{k} \sum_{j=1}^{k} D_{\Pi}^{i}(j).$$

The average delay over all S-D pairs for a particular realization of the random network is then $\overline{D}_{\Pi} = \frac{1}{n}\sum_{i=1}^{n} \overline{D}_{\Pi}^{i}$. The delay for a scheme Π is the expectation of the average delay over all S-D pairs and all random network configurations, i.e.,

$$D_{\Pi}(n) = \mathbb{E}\left[\overline{D}_{\Pi}\right] = \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}\left[\overline{D}_{\Pi}^{i}\right].$$

Note that when network coding is utilized, we consider the delay of getting original packets. When an original packet m_i belongs to the generation M, the delay of m_i under scheme Π is the time from the first packet belonging to M departs the source to the original packet m_i has been decoded at the destination.

4.4 Throughput-Delay-Storage Tradeoffs without Network Coding

In this section, we give a brief overview of the redundancy-based schemes as presented in [120] and establish the throughput-delay-storage tradeoffs in MANETs without network coding.

4.4.1 Throughput-Delay Tradeoffs with Infinite Buffer Spaces

We first describe three relay schemes with different redundancy proposed in [120] from a unified point of view.

Three Redundancy-Based Schemes Proposed in [120]

We can control the transmission redundancy of each packet with two methods: the number of hops each packet will take from source to destination, and the total number of copies (replicas) of each original packet in the network. The three schemes, namely, 2-hop relay without replicas, 2-hop relay with k_1 replicas, and multi-hop relay with k_2 replicas represent different combinations of the two methods.

Each scheme has two parts: (1) scheduling of active cells; (2) scheduling of transmission in an active cell.

The three schemes have the same cell scheduling policy (Part (1)) as follows:

- Each cell becomes active once in every K^2 timeslots as discussed in Proposition 7.1.
- In an active cell, transmission is always between two nodes within the same cell.

In every active cell with at least two nodes, intra-cell transmission scheduling (Part (2)) is needed.

- For 2-hop relay schemes, each packet at most takes two hops from source to destination. At every timeslot, a transmitter-receiver (T-R) pair is selected randomly from all node pairs in each active cell. With probability $1/2$, the chosen T-R pair will act in source-to-relay ($S \to R$) mode or relay-to-destination ($R \to D$) mode. The difference is that, for 2-hop relay without replicas, packets are not duplicated and are held by at most one node (source or relay) at any timeslot, while for 2-hop relay with k_1 replicas, in $S \to R$ mode, the source will send k_1 replicas to distinct nodes as relays, and in $R \to D$ mode, the receiver tells its transmitter which packet it is looking for before the transmission begins (using handshake), to guarantee that the receiver always gets new useful packets when it acts as a destination node.

Table 4-1. Network performances under fast mobility model

Scheme	Throughput	Delay
2-hop relay without replicas	$\Theta(1)$	$\Theta(n)$
2-hop relay with k_1 replicas	$\Theta(1/\sqrt{n})$	$\Theta(\sqrt{n})$
Multi-hop relay with k_2 replicas	$\Theta\left(\frac{1}{n \log n}\right)$	$\Theta(\log n)$

- Multi-hop relay with k_2 replicas is just another type of flooding scheme, which transmits k_2 replicas of each packet, and places no constrains on the number of hops. It assumes that each packet is stamped with the timeslot t in which it first leaves the source. At every timeslot in each active cell, of all packets that are contained in at least one node of the cell and that have never been received by any other node in the same cell, the packet that has the smallest timestamp t will be selected to send to all nodes in the cell (i.e., the local oldest policy for packet transmission scheduling).

Theorem 4.1. *Assuming infinite buffer space at each node, throughput-delay tradeoffs achieved by the three redundancy-based schemes proposed in [120] for MANETs under fast mobility model can be summarized in Table 4-1.*

Note that the performance above is achieved with $k_1 = \Theta(\sqrt{n})$ and $k_2 = \Theta(\log n)$, respectively.

Remark 4.1. *The handshake used in $R \rightarrow D$ mode for 2-hop relay with k_1 redundancy, and the local oldest policy for packet transmission scheduling used in multi-hop relay with k_2 redundancy, are adopted to reduce unnecessary transmission redundancy and delay, which is critical for achieving optimal throughput-delay tradeoffs. Note that these two techniques can also be replaced by a naive technique called Random Message Selection (RMS) proposed in [33] and [43], which selects the packet to be transmitted randomly without considering the receiver's preference. Therefore, it is possible for RMS to schedule the transmission of a packet already in the receiver's buffer, causing decreased throughput and increased delay. For example, it has been shown that the delay for RMS is $\Theta(n \log n)$, much larger than that of the schemes described here. The benefit of RMS is that it simplifies the algorithm running in each node, and saves the communication overhead introduced by complex techniques. For extended network model used in this chapter, node density in each cell is small, i.e., $d = \Theta(1)$. The*

Table 4-2. Network performances under slow mobility model

Scheme	Throughput	Delay
2-hop relay without replicas	$\Theta(1)$	$\Theta(n \log n)$
2-hop relay with k_1 replicas	$\Theta\left(\frac{1}{\sqrt{n}\log n}\right)$	$\Theta(\sqrt{n}\log n)$
Multi-hop relay with k_2 replicas	$\Theta\left(\frac{1}{n\sqrt{n}}\right)$	$\Theta(\sqrt{n})$

overhead and complexity introduced by the two redundancy-based schemes can be ignored. However, when the node density in each cell is high, e.g., in dense network model, optimal redundancy-based schemes are impractical, and RMS is the only choice, where the gain of using network coding will be amplified by a factor of $\Theta(\log n)$ *as shown in [33] and [43].*

Theorem 4.2. *Assuming infinite buffer space at each node, throughput-delay tradeoffs achieved by the three redundancy-based schemes proposed in [120] for MANETs under slow mobility model can be summarized in Table 4-2.*

Note that the performance above is achieved with $k_1 = \Theta(\sqrt{n}\log n)$ and $k_2 = \Theta(\sqrt{n})$, respectively.

4.4.2 Throughput-Storage Tradeoffs

In this subsection, we analyze the impacts of finite or limited buffer size on the scaling properties of MANETs. Note that [66] only provides a loose upper bound on throughput given buffer size b_n (cf. Section 4.2.1). Our results presented here provide a tighter bound than the previous ones in [66].

Theorem 4.3. *Assume each node has a buffer space of* b_n *packets, the throughput is upper-bounded by* $O(\frac{b_n}{b_n+n})$*, and this bound is achievable using 2-hop relay without replicas.*

Remark 4.2. *For a given S-D pair, the average packet queue size in the network is the product of per node throughput and average packet life time using Littles law. Since the packet life time is equal to the average delay, the average number of packets for a given pair (i.e., the number of buffers required to support a given S-D pair's flow) is determined by the throughput-delay product. From this observation, the average buffer*

requirements for supporting optimal throughput-delay tradeoffs in Theorem 4.1 are $O(n)$, $O(1)$, and $O(1/n)$, respectively. Therefore, we can conclude that the higher the redundancy in a scheme, the lower the requirement on the buffer size. At a first glance, this result conflicts with our intuition since excessive replication will tend to waste buffer space. However, the results presented in Section 4.4.1 show that replication reduces both throughput and delay because we trade throughput for delay improvement. As a result, the throughput-delay product decreases as well.

4.5 Throughput-Delay-Storage Tradeoffs with Network Coding: Schemes and Results

We first review RLC used in our network coding based schemes. This bears exactly the same setup as in [33]. Then we describe the schemes developed for analyzing tradeoffs in MANETs with network coding, and identify scenarios in which RLC improves network performance of MANETs significantly.

4.5.1 Network Coding Operation

Random linear coding (RLC) is applied to a finite set of k original packets (i.e., $M = \{m_1, m_2, \cdots, m_k\}$), that is called a generation. Each packet is viewed as an r-dimensional vector over a finite field, $\mathbb{F}_q$ of size q, i.e., $m_i \in \mathbb{F}_q^r$, $i = 1, 2, \cdots, k$. If the packet size is m bits, this can be done by viewing each packet as an $r = \lceil m/\log_2(q) \rceil$-dimensional vector over $\mathbb{F}_q$ (instead of viewing each packet as an m-dimensional vector over binary field). Typically, $\mathbb{F}_{2^8}$ (i.e., $\mathbb{F}_{256}$) is used. All the additions and multiplications in the following description are assumed to be over $\mathbb{F}_q$. We assume that all the k packets in M are linearly independent. During the execution of a RLC based relay scheme, the destination node of M collects linear combinations of the packets in M. Once there are k independent linear combinations at a node, it can recover all the original packets in M successfully.

Now, consider a certain timeslot t. Let $S_v(t)$ and $S_u(t)$ denote the set of all the coded packets (each coded packet is a linear combination of the packets in M) at node

v and u, respectively, at the beginning of the timeslot t. More precisely, if coded packet $f_l \in S_v(t)$, where $l = 1, 2, \cdots, |S_v(t)|$, then $f_l \in \mathbb{F}_q^r$ has the form $f_l = \sum_{i=1}^{k} \alpha_{li} \cdot m_i$, $\alpha_{li} \in \mathbb{F}_q$. The scheme ensures that a_{li}'s are known to node v by appending each packet f_l with a "code vector", which will be explained a little later. Let $S_v(t)^-$ and $S_u(t)^-$ denote the subspaces spanned by the coded packets in $S_v(t)$ and $S_u(t)$, respectively. If $S_v(t)^- \nsubseteq S_u(t)^-$, we say node v has useful information about M for u. In timeslot t, if node v is scheduled by the scheme to transmit a packet related to M to node u, v first checks if it has useful information for u. If so, v transmits a "random" coded packet with payload $f_{new} \in \mathbb{F}_q^r$ to u, where

$$f_{new} = \sum_{f_l \in S_v(t)} \beta_l \cdot f_l, \beta_l \in \mathbb{F}_q \text{ and } \mathbb{P}(\beta_l = \beta) = \frac{1}{q}, \forall \beta \in \mathbb{F}_q.$$

It is easy to check that f_{new} is still a linear combination of the k original packets, and can be written as $f_{new} = \sum_{i=1}^{k} \theta_i \cdot m_i$ where $\theta_i = \sum_{f_l \in S_v(t)} \beta_l \cdot \alpha_{li} \in \mathbb{F}_q$. For decoding purposes, the vector $(\theta_1, \theta_2, \cdots, \theta_k) \in \mathbb{F}_q^r$, called code vector, will be appended to f_{new}, and sent as overhead. This overhead clearly requires a padding of additional $k \log_2(q)$ bits. If the packet size $m \gg \log_2(q)$, which would be the case under our constant packet size model, then the overhead required by the RLC based scheme can be ignored in our analysis.[4] We say that v sends an innovative coded packet f_{new} to u, if f_{new} can increase the dimension of the subspace $S_u(t)^-$, i.e., $\dim(S_u(t)^-)$. Note that $\dim(S_u(t)^-) \leq k$ in general and if $\dim(S_u(t)^-) = k$, node u can recover all the k original packets at once. We now recall the following key result about RLC, which says that f_{new} will be an innovative coded packet for u with probability no less than $1 - \frac{1}{q}$.

[4] More precisely, the constant packet size model for original packets means that the packet size scales as $\Theta(\log n)$ bits, since it needs to carry the ID of the destination node with $\Theta(\log n)$ bits. For a fair comparison, we require that $k = O(\log n)$ for the coded packets throughout the chapter. Therefore the overhead introduced by the code vector will not change the order of our results on $T(n)$ and $D(n)$ for RLC-based schemes.

Proposition 4.2. *(Lemma 2.1 in [33]) Let* $S_u(t)^+ = S_u(t) \cup \{f_{new}\}$ *be the subspace spanned by the code vectors in* u *at the end of timeslot* t*, i.e., after receiving a coded packet* f_{new} *from* v *according to the RLC based scheme described as above. Then,*

$$\mathbb{P}\left(dim(S_u(t)^+) > dim(S_u(t)^-) \mid S_v(t)^- \nsubseteq S_u(t)^-\right) \geq 1 - \frac{1}{q}.$$

4.5.2 RLC-Based Relay Schemes

In this subsection, we describe RLC-based relay schemes with different routing strategies, which will be used later to exploit throughput-delay tradeoffs in MANETs.

We first introduce the concept of big generation. In what follows, when we say that the source node groups $k = \omega(\log n)$ original packets into one big generation, we in fact separate these k packets into $k/\Theta(\log n)$ generations, each with $\Theta(\log n)$ packets. When the destination node tries to decode one original packet, it first needs to collect $\Theta(k)$ coded packets from the big generation (with $\Theta(\log n)$ coded packets from each generation). Therefore the overhead introduced by RLC is ignorable in our analysis (cf. footnote 4).

Schemes 1: 2-hop Relay with RLC

(1) k original packets in each source node will be grouped into one (big) generation. Each source will send $m = (1+\epsilon)k$ coded packets for each (big) generation, where ϵ is a constant.

(2) Coded packets for each generation will have the same timestamp t_p. The value of t_p is the time the first coded packet of that generation leaves the source. All coded packets of a generation will be deleted from the relay buffer at the timeslot t if $t - t_p > th_p$, where the threshold th_p depends on $D(n)$ of the scheme and will be sufficiently larger than $D(n)$.

(3) Each cell becomes active once in every K^2 timeslots as discussed in Proposition 7.1. In an active cell, transmission is always between nodes within the same cell.

(4) For an active cell with at least two nodes, a random transmitter-receiver pair is selected, with uniform probability over all possible node pairs in the cell. With probability $1/2$, the transmitter is scheduled to operate in either "Source-to-Relay" or "Relay-to-Destination" mode, described as follows:

- Source-to-Relay Mode: The transmitter sends a coded packet of its current generation, and does so upon every transmission opportunity while it is in source-to-relay mode until m coded packets have been delivered to distinct nodes. If all other nodes in the cell already have one coded packet for that generation, the source will begin to transmit coded packets from the next generation. Every node stores a single packet per S-D pair per generation. When the node receives a new packet, a relay linearly combines the incoming packet with the stored one, and replaces the stored packet with the result. Note that the nodes operate in broadcast mode, i.e., every node will hear every transmission in its range, and update the packet storage as described above.
- Relay-to-Destination Mode: If the designated transmitter has a coded packet in its relay buffer for the destination node, and the rank of coded packets of that generation in the receiver is smaller than k, the coded packet is transmitted to the designated receiver.

Remark 4.3. *Since $m > k$, we need a mechanism to stop unnecessary relay of coded packets of a generation when it is already decoded in the destination. Here we use a proactive stopping mechanism, i.e., the timestamp of each generation, since we can bound the delay of the scheme. In the analysis part presented later, we will show that $k = \Theta(n)$, and $D(n)$ for this scheme is also $\Theta(n)$ for fast and slow mobility models. Therefore, th_p should be larger than $\Theta(n)$. More complicated reactive stopping mechanisms (cf. [102] and the references therein) can be adopted to enhance the efficiency of the scheme in practice. However, we follow the simplest design for analytical tractability of the scheme.*

Schemes 2: Multi-hop Relay with RLC

(1) k original packets in each source node will be grouped into one (big) generation. Each source will send $m = (1 + \epsilon)k$ coded packets for each generation, where ϵ is a constant. Two timestamps for each generation are used. One is called the generating

time t_g, based on the time for k original packets to be grouped into a generation in the source. Another is called transmission time t_p, based on the time the first coded packet of that generation is transmitted by the source.

(2) Each cell becomes active once in every K^2 timeslots as discussed in Proposition 7.1. In an active cell, transmission is always between nodes within the same cell.

(3) For an active cell with at least two nodes, perform the following: among all packets contained in at least one node of the cell and which have useful information for some other node in the same cell, choose the packet with the smallest generating time t_g. If there are ties, choose the packet from the S-D pair i which maximizes $(t_g + i)$ mod n. Transmit this packet to all other nodes in the cell. If the selected packet is in the source, then the source will transmit the linear combination of its k original packets of the same generation, instead of a particular packet belonging to that generation.

(4) Every node stores a single packet per S-D pair per generation. When the node receives a new packet, a relay linearly combines the incoming packet with the stored one, and replaces the stored packet with the result.

(5) All coded packets of a generation will be deleted from the relay buffer at the timeslot t if $t - t_p > th_p$, where the threshold th_p depends on $D(n)$ of the scheme and should be sufficiently larger than $D(n)$.

Remark 4.4. *The generating timestamp t_g is used to construct a flooding scheme for one particular S-D pair where all n S-D pairs are active and share the network resource. It is easy to see that the packets from the oldest generation that has not been delivered to all nodes will dominate the transmissions over the whole network very quickly.*
The long-term fairness between all S-D pairs is guaranteed since in the case of ties, packets from S-D pair i are given top priority in every n timeslots. Also note that, since at one particular timeslot, only one generation from one S-D pair dominates the whole network, the number of packets each relay needs to store in step (4) is 1, i.e., just for one generation w.h.p. Another timestamp t_p used here has the same functionality as the

previous scheme. The threshold th_p should be larger than $D(n)$, scaling as $\Theta(\log n)$ and $\Theta(\sqrt{n})$, respectively, for fast and slow mobility models.

4.5.3 Main Results about RLC-Based Schemes

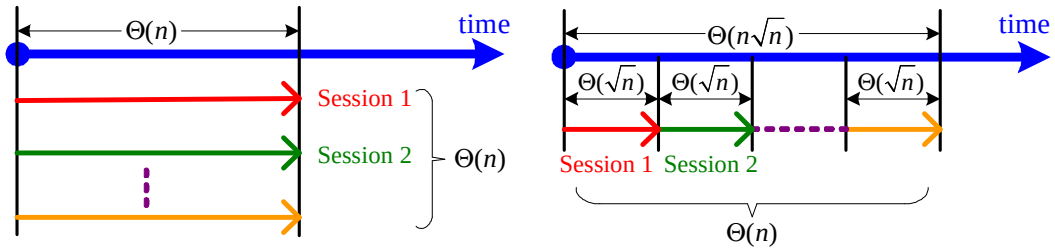
In this subsection, we summarize the performance of the above schemes under different mobility models. Here, we focus on the intuition and explanation of these results. Proofs of theses results will be given in the next section.

Theorem 4.4. *When 2-hop relay with RLC scheme is used and $k = \Theta(n)$, we have $T(n) = \Theta(1)$ and $D(n) = \Theta(n)$ for fast and slow mobility models.*

Remark 4.5. *Compare to Theorems 4.1 and 4.2, it is easy to see that, RLC provides delay improvement $\Theta(\log n)$ under slow mobility model. No gain is found under fast mobility model. It is not surprising, since 2-hop relay with RLC scheme is used to replace 2-hop relay without replicas, and we know that in the latter, there is no duplicated packets in order to maximize the throughput. Thus we cannot expect any gains when network coding is used. The gain $\Theta(\log n)$ of delay under slow mobility model comes from the lower information propagation speed, and the mixing of packets increase this speed by guaranteeing that every packet the destination received from relay nodes will contribute some information for the decoding of the packet from the same generation. For fast mobility model, this benefit vanishes since the information propagation speed is high enough, and the delay for waiting k coded packets for decoding dominates the whole delay.*

Theorem 4.5. *When multi-hop relay with RLC scheme is used, under fast mobility model with $k = \Theta(\log n)$, we have $T(n) = \Theta(1/n)$ and $D(n) = \Theta(\log n)$. Under slow mobility model with $k = \Theta(\sqrt{n})$, we have $T(n) = \Theta(1/n)$ and $D(n) = \Theta(\sqrt{n})$.*

Remark 4.6. *Under fast and slow mobility models, multi-hop RLC-based schemes always provide significant gains compared to flooding schemes. We can see that the RLC-based scheme can achieve minimal delay, with an improved delay-constrained*



(a) 2-hop relay with RLC coding (b) Multi-hop relay with RLC coding

Figure 4-3. Timetables for different RLC-based schemes under slow mobility model.

throughput. The intuition is that, when flooding is used, there exist enough opportunities to enhance performance by replacing replicas with more intelligent coding.

Figure 4-3 compares timetables of 2-hop and multi-hop RLC-based relay schemes. It can be found that in 2-hop relay schemes, multiple sessions operate in a parallel fashion, while in multi-hop relay schemes, they operate in a sequential fashion. Therefore, at each timeslot, for 2-hop relay schemes, traffic pattern is still multiple unicasts. For multiple unicasts, we seldom find gains from network coding. While for multi-hop relay schemes, at each timeslot, traffic pattern looks more like one broadcast session, where gains from network coding are naturally expected.

Remark 4.7. *Also notice that, multi-hop relay schemes can be divided into multiple phases, and in each phase, relaying for one generation from one S-D pair will dominate the network, which is in fact a type of information flooding in this phase (refer to Figure 4-3 (b) for an illustration). The result is that in each phase, packets from one generation will be broadcasted to the whole network, and if the other $n-1$ nodes are receivers, they can all decode the original packets in that generation at the end of that phase. So it guarantees that multi-hop relay with RLC coding can support all-to-all traffic pattern (n broadcast sessions) with the same performance. Note that this also means that the same network performance can be achieved for any n multicast sessions (since receivers in this case are just a subset of receivers in the broadcast case). From*

Theorem 4.5, we can easily obtain the following corollary on the performance of multiple broadcasts and multicasts with network coding.

Corollary 4.1. *For all-to-all communications or any multicasts with n sources, when multi-hop relay with RLC scheme is used, under fast mobility model with $k = \Theta(\log n)$, we have $T(n) = \Theta(1/n)$ and $D(n) = \Theta(\log n)$. Under slow mobility model with $k = \Theta(\sqrt{n})$, we have $T(n) = \Theta(1/n)$ and $D(n) = \Theta(\sqrt{n})$.*

In [43], Fragouli et al. designed an RLC-based scheme based on results from [33]. For all-to-all communications, they showed that their scheme achieves $T(n) = \Theta(1/n)$ and $D(n) = \Theta(n)$ under fast mobility model. Obviously, their scheme obtained the same throughput as ours at the cost of much larger delay. The basic idea of their scheme is that, k packets from k different sources will be grouped into one generation, and the relaying scheme is essentially the same as ours. The comparison here raises an interesting question—why in our RLC-based schemes we only mix packets from the same source? The reasons are the following: first of all, as shown in the above comparison, even for all-to-all communication scenarios, mixing packets from different sources is not a good choice. Second, for multiple unicast scenarios, we mix packets from different sources and these packets have different destinations. When one destination decodes a packet designated for another destination, this packet is in fact a duplicate at the first destination which will reduce the throughput. In our multi-hop relay with RLC, we also introduce redundancy for the same reason. However, the redundancy here is explicitly designed for decreasing the delay. While for the former case, it is purely a waste of network resource in multiple unicast scenarios. Finally, grouping packets from different sources requires coordinations. We are not sure about the cost for performing this coordination task, and we are interested in designing fully decentralized schemes, in which the operations from different nodes should be decoupled as much as possible.

From the above discussion, we can ascertain that for 2-hop relay with RLC, have one packet per S-D pair in any relay node is enough for obtaining the performance

proposed in Theorem 4.4, while one packet per node is enough for multi-hop relay with RLC. We can conclude that the storage requirements for these two schemes are exactly n and 1, respectively, for each node. Compared to the results on the storage requirements for redundancy-based schemes in Theorem 4.3, we obtain the following Corollary:

Corollary 4.2. *RLC cannot provide improvement on storage requirements better than a constant factor.*

It is easy to check that the throughput-storage tradeoffs obtained by RLC-based schemes in this chapter follow the same principle provided in Theorem 4.3.

4.6 Throughput-Delay-Storage Tradeoffs with Network Coding: Analysis

4.6.1 Preliminaries

The following lemma is useful in delay analysis, since it confirms that the effect of (intra-cell) transmission scheduling only contributes to a constant factor, which can be ignored in asymptotic analysis. Therefore, the time for two desired nodes to meet will dominate the delay of the scheme.

Lemma 3. *In the schemes mentioned above, every node will be scheduled to transmit or receive a packet in each timeslot with a constant, non-vanishing probability that is independent of n.*

Proof. This result can be obtained from Proposition 7.1. It only depends on the steady state node location distribution. Note that fast and slow mobility models have the same node location distributions in the steady state. Therefore, this result applies to both mobility models. □

To facilitate the theoretical analysis, we need first investigate two critical delays for fast and slow mobility models: minimal delays for 2-hop relays and for flooding. Here, 2-hop relay represents any scheme with controlled redundancy on the number of hops (in the 2-hop relay case, the number of hops for each packet is 2, and other schemes

with constant hop constrains will yield the critical delays on the same order of n), and flooding represents all schemes that remove this constraint totally.

Consider the following situation: initially, only one node's color is red, which we call the source. All other nodes are blue. Whenever a source node meets a blue node[5] , the latter is colored red. The time for $\Theta(n)$ nodes to become red is called minimal 2-hop delay. If we change the rule slightly: whenever a red node meets a blue node, the latter is colored red, then the corresponding time is named as minimal flooding delay. Obviously, these two critical delays reflect the intrinsic properties of how mobility will facilitate information propagation. These two quantities are scheme-independent, i.e., they hold for any scheme with or without replicas and with or without network coding.

We first consider two critical delays for fast mobility model.

Lemma 4. *The minimal 2-hop delay and the minimal flooding delay under fast mobility model are* $\Theta(n)$ *and* $\Theta(\log n)$*, respectively.*

Proof. (a) For the minimal flooding delay, Lemma 3 in [120] already established that the number of timeslots required for the source node to send packets to at least $n/2$ nodes is $\Theta(\log n)$.

(b) For the minimal 2-hop delay, we use the following ball-into-bin argument: For a given source node, we have $(n-2)$ possible relay nodes (i.e., $(n-2)$ bins). At each timeslot, the source node select 1 relay node and send a distinct coded packet to it (i.e., dropping one ball in a selected bin) with a constant probability (cf. Lemma 3). Note that we ignore this probability in the following analysis, since it will not change the order of

[5] We say that node u meets node v if and only if nodes u and v are in the same cell and scheduled as the sender-receiver pair. Recall that for random walk mobility model discussed in the literature, meeting is defined as two nodes being in the same cell. In this chapter we do not distinguish these two kinds of meets since in our model the total number of cells is $\Theta(n)$. Therefore, based on Lemma 3, the (intra-cell) transmission scheduling only contributes to a constant factor, which can be ignored in the following asymptotic analysis.

our result. Repeat this $m = \Theta(n)$ times (i.e., m balls into $(n-2)$ bins), and denote the number of distinct relay nodes as N (i.e., the number of non-empty bins). Here, we need to prove that $N = \Theta(n)$ for some $m = \varepsilon n$ where ε is a constant. Obviously, every bin is empty with probability $\exp(-m/n)$ independently when n goes to infinity. We assume these n bins as a $\sqrt{n} \times \sqrt{n}$ grid. Based on site percolation results [114], when $1 - \exp(-m/n) \geq 0.60$, the number of non-empty bins is on the order of $\Theta(n)$. Therefore, we can always find a constant ε, such that $N = \Theta(n)$ with $m = \varepsilon n$. □

Next, we present the results for slow mobility model.

Lemma 5. *The minimal 2-hop delay under slow mobility model is* $\Theta(n)$.

Proof. Here we need to show that after $m = \Theta(n)$ timeslots, there are $N = \Theta(n)$ distinct red nodes in the network. Based on Lemma 3, the source node u (the first red node) will be scheduled to transmit $m' = \Theta(n)$ times in m timeslots. Obviously $N \leq m'$ since we cannot guarantee that every time when node u is scheduled to be the sender, it will transmit a packet to a node it never met before.

Let $N(u, v, m)$ denote the number of times that node u meets v within m timeslots. Under random walk mobility model, the joint position of two nodes due to independent random walks can be viewed as a difference random walk relative to the position of one node. Then the inter-meeting times are just the inter-visit times of cell $(1, 1)$ for the difference random walk on a $\sqrt{n} \times \sqrt{n}$ torus. Let τ be the random variable representing the inter-meeting time defined as above, El Gamal et al. [49] prove that:

Lemma 6. $\mathbb{E}[\tau] = n$ *and* $Var[\tau] = \Theta(n^2 \log n)$.

Therefore, $N(u, v, m) = m/\tau = \Theta(n/\tau)$ (cf. footnote 5). Recall that for a random variable X, we have $\mathrm{Var}[f(X)] \approx (f'(\mathbb{E}[X]))^2 \mathrm{Var}[X]$. Therefore, for $m = \Theta(n)$, we have $\mathbb{E}[N(u, v, m)] = \Theta(1)$ and $\mathrm{Var}[N(u, v, m)] = \Theta(\log n)$.

Let V be the set of distinct nodes that source node u meets in m timeslots. For two distinct nodes v_1 and $v_2 \in V$, $N(u, v_1, m)$ and $N(u, v_2, m)$ are two independent random

variables with the same distribution. Note that $\mathbb{E}\left[\sum_{v\in V} N(u,v,m)\right] = m$. Because $\mathbb{E}\left[\sum_{v\in V} N(u,v,m)\right] = E[N]\cdot E[N(u,v,m)]$, we obtain that $\mathbb{E}[N] = \Theta(n)$. For two random variables X and Y, if $\mathrm{Var}[X]$ exists, we have the general formula for variance decomposition as the following: $\mathrm{Var}[X] = \mathrm{Var}[\mathbb{E}[X|Y]] + \mathbb{E}[\mathrm{Var}[X|Y]]$. Therefore,

$$\begin{aligned} Var\left[\sum_{v\in V} N(u,v,m)\right] &= \mathbb{E}\left[\mathrm{Var}\left[\sum_{v\in V} N(u,v,m)\middle| N\right]\right] \\ &\quad +\mathrm{Var}\left[\mathbb{E}\left[\sum_{v\in V} N(u,v,m)\middle| N\right]\right] \\ &= \mathbb{E}[N]\cdot\mathrm{Var}[N(u,v,m)] \\ &\quad +\mathrm{Var}[N]\cdot\mathbb{E}[N(u,v,m)] \\ &= \Theta(n)\Theta(\log n) + \mathrm{Var}[N]\Theta(1). \end{aligned}$$

Also note that $|V| \le m$, therefore

$$\mathrm{Var}\left[\sum_{v\in V} N(u,v,m)\right] \le m\cdot\mathrm{Var}[N(u,v,m)] = \Theta(n\log n).$$

From above discussions on $\mathrm{Var}\left[\sum_{v\in V} N(u,v,m)\right]$, we obtain that $\mathrm{Var}[N] = O(n\log n)$.

By Chebyshev inequality, for any $0<\kappa<1$,

$$\mathbb{P}\{N \le (1-\kappa)\mathbb{E}[N]\} \le \frac{\mathrm{Var}[N]}{\kappa^2(\mathbb{E}[N])^2} = O\left(\frac{\log n}{n}\right) \to 0,$$

which means that $N = \Theta(n)$ w.h.p. □

Lemma 7. *The minimal flooding delay under slow mobility model is* $\Theta(\sqrt{n})$.

Proof. (a) We first show that the minimal delay is $\Omega(\sqrt{n})$. From random walk model, node speed is upper bounded by $\sqrt{2A_n/m} = O(1)$ and the transmission range $r_c = \Theta(1)$. Therefore, information propagation speed will be no larger than $\Theta(1)$ per timeslot. It can be shown that the distance between the initial positions of S-D pair is $\Omega(\sqrt{A_n}) = \Omega(\sqrt{n})$ w.h.p. [90]. Hence, the expected delay is at least $\Omega(\sqrt{n})$ timeslots.

(b) We then show that the $\Theta(\sqrt{n})$ delay is achievable using flooding. We cite the following important result about rumor spreading on torus: Theorem 3 in [82] states that following the flooding rule mentioned in Section 4.6.1, at timeslot t, there exists a sub-torus of size $\sqrt{t} \times \sqrt{t}$, where for each cell in this sub-torus, there exists at least one red node. Therefore, in $\Theta(\sqrt{n})$ timeslots, we can cover the whole torus of size $\sqrt{n} \times \sqrt{n}$ w.h.p. □

4.6.2 Performance of 2-Hop Relay with RLC

In this subsection we prove Theorem 4.4 for Scheme 1.

Delay of Scheme 1 under Slow Mobility Model:

We consider a decoupled version of Scheme 1, which consists of three decoupled phases in time axis:

(1) The source node successfully transmits m coded packets of its current (big) generation to m distinct relay nodes. It takes N_1 timeslots, and obviously $N_1 \geq m = \Theta(n)$.

(2) These m relay nodes take independent random walks which takes N_2 timeslots. After this phase, m relay nodes will be uniformly distributed in the torus.

(3) The destination node collects $k = \Theta(n)$ coded packets from the network. It takes N_3 timeslots and obviously $N_3 \geq k = \Theta(n)$.

Obviously $D(n) = N_1 + N_2 + N_3$, and in what follows we will prove $D(n) = \Theta(n)$ by showing that $N_1 = N_2 = N_3 = \Theta(n)$. Note that instead of collecting coded packets as soon as possible, here the destination node begin to collect packets after $N_1 + N_2$ timeslots. Obviously this strategy is not as efficient as Scheme 1. However, its decoupling nature leads to analytical tractability. We will show that the inefficiency introduced by this decoupling strategy will not change the order of the delay of Scheme 1.

Phase 1: From Lemma 5, we directly obtain that $N_1 = \Theta(n)$.

Phase 2: From [7], we know that the mixing time of a simple random walk on a $\sqrt{n} \times \sqrt{n}$ torus is also $\Theta(n)$. Therefore, there exist a constant ε such that after $N_2 = \varepsilon n = \Theta(n)$ timeslots, these m nodes with coded packets are uniformly distributed in the torus w.h.p. which means that each node in the network has coded packets with a constant probability.

Phase 3: Given that m relay nodes are uniformly distributed over the torus, here we need to prove that after $N_3 = \Theta(n)$ timeslots, the source node can collect k distinct coded packets. Recall that Lemma 3 shows that the destination node will be scheduled as the receiver with a non-vanishing probability p_s in each timeslot, which is independent of n. Let N_3' denote the number of timeslots required by the destination node to meet m distinct relay nodes, we have that $N_3' = \Theta(N_3)$. Therefore, we only need to prove that $N_3' = \Theta(n)$. Recall the proof in Lemma 5, we know that after $N_3' = \Theta(N_3)$ timeslots, the source node will meet $k = \Theta(n)$ distinct relay nodes and obtain k distinct coded packets w.h.p.

Throughput of Scheme 1 under Slow Mobility Model:

According to above discussion, there are $\Theta(n)$ nodes that can have successful transmission simultaneously at any timeslot. Consider the transmission of packets from relays to destinations. Let $A(i,t)$ be the number of coded packets received by the destination $d(i)$ in timeslot t and $A(t) = \sum_{i=1}^{n} A(i,t)$ be the total number of coded packets received in timeslot t, we have $\mathbb{E}[A(t)] = \Theta(n)$. Note that the mobile random network is an irreducible finite-state Markov chain and $A(t)$ is a bounded non-negative function of the state of this Markov chain at time t. Therefore by the ergodicity of such a Markov chain,

$$\lim_{T\to\infty} \frac{1}{t} \sum_{t=1}^{T} A(t) = \mathbb{E}[A(t)] = \Theta(n).$$

Thus the total rate at which coded packets are transmitted from from relays to destinations is $\Theta(n)$. From the symmetry of the nodes and the randomness of the scheme it follows that each of the n destinations receives at rate of $\Theta(1)$ of coded

packets. Since each coded packet contains information of $m/n = \Theta(1)$ original packets, the throughput is still $\Theta(1)$ in original packets.

Delay of Scheme 1 under Fast Mobility Model:

We consider a decoupled version of Scheme 1, which consists of two decoupled phases in time axis:

(1) The source node successfully transmits m coded packets of its current (big) generation to m distinct relay nodes. It takes N_1 timeslots, and obviously $N_1 \geq m = \Theta(n)$.

(2) The destination node collects k coded packets from the network. It takes N_2 timeslots and obviously $N_2 \geq k$.

Phase 1: From Lemma 4, we directly obtain that $N_1 = \Theta(n)$.

Phase 2: For the destination node to meet $k = \Theta(n)$ distinct relay nodes, it is a standard coupon collector problem which requires that $N_2 = \Theta(n \log n)$. When RLC is used, it is proved in [33, 43] that the destination node only needs to collect $\Theta(n)$ coded packets, which requires that $N_2 = \Theta(n)$.

Therefore the total delay $D(n) = N_1 + N_2 = \Theta(n)$.

Throughput of Scheme 1 under Fast Mobility Model:

In [121, 122], it is shown that the capacity region depends only on the steady state node location distribution. Note that, for both i.i.d. and random walk mobility models, in steady state nodes are independently and uniformly distributed over the torus. Therefore, a given scheme will achieve the same throughput under these two different mobility models. Recall the result proved under random walk mobility model, we have $T(n) = \Theta(1)$ under i.i.d. mobility model also.

4.6.3 Performance of Multi-Hop Relay with RLC

In this subsection, we prove Theorem 4.5 for Scheme 2.

Delay of Scheme 2 under Fast Mobility Model:

We consider a decoupled version of Scheme 2, which consists of two decoupled phases in time axis:

(1) The source node successfully transmits $m = \Theta(\log n)$ distinct coded packets of its current (big) generation to relay nodes. According to Lemma 3, it only takes $N_1 = \Theta(\log n)$ time-slots on average. After this phase, according to the flooding scheme analyzed in Lemma 4 w.h.p. each node will have at least one coded packets. Obviously, some relay nodes may have the same coded packets. However, based on the property of mobility pattern (i.i.d.), the information contained in $k = \Theta(\log n)$ distinct coded packets is uniformly distributed over the torus.

(2) According to Proposition 4.2, the destination node collects k coded packets and recovers the original k packets within $\Theta(N_2) = \Theta(\log n)$ timeslots.

Throughput of Scheme 2 under Fast Mobility Model:

Note that in order to enable $\Theta(\log n)$ delay in the general case where n S-D pairs are active and share the network resources, we apply a flooding protocol in Scheme 2 in which the oldest generation that has not been delivered to all nodes is selected to dominate network resources. Scheme 2 is "fair" in that in case of ties, session i packets are given top priority every n timeslots. Since we get $k = \Theta(\log n)$ original packets in $\Theta(k)$ timeslots, the throughput for the phase when the transmissions of this S-D pair dominate the network is $\Theta(1)$. For fairness embedded in the scheme, this situation happens once for $\Theta(1/n)$ phases. Therefore, the long-term throughput is $T(n) = \Theta(1/n)$.

Delay of Scheme 2 under Slow Mobility Model:

We consider a decoupled version of Scheme 2, which consists of three decoupled phases in time axis:

(1) The source node successfully transmits $k = \Theta(\sqrt{n})$ distinct coded packets of its current generation to relay nodes. According to Lemma 3, it only takes $N_1 = \Theta(\sqrt{n})$ time-slots on average.

(2) Relay nodes with coded packets take independent random walks and perform packet "mixing" according to RLC rules. Note that, unlike i.i.d. mobility model, which will change the node position in a global-scale in each timeslot, it is not so obvious that whether after $N_2 = \Theta(\sqrt{n})$ timeslots, the information contained in $k = \Theta(\sqrt{n})$ distinct coded packets is uniformly distributed over the torus under random walk mobility model. Recall that it has been proved in Lemma 7 that based on our flooding scheme and RLC rules, w.h.p. after $\Theta(\sqrt{n})$ timeslots, every given packet can spread over the whole torus. Therefore, we can find a constant ε such that after $N_2 = \varepsilon n$ timeslots, the information contained in $k = \Theta(\sqrt{n})$ distinct coded packets is uniformly distributed.

(3) According to Proposition 4.2, the destination node collects k coded packets and recovers the original k packets within $\Theta(N_2) = \Theta(\sqrt{n})$ timeslots.

Throughput of Scheme 2 under Slow Mobility Model:

The proof is similar to the proof under fast mobility model. The only difference is replacing $k = \Theta(\log n)$ in the above proof with $k = \Theta(\sqrt{n})$ for slow mobility model.

4.7 Chapter Summary

In this chapter, we characterize the throughput-delay-storage tradeoffs in mobile ad hoc networks (MANETs) with network coding, and compares with the scenarios where only replication and forwarding are allowed in each node. The schemes/protocols achieving those tradeoffs in an effective and decentralized way are proposed. The scenarios in which network coding provides significant improvement on network performance are identified under different node mobility patterns. The insights on when and how information mixing is beneficial for MANETs with multiple unicast and multicast sessions are provided.

CHAPTER 5
PROVIDING INCENTIVES IN MULTI-HOP WIRELESS NETWORKS WITH NETWORK CODING

5.1 Chapter Overview

In a multi-hop wireless network (MWN), when the source and the destination nodes for a packet are not within direct transmission range of each other, they must rely on intermediate nodes to forward packets between them. Hence, the performance of a MWN heavily depends on the participating nodes' willingness to cooperate. If all nodes are cooperative, such as in military networks configured to behave correctly by a central authority, then cooperation can be taken for granted. However, for most current and emerging MWNs, participating nodes are owned and administered by different authorities such as different persons, and therefore are autonomous. When a node forwarding traffic for other nodes, it expends its own bandwidth and power resource without any direct benefit. A self-interested node therefore has a strong incentive to free ride, i.e., use the network resources of other nodes without contributing its own. If free-riding behaviors prevail, such networks even cannot function. Therefore, the proper design of incentive mechanism for encouraging resource sharing at the network layer is essential for the success of any MWN in civilian or commercial environments [20].

Obviously, the interactions among autonomous and self-interested entities can be modeled and analyzed as a socio-economic system, and how to stimulate cooperative behaviors in such a system is an extensively studied topic in sociology and economics with a rich collection of analyzing techniques and promising solutions [32, 117]. Therefore, it is not surprising that all proposed incentive mechanisms for MWNs in the literature draw analogies from their counterparts in human societies. Table 5-1 gives a glimpse of design space of incentive mechanisms for MWNs and points out their relationships with economic and social mechanisms enabling cooperations in human societies. Existing approaches for providing incentives basically can be classified into three categories as follows.

Table 5-1. The design space of incentive mechanisms*

Economic mechanisms	Mechanisms used in wireless networks		Social mechanisms
Bartering	Bartering (tit-for-tat) [19, 51, 115, 142, 167]		Direct reciprocity
Indirect bartering	Networked bartering [107]	Generous tit-for-tat [147]	Indirect reciprocity
Commodity currency	?	Reputation [18, 75, 111]	Reputation
Fiat currency	Virtual currency [31, 135, 158, 180, 181]		

* Note: If the reader feels uncomfortable with this table, please do not blame the author. The author of this dissertation always wants to offer the reader a clear and vivid presentation of his idea. But the Graduate School of the University of Florida has very strict and uniform formatting requirements for dissertations. For example, no vertical lines are allowed for any table and you can only have three horizontal lines for one table. The author has already tried his best to communicate with the Graduate School Editorial Office, but obviously as an international student he cannot fight the system. The good news is that this chapter has already been published by IEEE [172], so please refer to [172] for the original table. If the reader thinks that this table is in fact much better than the original one, then please thank the great work of our Graduate School Editorial Office.

The first category is barter based approaches, which are based on direct reciprocity: node A would provide resources/services for node B only if B simultaneously provides resources/services for node A. This kind of bilateral and synchronous resource/service trading makes barter extremely simple to implement. From a system perspective, there is no need to keep any long-term state information, in the form of either reputation or currency, and as a consequence the implementation cost of barter is almost zero. However, synchronous trading is easy to fail when an action and its reward are not simultaneous. This is true for MWNs: the action is packet forwarding and the benefit is being able to send its own packet. Therefore, barters in general cannot provide sufficient incentive to sustain full cooperation in a MWN. The second category is virtual-currency based, in which participating nodes would earn virtual currency by providing resources/services to others and spend the virtual currency to obtain

resources/services from others. By taking virtual currency as a medium of exchange, nodes can then trade resources/services asynchronously, which overcomes the shortcoming of barters. Virtual currency, however, incurs a high implementation overhead, e.g., billing and e-cash transfers, implementations of centralized bank and elaborating dispute-resolution mechanisms, etc. In the third category, i.e., reputation based approaches, participants build up their reputation scores by providing services for others, and highly reputed participants receive preferential treatment when they need help. Obviously, reputation scores here can be treated as another form of virtual currency. Therefore, reputation based approaches share the same advantages and disadvantages as virtual-currency based ones. The pros and cons of these approaches are also discussed in detail in Section 5.2.1.

In the spectrum of incentive mechanisms as illustrated in Table 5-1, ranging from barter at one extreme to virtual currency/reputation at the other, the sweet spot for incentive mechanism design in MWNs likely lies somewhere in-between. By exploring the whole design space, we propose a new paradigm, Controlled Coded packets as virtual Commodity Currency (**C4**), for providing incentives in a generic MWN. In our **C4**, through introducing several techniques from network coding, coded information packets are utilized as a new kind of virtual currency to facilitate resource/service exchanges among self-interested nodes in a MWN. By introducing **C4**, we make the following contributions in this chapter:

• Since the virtual currency implemented in **C4** also carries useful data information, it is the counterpart of the so-called commodity currency used in the physical world. Therefore, our **C4** fills the gap in the design space (i.e., the question mark in Table 5-1), and represents a novel design paradigm.

• By introducing a new kind of virtual currency, our **C4** can solve fundamental problems stumbling barters: asynchronicity over time and asychronicity over nodes, and at the same time only incur a low implementation overhead comparable to barter based

approaches. Therefore, our **C4** has significant advantages in comparison with existing solutions.

• We develop mathematical models to study the effectiveness and efficiency of **C4**. We theoretically show that **C4** is perfectly efficient to support MWNs with broadcast and multicast traffics. For pure unicast communications, by adjusting coding parameters, **C4** provides a systematic way to smoothly trade incentive effectiveness for small implementation cost, and traditional barter based and virtual-currency based schemes are just two extreme cases of **C4**.

• When the social contact information among mobile users is available, we propose two techniques to further reduce the implementation costs of **C4** without sacrificing incentive effectiveness for pure unicast communications in MWNs.

The rest of this chapter is organized as follows. Related work is summarized in Section 5.2. The design of **C4** for a generic MWN is described in Section 5.3. In Section 5.4 we evaluate the performance of **C4** through theoretical analysis and simulations. In Section 5.5, we show how to utilize the social network formed by mobile nodes to further improve the performance of **C4** with pure unicast communications. Conclusion and future work are described in Section 5.6.

5.2 Related Work and Motivation for C4

In this section, we first present a systematic and historical overview of incentive mechanisms for MWNs. Then the design of our **C4** is motivated. Moreover, the incentive problems of network coding itself is also investigated in this section.

5.2.1 Existing Incentive Mechanisms for MWNs

As mentioned in Section 5.1, we can use a socio-economic system (i.e., a market) to model resource/service exchanges among individual nodes, and any interaction in this system can be decomposed into a set of elementary interactions (i.e., trades illustrated in Figure 5-1) between two nodes. The time spent in searching for a

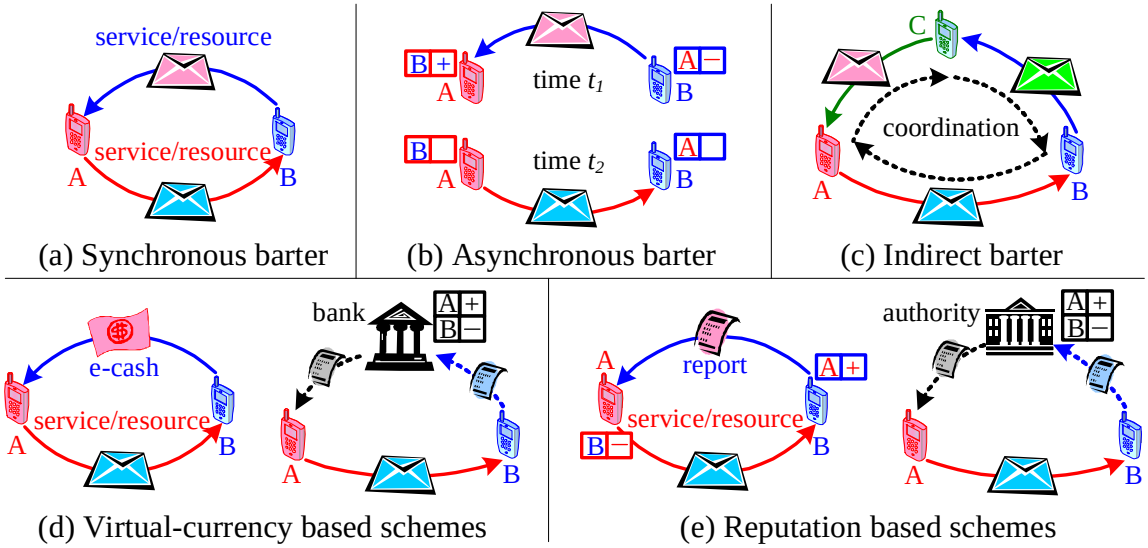


Figure 5-1. Trade models for elementary interactions between two nodes. Note that the arrow here only represents the service provider-receiver relationship between two nodes, not the data packet transmission. When the provided service is forwarding data packets, then the receiver of this service is the transmitter of data packets. When the provided service is content distribution, then the provider of this service is the transmitter of data packets.

successful trade is defined as transaction cost in economics [117], which should be minimized by an effective incentive mechanism.

[Barter Based Schemes] Barter (also called direct reciprocity in sociology), a bilateral and synchronous exchange of services/resources (cf. Figure 5-1(a)), is the oldest and simplest trading form in human history [117]. A barter can be successful only if involved two parties satisfy a “double coincidence of wants”, i.e., they are mutually interested in each other’s services/resources. Pure barter-like scheme is first proposed for content distribution applications in wireless networks [51, 167], e.g., sharing files of a popular movie among mobile nodes. Because all nodes have the common interest in all files related to the movie and every node only has part of files, when two nodes

meet they can exchange the files they do not have with high probability.[1] However, for other kinds of traffic pattern like multiple unicasts, barter is easy to fail when facing the following two asynchronicities [19, 107]: (*i*) asynchronicity across time: Two nodes might not simultaneously require each other's forwarding services. To overcome this shortcoming, asynchronous barter (also called tit-for-tat) is introduced in [115, 142]. As illustrated in Figure 5-1 (b), each node records the interaction history with other nodes and only cooperates with the nodes who also cooperate in the history. As a consequence, each node has to store information proportional to the number of nodes it interacted with, which is a large overhead in a large system. (*ii*) asynchronicity across space: Barter system is not transitive, for the situation shown in Figure 5-1 (c), if Alice (A) does a favor to Bob (B), and Bob to Carol (C), Alice cannot call for a favor from Carol. Such a situation can arise due to topological asymmetry in the positions of two nodes, and is normal in MWNs. To facilitate multilateral trading in this situation, indirect-barter based schemes like networked bartering [107] and generous tit-for-tat [147] are proposed. The common problem of this kind of schemes is that the cycle involved in indirect barter may be very long, and a large number of control packets need to be exchanged in order to coordinate all participants' behaviors. In fact, currency or reputation can be utilized as a more efficient way to coordinate participants' behaviors in this situation and decompose this circle into elementary interactions. Therefore, if we want to keep the low implementation cost of barter-like schemes, mobile nodes may need to spend more time to find another node with a "double coincidence of wants", which implies a high transaction cost and a low incentive effectiveness.

[1] Although strict synchronicity is impossible for wireless environments with the shared channel, this trade is still synchronous in the sense that it is taking place within very short time periods, e.g., on the order of seconds.

[Virtual-currency Based Schemes] The history of economics [32, 117] has shown that currency as a medium of exchange can facilitate asynchronous and multilateral trading, overcome the shortcomings of barters, and therefore minimize transaction cost. Virtual currency proposed in the literature (cf. [31, 135, 158, 180, 181]) mimics its counterpart in the physical world: participating nodes possess a certain amount of virtual currency (or credit), and if a node wants to send a packet, it has to pay all the nodes participating in relay. An uncooperative node will eventually run out of currency and stop transmitting. The payment can be direct by using some form of digital cash (e-cash) or via a central authority that serves as a bank, as shown in Figure 5-1 (d). It can be easily observed that for schemes with e-cash, the number of control packets for e-cash transfer are on the same order of that of data packets. When centralized bank is involved, even more control packets are needed for billing, dispute resolution, etc. This is a huge implementation overhead for every virtual-currency based schemes. Another common problem for those schemes is that, since the virtual currency is created ex nihilo on a market, it is extremely difficult if not impossible to maintain an reasonable average currency level within the system [117]. However, this is crucial for incentives to work properly. If the average currency level grows too high, everyone will be rich and no longer have an incentive to cooperate (just like the inflation in real world), and conversely, if there is not enough currency within the system then hardly anyone will be able to transmit.

[Reputation based Schemes] Reputation works in a similar way to currency. We earn our reputation scores by exhibiting cooperative behaviors and others decide whether to cooperate with based on our reputation scores. In MWNs, as proposed in [18, 75, 111], nodes monitor their neighboring nodes' behaviors, and give first-hand reputation scores to each other. In general, first-hand reputation information is not enough for making a fair judgement on a particular node (note that when only first-hand reputation information is available, then we obtain asynchronous barter). Therefore,

second-hand reputation information need be propagated and integrated in the system level. The reputation scores can be managed locally or globally, as illustrated in Figure 5-1 (e). For localized approach, each node will ask its neighbors to give a report about its own behaviors, and this report will be provided when the node is required to give evidence to prove its reputation. For global approaches, a trusted authority will collect second-hand reputation information, and broadcast the integrated reparation scores to the whole network. Like virtual-currency based schemes, the maintaining and updating of reputation information in the system level incur high implementation overhead. Reputation based schemes also have several inherited problems as follows. Firstly, reputation-based solutions usually cannot resist Sybil attack, whitewashing attack, and collusion. Also it is difficult to secure second-hand reputation propagation. Secondly, it is almost impossible to evaluate the incentives provided by reputation in a formal manner. The third weakness in reputation-based schemes is their heavy reliance upon the premise that misbehavior can be detected by neighboring nodes. However, to implement effective misbehavior detection schemes among mobile users is really difficult in practice.

5.2.2 Motivation for Our C4

From above discussions, we can see that existing solutions are either less effective or incur high implementation costs, and therefore do not fit well with the unique requirements of MWNs. A new design paradigm is needed.

The first question we should ask is: Is there any room for further exploration? If we go through the design space of incentive mechanisms, there do exist an unexplored area which is the counterpart of commodity currency in the physical world (i.e., the question mark in Table 5-1). Economists define currency (or money) as a generally acceptable medium of exchange [32, 117]. With the use of currency, the problem of "double coincidence of wants" is avoided, and the transaction cost of searching a successful trade is reduced. Historically, currency originated as commodity currency.

Table 5-2. Cost analysis of different incentive mechanisms

Cost \ Scheme	In the physical space			In the digital space		
	Barter	CC	FC/Rpt	Barter	**C4**	VC/Rpt
Transaction	High	Low	Low	High	Low	Low
Transportation	Zero	High	Low	Zero	Low	Low
Implementation	Zero	Zero	Low	Zero	Low	High

When a physical commodity (e.g., compressed tea leaves in ancient China) has value to everyone and is generally accepted as the payment for other goods/services, it can serve as a currency. This kind of currency has intrinsic value (i.e., valuable in its own right), and still can be consumed as a physical commodity (i.e., to make tea) when not needed for trade. The problem of commodity currency like tea leaf bricks is that it is heavy and hard to transport from one place to another. Fiat currency (paper currency) is invented to reduce the costs involved in storing and carrying commodity currency in trading (i.e., transportation cost). Fiat money is without intrinsic value as a physical commodity, and derives its value by being declared by a government to be legal tender. Obviously, virtual currency discussed in Section 5.2.1 is exactly the counterpart of fiat currency in the physical word, because virtual currency has no intrinsic value, i.e., it carries no useful data and is just used as a medium of exchange.

So, our second question comes: What is the counterpart of commodity currency in the digital world? There are only two conditions for a likely candidate: (*i*) it should carry data information because the commodity in a MWN can only be data packet; (*ii*) it has value to everyone. Following these clues, the key idea of our **C4** naturally emerges: coded packets can serve as virtual commodity currency for MWNs. By utilizing network coding, original data packets from/to different mobile users are mixed to produce coded packets. As a consequence, each coded packet has value to everyone, and is ready to act as virtual currency to facilitate cooperations.

The only question left then is: What are the benefits of using virtual commodity currency (i.e., **C4**) in MWNs? To fairly evaluate an incentive mechanism, we need

consider three kinds of costs. In our daily life, compared to the exchange of physical commodities, the implementation cost of fiat currency (FC) or the mouth-to-mouth reputation (Rep) can be ignored. As described in Table 5-2, in the physical space: barter has a high transaction cost; commodity currency (CC) has a high transportation cost; only FC can keep all three costs listed in Table 5-2 low, and therefore is the best choice. This explains the following fact: nearly all contemporary economic systems are based on fiat currency. So what happens in the digital space like a WMN? The key point here is that the goods/services in a WMN are information packets/packet transmissions. Virtual currency (VC) and reputation are also stored and transacted in the form of information bits. Therefore, compared to the data packet communication, the implementation cost of VC/Rep cannot be ignored and in fact is pretty high. The implementation cost of our **C4** is always smaller than that of VC, because when e-cash is used as the medium of exchange, it only represents control overhead; while when coded packet is used, it also carries useful data. Unlike the physical commodity, transportation costs of coded packets are small. Therefore, from Table 5-2, we can see that **C4** is the best choice in the digital space.

5.2.3 Network Coding and Incentives

The benefits of utilizing network coding in wireless networks, such as improving throughput, reducing energy consumption, simplifying transmission scheduling, etc., have been extensively studied in the literature (cf. [170] and references therein). However, as far as we know, our **C4** is the first to demonstrate a new potential benefit brought by network coding for wireless networks, i.e., providing incentives for cooperation. Incentive problems in wireless networks with network coding have also been studied in [28, 176]. Both of them apply game theory to study coding strategies adopted by individual nodes and still use virtual currency (credit) to create incentives. In our **C4**, we assume that the main application of MWNs is to provide Internet access services, and most traffic in a MWN will go through the infostation (defined below), which

enforces random linear network coding (cf. Section 7.3.1) to improve the whole system's efficiency. Therefore, no room is left for self-interested mobile nodes to change coding strategies, and coded packets can be safely utilized to create incentives. By treating network coding as a tool to produce virtual commodity currency, our **C4** provides an effective and lightweight solution to induce cooperation, which is impossible for [28, 176] based on traditional virtual currency.

5.3 Design and Implementation of Our C4

In this section, we describe the design of our **C4**. We first clarify the target scenarios and basic assumptions.

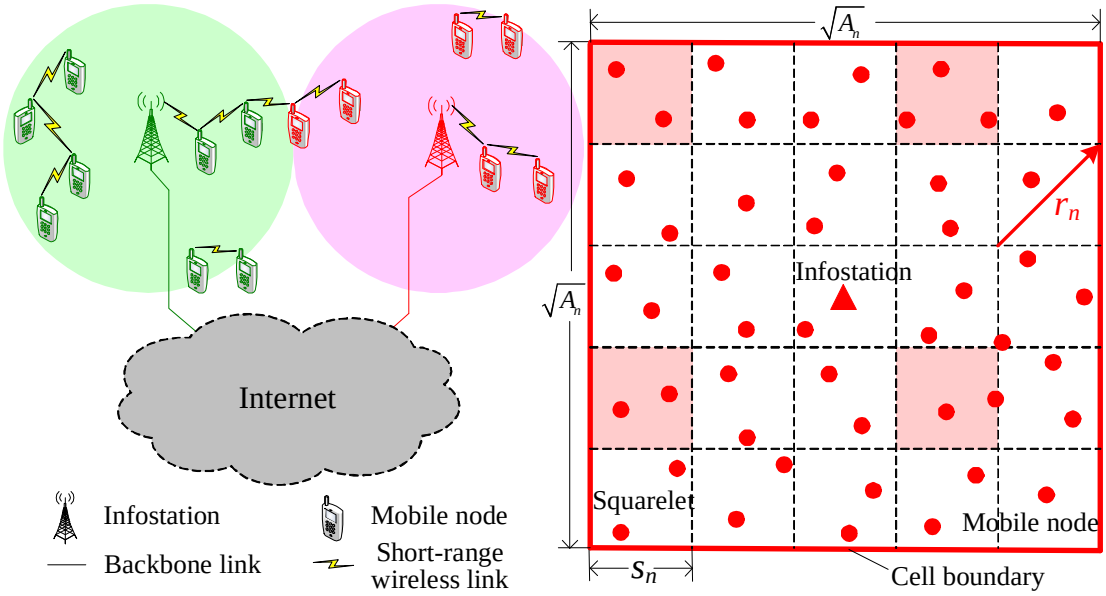
5.3.1 System Model and Problem Formulation

Based on the common requirements of future mobile communication environment [5, 72, 103, 162], our **C4** assumes the following generic model for MWNs. As illustrated in Figure 6-1 (a), there are two kinds of entities: mobile nodes and infostations.

• Mobile nodes are controlled by autonomous and self-interested clients and are interested in Internet access services. A mobile node can establish a short-range wireless link (e.g., Wi-Fi) with other mobile nodes in its vicinity. The short range links tend to be intermittent because of node mobility.

• Infostations are managed by the system operator and directly connected to the Internet with reliable and high-bandwidth links (i.e., backbone links). An infostation can use a long-range low-bandwidth radio (e.g., cellular interface) to connect with a remote mobile node (but we do not assume that all mobile nodes are covered by infostations), or use a short-range wireless link with high data rate to connect with a close mobile node. Infostations are the data sources within the wireless domain and want to provide better services to clients.

As illustrated in Table 5-3, this generic model includes several important multi-hop wireless network architectures which attract great interest from both academia and industry.



(a) Multi-hop wireless network (b) Network model for performance analysis

Figure 5-2. A generic architecture for multi-hop wireless networks.

We assume that all mobile nodes are self-interested and still rational. They have the non-cooperative behaviors mainly because they want to save resource such as bandwidth and battery power. We also assume that all infostations are under the control of one authority, and they will do all they can to encourage mobile nodes to use short-range links.

Unlike previous work [18, 28, 31, 75, 111, 135, 158, 176, 180, 181], our **C4** does not make any assumptions about routing protocols used in MWNs. Our **C4** is designed to support traditional store-and-forward routing schemes as well as new routing schemes in a DTN fashion (i.e., store-carry-and-forward [178]). Also our **C4** allows all possible combinations of traffic patterns (e.g., broadcasts, multicasts and unicasts). The only requirement is that most traffics go through infostations.

5.3.2 Methodology of Our C4

We take one broadcast session from the infostation to mobile nodes as an example to illustrate the basic design of our **C4**.

Table 5-3. Examples of multi-hop wireless networks

Multi-hop wireless network	Infostation	Mobile node
Multi-hop cellular network [103]	Base station	Mobile station
Wireless mesh network [5]	Mesh router	Wireless client
Mobile social network [72]	Service provider	Mobile user
Vehicular ad hoc network [162]	Roadside info. unit	Vehicle

Suppose this broadcast session is about distributing a movie stored in a web server in the Internet. The infostation treats all network layer packets received from the web server as original data packets (OPs). We assume each OP has l bytes. At the infostation, random linear network coding (RLNC) [67] is applied to a finite set of k OPs (i.e., $\mathtt{OP}_1$, $\mathtt{OP}_2$, ..., $\mathtt{OP}_k$), which is called a generation. Then each generation can be regarded as a $k \times l$ matrix $\mathtt{OP}$, with rows being the k OPs of the generation, and columns the l bytes of each OP. The encoding operation produces a linear combination of the OPs by $\mathtt{CB} = \mathtt{CV} \cdot \mathtt{OP}$, where $\mathtt{CV}$ is an $k \times k$ matrix composed of randomly selected coding coefficients in the Galois field $\mathbb{GF}_q$ of size q. The coded data blocks (rows in $\mathtt{CB}$) and the coding vectors (rows in $\mathtt{CV}$) are concatenated as the coded data packets (CPs). For example, the coded data block $\mathtt{CB}_j = \sum_{i=1}^{k} \mathtt{CV}_{ji} \cdot \mathtt{OP}_i$ where $\mathtt{CV}_{ji} \in \mathbb{GF}_q$ and the coded packet $\mathtt{CP}_j = \mathtt{CV}_j \parallel \mathtt{CB}_j$. Two coded packets $\mathtt{CP}_i$ and $\mathtt{CP}_j$ are called independent if $\mathtt{CV}_i$ and $\mathtt{CV}_j$ are independent vectors. In our **C4**, instead of sending OPs, the infostation sends CPs to mobile nodes with short-range links. The decoding operation at the destination (i.e., mobile nodes), in its simplest form, is the matrix multiplication $\mathtt{OP} = \mathtt{CV}^{-1} \cdot \mathtt{CB}$, where each row of $\mathtt{CB}$ represents a coded data block and each row of $\mathtt{CV}$ represents the coding vectors accomplished with it. The successful recovery of the original packets $\mathtt{OP}$ requires that the matrix $\mathtt{CV}$ be of full rank, i.e., the destination must collect k independent CPs.

Each mobile node can either download packets from an infostation or exchange packets with neighboring mobile nodes. In a non-cooperative network without any incentive mechanisms, the former is the only mechanism for packet dissemination. It only uses the high-speed channel between an infostation and a node near it, while

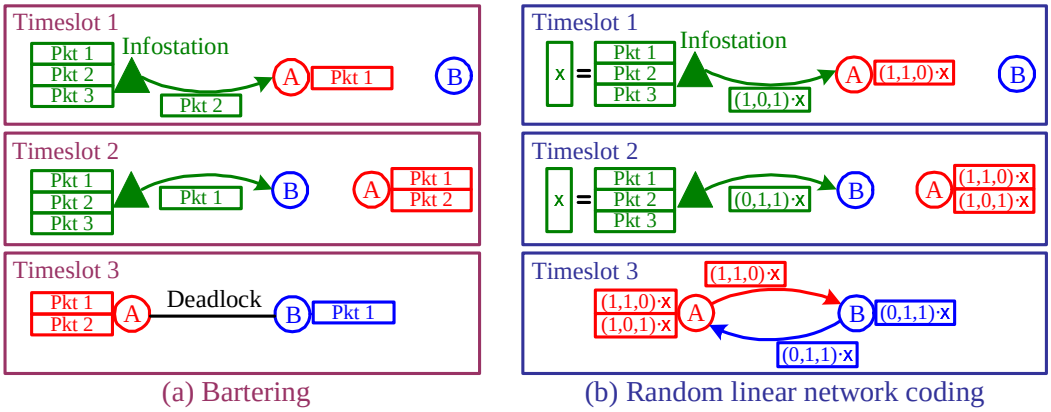


Figure 5-3. A comparison between bartering (without coding) and our **C4**.

wasting all the equally excellent channels between neighboring mobile nodes. A barter-like scheme (without network coding) alleviates this problem as follows: when two mobile nodes contact, they inspect the packet contents of each other. If each node identifies a packet that it wants, a bilateral packet exchange takes place. However, even for broadcasts, nodes can easily end up in a deadlock situation because of the requirement of mutual wants. In our **C4**, let $S_u(t)$ and $S_v(t)$ denote the subspaces spanned by the CPs at neighboring node u and v, respectively, at the beginning of the timeslot t. If $S_u(t) \nsubseteq S_v(t)$, we say node v wants CPs at u. If there exists mutual wants, then node u and v can successfully exchange CPs as follows: node u sends a new CP $p_u \in S_u(t)$ and $p_u \notin S_v(t)$, and node v sends back a new CP $p_v \in S_v(t)$ and $p_v \notin S_u(t)$ as return.

The design of **C4** relies on two unique features of RLNC:

• RLNC can greatly improve the possibility of successful exchange. As proved in Lemma 2.1 in [33], when RLNC is used, the probability that two randomly selected nodes have the mutual wants tends to 1 when q, the size of $\mathbb{GF}_q$, is large enough (e.g., $q = 2^8$). Figure 5-3 illustrates this effect. Suppose there are only two mobile nodes, and three OPs need be broadcasted at the infostation. It is easy for two nodes run into a deadlock situation as exemplified in Figure 5-3 (a). When virtual currency is introduced, node B can buy packet 2 from node A by sending back an e-cash at timeslot 3. However

only one data packet is transmitted, and e-cash transfer is overhead. By mixing the OPs, because with high probability each CP brings some new information, it can be treated as virtual commodity currency, i.e., generally acceptable in payment. For example, in Figure 5-3 (b), at timeslot 3, node B can buy a packet from A by paying a CP $(0,1,1)\cdot \mathtt{X}$. After timeslot 3, node A can decode all OPs, and node B only need one more CP. High exchange efficiency achieves without incurring any overhead.

• RLNC can serve as a free cipher. Before the destination obtains enough independent CPs, it cannot recover all OPs, even though every CP contains some information about every OP. Therefore, RLNC can serve as a free cipher for a time period. This functionality is extremely useful for supporting unicasts. Also take Figure 5-3 (b) as an example. Now suppose three OPs are destined for different mobile nodes (A or B). Before node A receives three independent CPs, he cannot tell which packet is destined to whom. Therefore, even when three OPs are all destined to node B, node A still has incentive to participate in exchanges before he collects three independent CPs. In this way, we enforce node A to help relay one CP to node B at timeslot 3, which is impossible for bartering.

5.3.3 Implementation Details of Our C4

According to source-destination relationship within the wireless domain, all original data packets in our **C4** can be classified into two categories: (1) download data packets: the packets from the infostation to mobile nodes; and (2) upload data packets: the packets from mobile nodes to the infostation.

Our **C4** take the whole network layer packet as an original data packet (including packet header). Each original data packet has a unique packet ID, and a session ID indicating which session this packet belongs to. A mobile node can tell whether he/she is interested in it by checking its session ID. In our **C4**, every transmitted packet within the wireless domain is a coded packet, and has a special format. At the infostation, some download data packets are selected and coded to produce virtual commodity

currency (VCC). We call these coded data packets as VCC packets. As illustrated in Figure 5-4 (a), a coded packet consists of a header and a body. The body stores the coded data block. For a VCC packet, the header consists of three parts: (1) TPE field, which is set to 11 to indicate a VCC packet; (2) session field, which indicates session IDs involved in the coded data block in the body part, and it can be implemented by a Bloom filter; (3) coding-vector field, where k is the generation size, p is the number of OPs encoded in the packet and c_i is the coding coefficient related to the OP with packet ID oid_i for $1 \leq i \leq p$. Unlike traditional RLNC packet, here $k \neq p$. The reason will be explained a little later. A VCC packet $\mathtt{CP}_{VCC}$ is valid for a mobile node u, only if u is interested in at least one session involved in $\mathtt{CP}_{VCC}$ and does no collect enough coded packets to recover all OPs of this session. Download/upload data packets will not be directly transmitted in our **C4**. They will be carried by VCC packets. For a download data packet with packet ID oid_q and destination node ID d_id, which is not selected to produce VCC packet, the infostation first selects one $\mathtt{CP}_{VCC}$ as the carrier, and then combines these two packets as shown in Figure 5-4 (b). The TPE field is set to 01 to indicate a coded download packet. The coded block is the linear combination of the coded block of $\mathtt{CP}_{VCC}$ with the coding coefficient 1 and the download data packet with the coding coefficient c_q. At a mobile node with node ID s_id, the coded upload packet is constructed in a similar way, as illustrated in Figure 5-4 (c). The TPE field is set to 00 to indicate a coded upload packet and the source node ID s_id is also included in the packet header.

There are only two kinds of elementary interactions when two mobile nodes u and v contact in our **C4**: (i) exchanges of coded packets: When node u has a coded packet which v is interested in, and the same thing happened for node v, then u and v have incentives to exchange these packets. (ii) exchanges of relay service and VCC packets: When node u wants node v to relay an upload/download data packet, and node u has at least two valid VCC packets for v, then in this interaction, node u sends two packets to

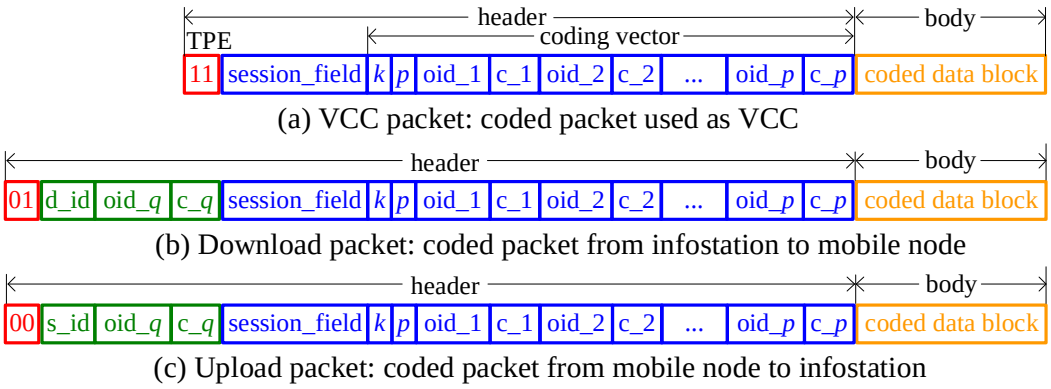


Figure 5-4. Packet format in our **C4**.

v: the coded upload/download packet needs to be relayed and one valid VCC packet as the payment. The upload/download data packet here should be carried by another valid VCC packet for v. Note that if the upload/download data packet is directly sent to v, then node v can drop this packet a little later without being detected. By combining the data packet with the VCC packet, our **C4** provides incentives for relay node v to keep this data packet before v can decouple it from the VCC packet. By decomposing all possible end-to-end communications into a series of elementary interactions, our **C4** puts no constraints on the underlying routing protocols or traffic patterns, and therefore achieves the maximum flexibility in supporting different application scenarios in MWNs.

The coding strategy adopted by infostations to generate VCC packets determines the performance of our **C4**. The key point is how to select sessions that need be mixed in generating VCC packets. There exists a tradeoff: if we mix the sessions interested by different mobile nodes, we can obtain VCC packets which are valid for a larger population of mobile nodes. Then the VCC packets can achieve a better exchange efficiency. However, the useful data information contained in the VCC packet for a mobile user is also decreased, which means the overhead of using VCC packets becomes larger. Therefore, a cautious choice should be made in order to obtain an optimized balance. A quantitative analysis of optimized coding strategies will be presented in Section 5.4 in detail. Here, we just give two qualitative guidelines. (i)

Because all users are interested in broadcast data packets, they can be safely mixed without worrying about overhead. The only problem is that in order to prolong the valid time of VCC packets, the size of one generation, i.e. k, should be large enough. For traditional RLNC, this means the size of the packet header is pretty large, because the size of coding vector which should be included in the coded packet header is $\Theta(k)$. To avoid this problem, our **C4** utilizes sparse RLNC proposed in [125] to control overhead. Given k original data packets as one generation, sparse RLNC first randomly chooses p data packets, then performs RLNC on these p packets to generate a coded packet. It can be shown that the successful exchange probability is on the same order of that of RLNC, while the size of the packet header is $\Theta(p)$ where $p \ll k$. Note that our packet format shown in Figure 5-4 already includes the unique feature of sparse RLNC by embedding k, p and OP IDs into coding-vector field. (*ii*) When only unicast traffics exist in a MWN, we have to mix OPs for different users to generate valid VCC packets. When k OPs for different sessions are grouped for one generation and coded with sparse RLNC, the session field in a VCC packet is determined by k OPs in the generation, not by p OPs selected for this VCC packet. However, it is possible that a VCC packet does not contain any useful data information for a mobile user, but that user still treats it as valid. A mobile node even cannot find this before he/she can recover all original packets. Therefore, the infostation can intentionally do this to provide incentives or optimize the whole system's performance. Our **C4** only requires the long-term fairness among mobile nodes and utilizes the free cipher functionality provided by RLNC to enforce this kind of operations.

5.4 Performance Analysis of Our C4

5.4.1 Network Model for Performance Analysis

In order to make the quantitative study possible, we detail our network model proposed in Section 7.3.1 as follows.

[Cell Model] Consider a square geography of area A_n with a fixed infostation at the center, as shown in Figure 6-1 (b). We assume the geography wraps around each boundary, effectively creating a torus. We refer to this torus as a cell. A cell is intended to mimic a typical multi-infostation network in which an infinite grid of infostations populate an infinite plane. The area A_n relative to the single infostation serves to characterize the density of fixed infostations over the terrain.

[Mobility Model] The cell is populated with n mobile nodes with independent mobility processes as follows. We first divide the cell into $m = \Theta(n)$ squarelets of area s_n^2 each, resulting in a two-dimensional $\sqrt{m} \times \sqrt{m}$ discrete torus (we assume $\sqrt{m}$ is an integer). We assume the time is slotted and each node independently performs a simple random walk on the two-dimensional $\sqrt{m} \times \sqrt{m}$ discrete torus, i.e., let a node be in squarelet s at timeslot t, then, at timeslot $t+1$, the node is equally likely to be in the same squarelet s or any of the four adjacent squarelets (i.e., up, down, left, and right squarelets).

[Protocol Model] For characterizing the condition for a successful wireless transmission, we adopt the protocol model proposed in [60]. We assume that all mobile nodes (including the infostation) use a common range r_n for their short-range transmissions, and a transmission from node i to node j is successful if and only if $d_{ij} \leq r_n$ and $d_{kj} \geq (1+\Delta) r_n$ for any other simultaneous transmitter, say node k. Here, d_{ij} is the distance between nodes i and j, and Δ is a positive constant independent of n. We assume that when two nodes are in the same squarelet, they are neighbors, i.e., they can establish a short-range wireless link. Therefore, $r_n = \sqrt{2} s_n$.

[Transmission Scheduling] A squarelet is called active at timeslot t, if the MAC scheduling scheme allows one node in that squarelet to transmit at timeslot t. Based on protocol model, we can guarantee that there exists an interference-free schedule such that each squarelet becomes active regularly once in L timeslots and it does not interfere with any other simultaneous active squarelets. Here L depends only on Δ, and

is independent of n (cf. Proposition 1 in our previous work [171]). The shaded squarelets in Figure 6-1 (b) illustrate an example of a group of simultaneous active squarelets when $L = 9$. For an active squarelet, at most one node pair will be scheduled to communicate. When the squarelet with the infostation (i.e., central squarelet) is active, the infostation will be one party of that node pair and another party will be randomly chosen from all nodes in that squarelet. For any other squarelet (i.e., regular squarelet), when it is active, a node pair is randomly selected from all possible node pairs in that squarelet. This is intended to mimic the behaviors of IEEE 802.11-like MAC protocol. For each timeslot, we assume only two packets can be transmitted, i.e., just enough for one successful elementary interaction.

We note that above model is the standard network model widely used in the literature for MWN performance analysis (refer to [58, 60, 171] and references therein), and its behaviors are characterized by the following Lemma when $n \to \infty$.

Lemma 8. *(1) Given $r_n = \sqrt{2A_n/m} = \Theta(\sqrt{A_n/n})$, the network formed by short-range wireless links is disconnected, i.e., there does not exist a contemporaneous path between two random selected nodes with high probability. (2) For a given timeslot, the probability that a mobile node is in the active central squarelet and selected into the communication pair is $p_l = \frac{\gamma}{m \cdot L} = \Theta(1/n)$, where γ is a constant. (3) For a given timeslot, the probability that a mobile node is in an active regular squarelet and selected into the communication pair is μ, and μ is a constant, i.e., independent of n.*

5.4.2 Performance Analysis for Broadcast and Multicast Traffics

We first consider broadcast scenarios. Suppose the infostation has K data packets to be distributed to n mobile nodes. For this scenario, in the central squarelet the elementary interaction is for the infostation to send two packets to a mobile node, while in regular squarelets the elementary interaction is packet exchanges between two mobile nodes.

The effectiveness of incentive mechanism IM is measured by the expected packet delivery time, i.e., $T_D(IM)$. For broadcast scenarios, let T_{IM} denote the time for all mobile nodes to obtain K data packets under the mechanism IM, then $T_D(IM) = \frac{\mathbb{E}[T_{IM}]}{K}$. The smaller the time $T_D(IM)$, the more effective the mechanism IM. The time $T_D(IM)$ for different mechanism IM is characterized by the following Theorem:

Theorem 5.1. ***[$T_D(IM)$ for Broadcasting K Data Packets]*** *When all nodes are fully cooperative and no network coding scheme is used (i.e., $IM = C$), then $T_D(C) = \Theta\left(\log n + \frac{n}{K}\right)$. When all nodes are non-cooperative, and (1) if bartering is used (i.e., $IM = B$), then $T_D(B) = \Theta\left(\log^2 n + \frac{n}{K}\right)$; (2) if virtual currency is used (i.e., $IM = VC$), then $T_D(VC) = \Theta\left(\log n + \frac{n}{K}\right)$; (3) if our* **C4** *is used with $k = K$ and p is a constant (i.e., $IM =$* **C4***), then $T_D(\mathbf{C4}) = \Theta\left(1 + \frac{n}{K}\right)$.*

Proof. We observe that for a given timeslot, whether a mobile node can obtain a new packet depends on two conditions: (i) Whether this node is selected into a communication pair? This condition is characterized by Lemma 8 and is independent of the incentive mechanism used. (ii) When this node is selected into a communication pair, whether it can make a successful exchange with its partner? Here, we assume that for an arbitrary node pair, there exists a probability p_E of successful exchange which only depends on the incentive mechanism. We want a static, summarized characterization of p_E, i.e., we obtain a value of p_E which is averaged over all possible node pairs and all timeslots. We then simplify our analysis by assuming that given the incentive mechanism, the probability of successful exchange for any node pair at any timeslot is the same as this value. It is not hard to see that this key assumption is inconsistent with the interaction dynamics. Nevertheless, our simulation results agree closely with the analytical results, indicating that this assumption works well in systems with moderately large number of nodes $n \geq 50$.

Based on this assumption, we can model the dynamics of obtaining packets for a given mobile node as a discrete time Markov chain illustrated in Figure 5-5. Denote the

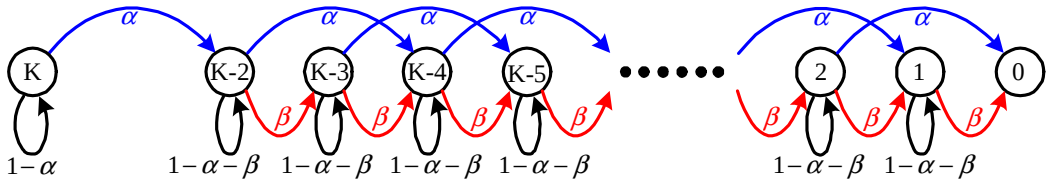


Figure 5-5. State transition diagram for obtaining packets of a mobile node.

state as the number of packets remaining to be obtained by a mobile node. Initially a node is at state K. Since the first two packets must be obtained from an infostation, the next state is $K-2$. Subsequently, in states $s \in \{1, \cdots, K-2\}$, there are three possibilities in each timeslot:

- With probability α, the node is in the active central squarelet and obtains two packets from the infostation. The state goes from s to $s-2$.
- With probability β, the node is in an active regular squarelet and obtains one packet from another mobile node. The state goes from s to $s-1$.
- With probability $1-\alpha-\beta$ no new packets are obtained because the node is not selected into communication pairs or because the node cannot make a successful exchange with its partner and the state stays the same.

For α, because the infostation always has the incentive to send packets to mobile nodes, this probability is independent of the incentive mechanism and from Lemma 8(2) we know that $\alpha = p_I = \Theta(1/n)$. For β, obviously $\beta = \mu \cdot p_E = \Theta(p_E)$.

We denote the expected first passage time from state i to state 0 as PT_i, where $2 \leq i \leq K-2$. Conditioning on the next state transition and rearranging yields the difference equation,

$$PT_i = \frac{1}{\alpha+\beta} + \frac{\beta}{\alpha+\beta} PT_{i-1} + \frac{\alpha}{\alpha+\beta} PT_{i-2}$$

where the boundary conditions are given by $PT_0 = 0$ and $PT_1 = \frac{1}{\alpha+\beta}$. Using z-transforms, we can obtain

$$PT_i = \frac{i(\beta+2\alpha) + \left(1 - \left(\frac{-\alpha}{\alpha+\beta}\right)^i\right)\alpha}{(\beta+2\alpha)^2}$$

It is obvious that $\mathbb{E}[T_{IM}] = 1/\alpha + PT_{K-2}$, where $1/\alpha$ is the expected time until a node first obtains two packets from the infostation. When $n \to \infty$, we can obtain

$$\mathbb{E}[T_{IM}] \to \frac{K}{\beta} + \frac{1}{p_I} = \frac{K}{\mu \cdot p_E} + \frac{1}{p_I}.$$

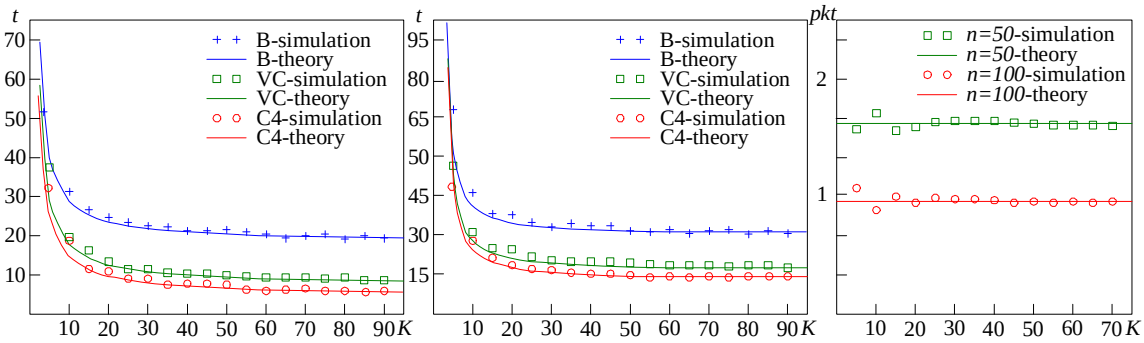
When all mobile nodes are fully cooperative and no network coding techniques is used, p_E is the probability that a node has a new packet for another node. Using the result about coupon collecting problem [33], we have $p_E = \Theta(1/\log n)$. Therefore, $T_D(C) = \Theta\left(\log n + \frac{n}{K}\right)$.

When mobile nodes are non-cooperative, we have three possibilities: (1) For bartering, p_E is the probability that two nodes can provide a new packet for each other, thus $p_E = \Theta\left(1/\log^2 n\right)$ and $T_D(B) = \Theta\left(\log^2 n + \frac{n}{K}\right)$. (2) For VC, we assume each node has enough VC. Therefore $T_D(VC)$ is the same as $T_D(C)$ in cooperative case. (3) For our **C4**, using the property of RLNC [67], we have $p_E \geq (1 - 1/q)^2 = \Theta(1)$. Therefore, $T_D(\mathbf{C4}) = \Theta\left(1 + \frac{n}{K}\right)$. □

The cost efficiency of incentive mechanism *IM* is measured by the expected number of control packets (i.e., $C_P(IM)$) that need be transmitted for obtaining one data packet. For broadcast scenarios, let C_{IM} denote the expected total number of control packets transmitted for one mobile node to obtain K data packets under the mechanism *IM*, then $C_P(IM) = \frac{\mathbb{E}[C_{IM}]}{K \cdot n}$. The smaller the number $C_P(IM)$, the more cost-efficient the mechanism *IM*. The number $C_P(IM)$ for different *IM* is characterized by the following Theorem:

Theorem 5.2. ***[$C_P(IM)$ for Broadcasting K Data Packets]*** *(1) $C_P(C) = C_P(B) = C_P(\mathbf{C4}) = 0$ and (2) $C_P(VC) = \Theta\left(1 - \frac{\log n}{n}\right)$. (The proof is omitted for the reason of space.)*

In Figure 5-6, we compare theoretical results obtained from Theorem 5.1 and 5.2 with simulation results when $L = 4$. We show $T_D(IM)$ and $C_P(IM)$ averaged over 100 simulation runs for different *IM*s. The number of nodes is held constant at $n = 50$ ($m = 25$) for Figure 5-6 (a) and at $n = 100$ ($m = 64$) for Figure 5-6 (b), while the number of data



(a) T_D (*IM*) with n=50 (b) T_D (*IM*) with n=100 (c) C_P (*VC*)

Figure 5-6. $T_D(IM)$ and $C_P(IM)$ as functions of K.

packets K is varied. Figure 5-6 (c) shows $C_P(VC)$ for $n = 50, 100$. The results about $C_P(C)$, $C_P(B)$, and C_P (**C4**) are all zeros, and are not shown in the figure. In all cases, the differences between the simulation results and our theoretical results are found to be very small.

From Theorem 5.1 and 5.2 and simulation results, we can make the following observations about broadcast scenarios:

• For all possible values of K, the packet delivery time required by our **C4** is always smaller than that required by other possible incentive mechanisms. Our **C4** is even better than the case when all nodes are cooperative but no coding is used. All these improvements are obtained with zero cost. Therefore, our **C4** is the best choice for broadcast traffics.

• We assume the number of data packets which need be broadcasted is pretty large, i.e., $K = \Omega(n)$. In this regime, from Theorem 5.1, we have $T_D(B) = \Theta\left(\log^2 n\right)$, $T_D(VC) = \Theta\left(\log n\right)$, and $T_D(\mathbf{C4}) = \Theta(1)$. The improvements converge to a factor on the order of $\log n$ which is independent of K.

• It is believed in previous work [19, 51, 115, 142, 167] that for broadcast traffic, bartering is better than virtual-currency based schemes. Our results also provide an analytical evidence to support this belief because compared to bartering, the VC based

scheme only provides an $\Theta(\log n)$ improvement to the packet delivery time, with a cost of $\Theta(n)$ control packets per node.

We then consider multicast scenarios. The following corollary shows that the same performance can be achieved.

Corollary 5.1. *Suppose the infostation has K data packets to send to $\epsilon \cdot n$ randomly selected mobile nodes. The values of $T_D(IM)$ and $C_P(IM)$ remain on the same order as that in Theorem 5.1 and 5.2 when ϵ is a constant.*

5.4.3 Performance Analysis for Pure Unicast Traffics

Based on above discussions, we conclude that when broadcast or multicast traffics exist in the network, all of them should be utilized to generate VCC packets by performing intra-session network coding. VCC packets generated in this way can facilitate packet/service exchanges without incurring any cost. However, the situation will be complicated for scenarios with pure unicasts.

In this subsection, we assume there exist at least n unicat sessions, each of them is destined to one mobile node. We divide these unicast sessions into g groups ($1 \leq g \leq n$), each group (called coding group) containing n/g distinct sessions. Only the data packets destined to mobile nodes in the same group will be mixed to generate VCC packets, i.e., we perform group-based inter-session network coding (with grouping parameter g) to generate VCC packets. There may exist other unicast traffics, but they will not be involved in generating VCC packets. For pure unicast scenarios, $T_D(IM)$ is the expected delivery time for unicasting one data packet from the infostation to a randomly selected mobile node (or in a reverse direction) under the incentive mechanism IM, and $C_P(IM)$ is the expected total number of control packets needed to support achieving $T_D(IM)$ for that data packet.

When a mobile node receives a valid VCC packet, in broadcast (or mixed traffic) scenarios this VCC packet incurs no control overhead. However, in pure unicast scenarios, this VCC packet only contains g/n useful information on average. Therefore,

this VCC packet incurs $(1 - g/n)$ control overhead, i.e., there is no VCC packet without incurring any cost. This is the key feature emerging in pure unicast scenarios.

Another unique feature for pure unicast scenarios is about routing. From Lemma 8 (3), we know that the network is disconnected, and because every node performs a random walk, there is no way to predict contacts. Therefore, the source or destination node cannot provide incentives to intermediate nodes except the direct next hop. As a consequence, we can only provide incentives to sustain two-hop relay [58, 171], i.e., every packet can at most take two hops from the source to the destination. In Section 5.5, we will remove this constraint by considering human mobility traces from the real world.

We first characterize performance of traditional approaches:

Theorem 5.3. ***[Performance of Bartering and VC for Pure Unicasts]*** *For pure unicast traffics, (1)* $T_D(B) = \Theta(n)$ *and* $C_P(B) = 0$*; and (2)* $T_D(VC) = \Theta\left(\sqrt{n}\right)$ *and* $C_P(VC) = \Theta\left(\sqrt{n}\right)$*. (The proof is omitted here for the reason of space.)*

It has already been shown in [171] that when all nodes are fully cooperative, $T_D(C) = \Omega\left(\sqrt{n}\right)$ for unicasts. From Theorem 5.3 (2), we can see that VC based scheme can achieve the lower bound of $T_D(C)$ with large cost. This is because in order to obtain $\Theta\left(\sqrt{n}\right)$ packet delivery delay, it takes $\Theta\left(\sqrt{n}\right)$ nodes to provide packet relay services. For bartering, because there is no mutual wants between two mobile nodes at all, the only way for a mobile node to receive a packet is to contact the infostation, which causes a large delay on the order of n.

For our **C4**, by taking different values of g, i.e., the grouping parameter, **C4** in fact can provide a series of incentive mechanisms with different network performance as follows.

Theorem 5.4. ***[Performance of Our* C4 *for Pure Unicasts]*** *For pure unicast traffics, (1) when* $g = O\left(\sqrt{n}\right)$*,* $T_D(\mathbf{C4}) = \Theta\left(\sqrt{n}\right)$ *and* $C_P(\mathbf{C4}) = \Theta\left(\sqrt{n} - \frac{g}{\sqrt{n}}\right)$*; (2) when* $g = \Omega\left(\sqrt{n}\right)$*,* $T_D(\mathbf{C4}) = \Theta(g)$ *and* $C_P(\mathbf{C4}) = \Theta\left(\frac{n}{g} - 1\right)$*.*

Proof. (1) when $g = O\left(\sqrt{n}\right)$, from our previous work [171] we know that asking $\Theta\left(\sqrt{n}\right)$ mobile nodes to act as relay nodes is enough to achieve the lower bound of packet delivery delay on the order of $\sqrt{n}$. Asking more nodes to help only increases the cost of our **C4** and cannot further decrease the packet delivery delay. Therefore, when $g = O\left(\sqrt{n}\right)$, the infostation only randomly selects $\Theta\left(\sqrt{n}\right)$ mobile nodes from the same group to act as relay nodes. From our previous work [171], we know that in this situation $T_D\left(\mathbf{C4}\right) = \Theta\left(\sqrt{n}\right)$. Because each coded packet only contains g/n useful information for a particular mobile node, each coded packet incurs $\left(1 - \frac{g}{n}\right)$ control overhead. On average, each mobile node need exchange one coded packet with every relay node, and totally $\Theta\left(\sqrt{n}\right)$ coded packets need be exchanged for the destination node successfully decoding one data packet. Therefore, $C_P\left(\mathbf{C4}\right) = \Theta\left(\left(1 - \frac{g}{n}\right)\sqrt{n}\right)$.

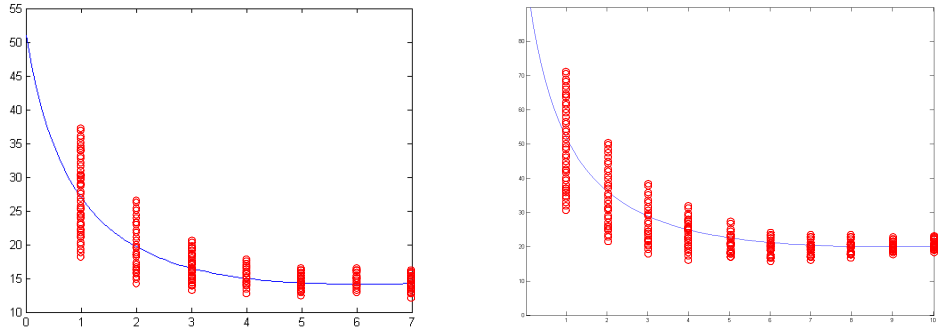
(2) when $g = \Omega\left(\sqrt{n}\right)$, the infostation utilizes n/g mobile nodes (i.e., all nodes in the same group) to act as relay nodes. From the proof of Theorem 5.1 we know that in order to successfully decode one original data packet, on average the destination node need wait $\Theta\left(g\right)$ timeslots, and this is exactly the $T_D\left(\mathbf{C4}\right)$. Because each coded packet only contains g/n useful information for a particular mobile node, each coded packet introduces $\left(1 - \frac{g}{n}\right)$ control overhead. On average, each mobile node need exchange one coded packet with every relay node, and totally $\Theta\left(n/g\right)$ coded packets need be exchanged for the destination node successfully decoding one data packet. Therefore, $C_P\left(\mathbf{C4}\right) = \Theta\left(\left(1 - \frac{g}{n}\right)\frac{n}{g}\right)$. □

From Theorem 5.4, we directly obtain the following corollary.

Corollary 5.2. *The effectiveness-cost tradeoff of our* **C4** *for pure unicasts when* $C_P(\mathbf{C4}) = O\left(\sqrt{n}\right)$ *is given by*

$$T_D(\mathbf{C4}) = \Theta\left(\frac{n}{C_P(\mathbf{C4}) + 1}\right).$$

Figure 5-7 illustrate above effectiveness-cost tradeoff of our **C4** for pure unicasts. The solid lines here are theoretical results while points represent $\left(C_P(\mathbf{C4}), T_D(\mathbf{C4})\right)$



(a) T_D(**C4**) and C_P(**C4**) curve with n=50 (b) T_D(**C4**) and C_P(**C4**) curve with n=100

Figure 5-7. The effectiveness-cost tradeoffs of our **C4** for pure unicasts.

pairs obtained from simulations. For both $n=50$ and $n=100$, we can see that simulation results agree with our theoretical results, i.e., the effectiveness-cost tradeoff indeed exists.

From Theorem 5.4, Corollary 5.2 and simulation results, we can make the following observations about unicast traffics:

• When $C_P(\mathbf{C4}) = O\left(\sqrt{n}\right)$, $T_D(\mathbf{C4})$ is a strictly decreasing function of $C_P(\mathbf{C4})$, as shown in Corollary 5.2 and Figure 5-7. This means that there exists a fundamental tradeoff between packet delivery delay and cost. If we want to decrease packet delivery delay (i.e., increase incentive effectiveness), then the implementation cost of **C4** must be increased. By adjusting the grouping parameter, **C4** provides a systematic way to smoothly trade incentive effectiveness for implementation cost.

• In our **C4**'s tradeoff curve, when $g = n$, we obtain $C_P(\mathbf{C4})=0$ and $T_D(\mathbf{C4})=\Theta(n)$. This is the same as $(C_P(B), T_D(B))$ pair (cf. Theorem 5.3 (1)). We note that this is not coincident because when $g = n$, no inter-session network coding is performed and no valid VCC packets generated in **C4**. So it is equivalent to bartering cases.

• In our **C4**'s tradeoff curve, when $g = 1$, we obtain $C_P(\mathbf{C4}) = \Theta\left(\sqrt{n}\right)$ and $T_D(\mathbf{C4}) = \Theta\left(\sqrt{n}\right)$. This is the same as $(C_P(VC), T_D(VC))$ pair (cf. Theorem 5.3 (2)). We note that this is also not coincident because when $g = 1$, every VCC packet only

Table 5-4. Dataset properties

Dataset	Cambridge	MIT
Device	iMote	Phone
Network type	Bluetooth	Bluetooth
Duration (days)	11	246
Granularity (sec.)	600	300
# of devices	54	97
# of contacts	10,873	54,667
Data source	www.haggleproject.org	reality.media.mit.edu

contains $1/n$ useful information, which tends to 0 when n is pretty large. Therefore, in this case, VCC packet is equivalent to e-cash packet.

5.5 Improving Our C4's Performance with Social Contact Information

In previous section, there are two problems left for our **C4** for pure unicast scenarios, i.e., (1) two-hop relay constraints and (2) unavoidable effectiveness-cost tradeoffs. We note that these two problems are not caused by our **C4**, instead they are the consequence of oversimplified and unrealistic mobility model used for performance analysis. In this section we address these two problems by considering unique features in real-world user mobility patterns. This section also serves to validate our **C4**'s performance with human mobility traces.

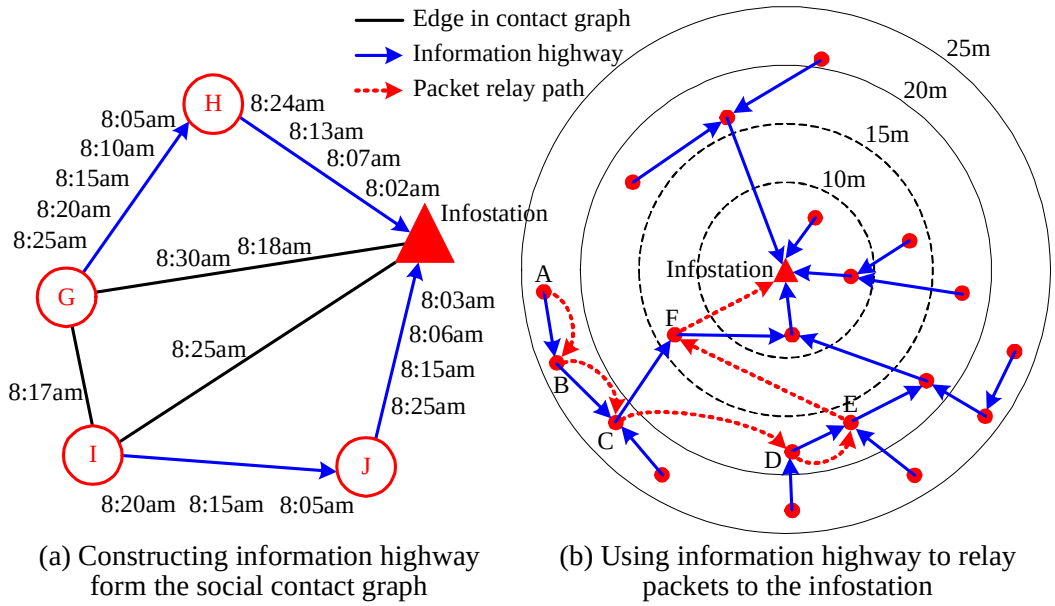
In this section, we use two experimental datasets gathered by the Haggle Project (referred to as Cambridge) and the MIT Reality Mining Project (referred to as MIT). The characteristics of these datasets are summarized in Table 5-4. We first convert all these real life data into social contact graphs, with devices/mobile users as node sets and contacts between two nodes as edge sets. From these social contact graphs, we can easily find that they are heterogeneous both in terms of edges and community structures. Note that from the random walk mobility model, we can only obtain a homogeneous contact graph, i.e., a complete graph with equal edges. There are no link heterogeneity or community structures because every node has the same probability to contact every other node. In what follows, we show how to utilize these two kinds of heterogeneities in reality to facilitate our **C4** design. Here we do not mean our **C4** can

work only if the information about social contact graph is available. We only want to show that **C4** can even work better if this information is available. In fact, all DTN routing schemes try to collect and utilize this information [178], and our **C4** does not require anything more.

5.5.1 Information Highway and Multi-hop Relay

The first kind of heterogeneity is called edge heterogeneity [88]. As shown in Figure 5-8 (a), edges are annotated with one or more times at which two nodes contact. We can see that not all edges in the social contact graph are with the same importance. For example, we assume that the infostation updates its status information every timeslot, and every mobile node exchanges its newest status information about the infostation with others during contacts. Then, we can observe from Figure 5-8 (a) that, although node G can contact the infostation, for most of timeslots, node G obtains the newest status information about the infostation from node H. So, given the existence of the information propagating highway "G−H−infostation", edge "G−infostation" can be deleted from the social contact graph. For the same reason, edge "I−infostation" and edge "G−I" can also be deleted. By continuing this deleting procedure, at last, we will obtain a tree rooted at the infostation. This tree is called the information highway, i.e., the structure of fast indirect paths from the infostation to all mobile nodes [88]. By assuming contacts between pairs of nodes are perfectly periodic, we can calculate the delay of the newest status updates from the infostation to every mobile node along this tree. The result is Figure 5-8 (b), the information highway with every node knows the latency value from the infostation to itself. The concentric circles denote ball radii increasing by 5 minutes, and the distance of each node from the common center is its latency value from the infostation.

For every node u, its latency value $d_T(u)$ is a kind of temporal distance between itself and the infostation, and therefore can be used as a routing metric. All nodes have incentives to maintain a correct $d_T(u)$ for securing its own packet delivery and



(a) Constructing information highway form the social contact graph

(b) Using information highway to relay packets to the infostation

Figure 5-8. Information highway and multi-hop relay.

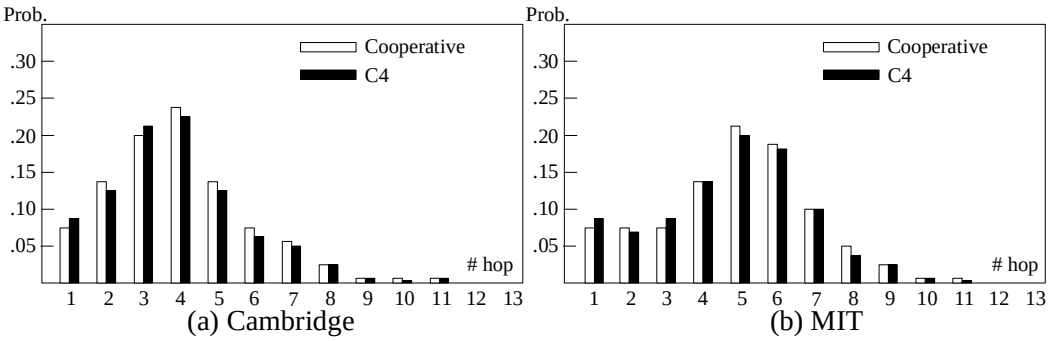


Figure 5-9. The distribution of hop counts.

the information highway can be constructed in a fully distributed way. For unicasts from mobile nodes to the infostation, multi-hop relay can be performed in the following way. When a relay node u contacts a node v, and v has a smaller $d_T(v)$, then packets destined to the infostation will be relayed to v. In Figure 5-8 (b), one of such relay path from node A to the infostation is illustrated by red curves. The unicasts from the infostation to the mobile nodes will be even simpler. The routing path will follow the tree of information highway, because by the definition of information highway, this is the path with the minimum packet delivery delay from the infostation to the destination.

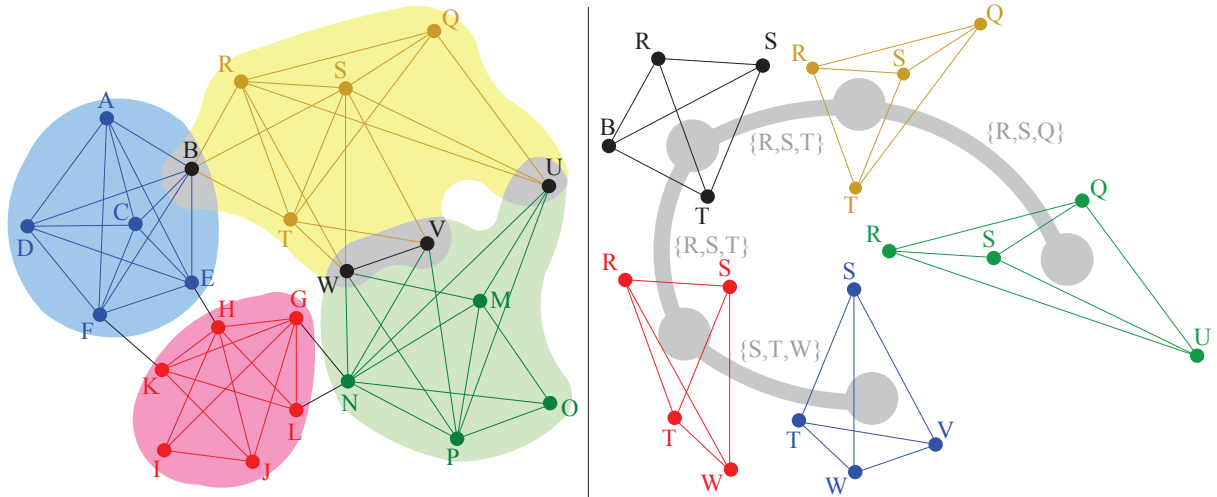
The only problem left here is why our **C4** can sustain two-more-hop relay when the information highway emerges. For those intermediate relay nodes who do not contact with the source or the destination of packet pkt, the incentives for them to buy pkt come from the fact the the nodes which are closer to the destination will have more incentives to buy it. Therefore, they can sell it a little later and from this buy-sell procedure they can obtain more VCC packets. For example, in Figure 5-8 (b), on the path from node A to the infostation, node E will buy pkt from node D because he knows some nodes like E has the willingness to buy pkt, because E is closer to the infostation and has more chances to sell pkt to the infostation.

Figure 5-9 shows how many hops the packets take to reach the destination. We can see that for both datasets, the hop count distributions of our **C4** are the same as that of the cases when all nodes are fully cooperative. Therefore, we can conclude that our **C4** provides adequate incentives to sustain multi-hop relay when the information highway is available.

5.5.2 Community Structure and Grouping Parameter Selection

The second kind of heterogeneity is called community structures. Social contact graphs typically contain parts in which the nodes are more highly connected to each other than to the rest of the graph. The set of such nodes is usually called a community [126]. Figure 5-10 (a) illustrates four communities in an exemplary social contact graph, each with a different color. We can observe that (1) a typical member in a community is linked to many other members, but not necessarily to all other members in the community, (2) different communities may overlap. In Figure 5-10 (a) overlapping parts are emphasized by grey color.

The use of community structures in our **C4** is straightforward. When there exists no broadcast or multicast traffics, we need to group some unicast sessions destined to different nodes to perform group based inter-session network coding to generate VCC packets. The way of grouping will significantly affect our **C4**'s performance as



(a) Communities in an exemplary contact graph (b) A set of 4-cliques in the yellow community in (a)

Figure 5-10. Community structures in the social contact graph.

discussed in Section 5.4.3. We know that if we decrease the group size (i.e., n/g) to reduce costs, the effectiveness of **C4** will also be reduced. This relationship is called effectiveness-cost tradeoffs characterized in Corollary 5.2. However, all these results are based on the contact graphs without community structures. For the example given in Figure 5-10 (a), we can directly observe that, when we group the yellow community and blue community separately, compared to grouping yellow and blue communities together, the effectiveness will not be affected. The reason is simple: most of nodes in the yellow community in fact have no chance to contact most of nodes in the blue community. Therefore, when the VCC packets are only valid for the yellow community, there is no efficiency loss. So, every community can has its own valid VCC packets, just like every country has its own currency. Compared to a global currency, the loss of efficiency is ignorable. Therefore, when we shrink the group to the community, we can reduce overhead without losing effectiveness. Based on above discussions, we conclude that the best choice of grouping should be based on community structures in the social contact graphs.

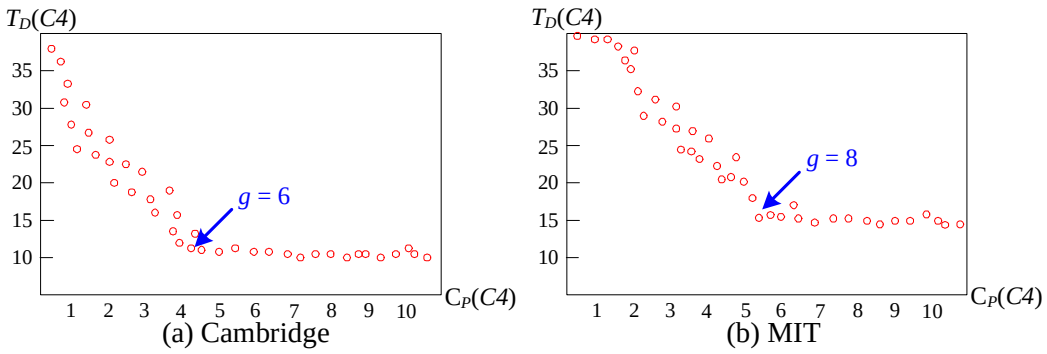


Figure 5-11. The effectiveness-cost tradeoffs under different g (or k) values.

The problem left is how to identify communities automatically. The difficulty comes from the unique features of community strictures in social graphs. Because a node in a community is not necessarily linked to all other nodes in the community, the community is not a clique (i.e., a complete subgraph). Different communities may overlap, and therefore traditional divisive and agglomerative methods cannot be applied [126] for this situation. Here, we use a technique called k-clique-communities proposed in [126]. A fully connected subgraph of k nodes is called a k-clique. we define a k-clique-community as the union of all k-cliques that can be reached from each other through a series of adjacent k-cliques, where two k-cliques are said to be adjacent if they share $k-1$ nodes. Figure 5-10 (a) illustrates overlapping 4-clique communities, and Figure 5-10 (b) shows the procedure of forming the yellow community in Figure 5-10 (a).

Obviously, the best choice of k here corresponds to the best choice of grouping parameter g. Here we develop a distributed scheme to find the optimal k and construct the corresponding k-clique-communities. We call the maximal complete subgraphs as cliques. In contrast to the k-cliques, cliques cannot be subsets of larger cliques, therefore they have to be located in a decreasing order of their size. The largest possible clique size in the graph is determined from the degree-sequence. Therefore, the infostation first finds the node with the highest degree in the social contact graph, then finds the clique containing that node. After recording this clique, the infostation deletes

the node and its edges in this clique from the social contact graph. Then the infostation repeats this procedure until no nodes are left. After that, the recorded set of cliques is a set of independent k-cliques in original social contact graph. Then the infostation chooses several largest k-cliques as seed groups and broadcast the node lists for each seed group. Other nodes can choose group memberships by themselves. The infostation will perform group-based inter-session network coding on packets from the same group to generate VCC packets. Since all mobile nodes have the incentives to choose the most appropriate groups for themselves, after several iterations, the best community structures will emerges by themselves and the optimal k (or g) value will be automatically determined.

Figure 5-11 shows all $(C_P(\mathbf{C4}), T_D(\mathbf{C4}))$ pairs obtained by the system itself. We can observe that there indeed exists an optimal value of g, which corresponds to the critical point at which when we further increase the cost $C_P(\mathbf{C4})$, the corresponding packet delivery delay $T_D(\mathbf{C4})$ cannot be decreased. This optimal value for Cambridge dataset is 6 while that for MIT dataset is 8. This means that, for example, the community structures of social contact graph of Cambridge dataset is most appropriately described by 6-clique-communities.

5.6 Chapter Summary

The benefits of utilizing network coding in wireless networks, such as improving throughput, reducing energy consumption, simplifying transmission scheduling, etc., have been extensively studied in the literature (cf. [33, 67, 171] and references therein). However, as far as we know, our **C4** is the first to demonstrate a new potential benefit brought by network coding for wireless networks, i.e., providing incentives for cooperation. Incentive problems in wireless networks with network coding have also been studied in [28, 176]. Both of them apply game theory to study coding strategies adopted by individual nodes and still use virtual currency (credit) to create incentives. In our **C4**, we assume that the main application of MWNs is to provide Internet access

services, and most traffic in a MWN will go through the infostation, which enforces random linear network coding (cf. Section 7.3.1) to improve the whole system's efficiency. Therefore, no room is left for self-interested mobile nodes to change coding strategies, and coded packets can be safely utilized to create incentives. By treating network coding as a tool to produce virtual commodity currency, our **C4** provides an effective and lightweight solution to induce cooperation, which is impossible for [28, 176] based on traditional virtual currency. How to extend our **C4** to pure wireless ad hoc networks will be an interesting and important topic and we will study it in our future work.

CHAPTER 6
TRUST-BASED ROUTING AND NON-CLASSICAL ROUTING ALGEBRA

6.1 Chapter Overview

A wireless ad hoc network (WANET) is a collection of wireless mobile nodes dynamically forming a temporary network without requiring any centralized authority or fixed network infrastructure [20, 71, 119, 131]. In WANETs, every node has two roles: it serves as both an independent end-system (terminal) and a router actively participating in packet forwarding for other nodes. Communications between two nodes can be performed directly if the destination is within the source's transmission range, or through intermediate nodes acting as routers (multi-hop transmission) if the destination is outside source's transmission range. It has been widely recognized that WANETs are among the most promising networking technologies to provide rapid, untethered access to information and computing, eliminating the barriers of distance, time, and location for many applications ranging from collaborative and distributed mobile computing to disaster recovery (such as fire, flood and earthquake), law enforcement (crowd control, search and rescue) and military communications (command, control, surveillance and reconnaissance) [119, 131]. However, due to their open, distributed and dynamic nature, WANETs are highly vulnerable to various (external or internal) malicious attacks [20, 71] and selfish behaviors of participating nodes. As routes in WANETs are composed of only mobile terminals based on a multi-hop mechanism, WANETs may encounter situations where intermediate nodes interrupt packet delivery adversely or selfishly. Accordingly, it is very important to realize robust, efficient and secure routing protocols in order to guarantee a large-scale deployment of WANETs [20, 132].

One important observation is that trust plays an unique and fundamental role in solving the routing security problems of WANETs, and therefore trust evaluation, trust propagation, and trustworthy path selection should be integrated into routing protocol design in order to enhance the routing security of WANETs [20, 54, 148].

- Trust underlies any security mechanisms. Each wireless node acts as a router and participates in the routing protocol. Routing relies therefore on explicit or implicit trust relationships among participating nodes. Only based on the existing trust relationships, can key managements be correctly performed and other cryptographic primitives be further developed for securing a WANET.

- Trust provides a uniform mechanism to defend malicious and selfish behaviors. It is well known that cryptographic techniques alone cannot cope with routing disruptions due to internal attacks and selfish behaviors. A network-wide trust/reputation management system can help in internal attacker detection and encourage the desired cooperative behaviors among individual nodes. When nodes join and leave a WANET dynamically, trust also provides a quantitative way for a node to join all other nodes into internal and external categories securely.

- Trust extends the design space of secure routing protocols. Instead of applying cryptographic primitives to secure existing routing protocols, we can actively utilize the trust as a metric for a source node to select the most trustworthy path to the destination. This leads to the so called secure-aware routing, which incorporates security requirements into the routing operations from the very beginning [20, 148].

To facilitate the implementation of this idea, various trust metrics, which quantify trust relationships according to different applications' security requirements, have been designed and integrated into routing metrics[1] in the security research community. For example, PGP-style authentication schemes with certification chains [23, 168] use a binary trust valuation (e.g., 1-or-0, all-or-none). Reputation-based schemes [124, 151] employ real numbers to measure the trustworthiness. In some evidence-based schemes [77, 152], a two-dimensional vector in $[0, 1]^2$ describes the trust opinion. In [166], trust measurement is even combined with other QoS requirements to act as the routing metric. Specific trust inference and trustworthy path selection algorithms are also designed for peer-to-peer networks, ad hoc networks, and sensor networks, which provide abundant choices for trust-based routing protocol design (cf. Section 5.2.2 for detailed discussions).

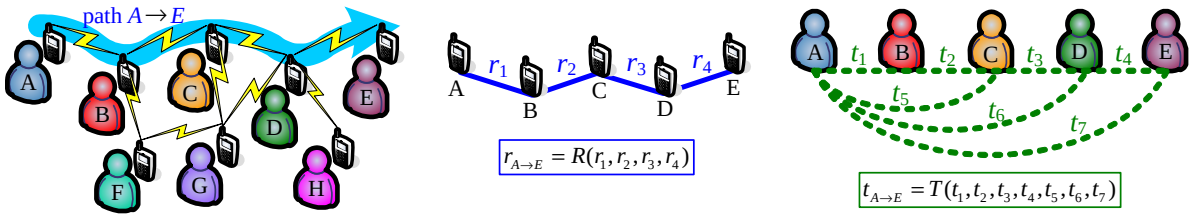
[1] When the trust metric is utilized as a (part of) routing metric, we call the latter as the trust-related routing metric, or just trust metric for short, and the corresponding routing protocol as trust-based routing.

While these application-specified trust metrics capture different characteristics of their target scenarios or trust relationships, there is a lack of understanding on the impact of trust metrics on the operations of routing protocols. One important lesson we have learned from Internet routing protocol design is that in general we cannot arbitrarily change one routing metric to another without considering the routing protocols used in the network. If routing metrics are unscrupulously combined with an incompatible routing protocol, the routing protocol may fail to find an optimal path or lead to routing loops [143, 145]. This principle is still applicable here, since the trust metric is just a special kind of routing metrics, and routing protocols in WANETs share many common features with their Internet counterparts [131, 164]. In fact, our case study shows that some trust metrics mentioned above lead to routing anomalies when they are combined with a Dijkstra-based routing algorithm (cf. Example 4 and 5 in Section 6.2 for detailed discussions). Therefore, without a good understanding on the interactions between trust metrics and routing protocols, the evaluation and design of trust-based routing are still at the empirical stage.

In the networking research community, a theoretical framework called routing algebra[2] has been developed for the study of the compatibility of routing metrics and routing protocols in the context of QoS routing [143] and BGP protocols [145] used in the Internet. It has also been extended and applied to multi-hop wireless networks recently [108, 164]. So why do not we just use the existing routing algebras mentioned above to study trust metrics (as a part or total routing metrics)?

The key point here is that trust metrics are significantly different from normal routing metrics such as the number of hops, data rates or other QoS requirements. In what

[2] Please note that the algebra here may not meet all conditions in the definition of algebra used in the mathematical literature, rather it represents an abstract algebraic structure.



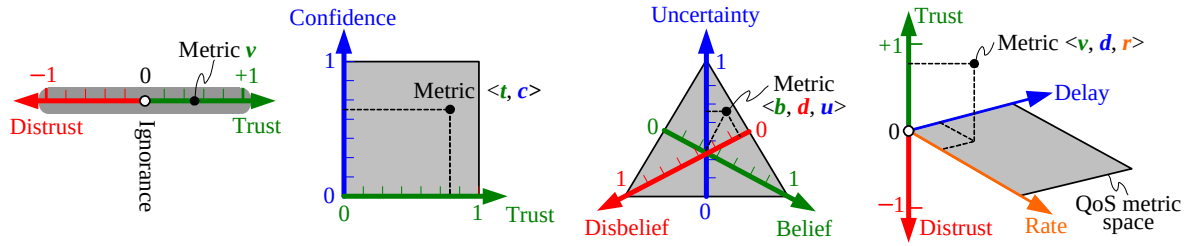
(a) An exemplary WANET (b) Physical graph for path $A \to E$ in (a) (c) Trust graph for path $A \to E$ in (a)

Figure 6-1. Physical graph and trust graph for an exemplary path from A to E. Here, solid lines and dashed lines represent wireless communication links and trust relationships, respectively. We can observe from this example that: for computing the normal routing metric (e.g., data rate) of path $A \to B \to C \to D \to E$ ($A \to E$ for short), i.e. $r_{A\to E}$, we only need 4 links information in physical graph (b); while for computing the trust-based routing metric of path $A \to E$, i.e., $t_{A\to E}$, we need at least 7 links information in trust graph (c).

follows, we identify five unique features of trust metrics, which make previous results unapplicable to trust-based routing. First of all, the topological structures related to trust metrics are more complicated. For normal routing metrics, one physical graph, which consists of edges representing physical links between individual nodes, is enough to describe the topological structure of the routing problem. For trust metrics, trust relationships among individual nodes form another topological structure called trust graph, which may have totally different structures compared with the physical graph (refer to Figure 6-1 for an illustration). Trust-based routing is restricted by the physical conditions as well as trust conditions, and therefore requires a new routing algebra built upon both graphs. Secondly, trust metrics have different algebraic properties compared with normal routing metrics. For example, in Example 3 in Section 6.2, we will show that trust metrics in general are non-distributive, while most of previous routing algebras assume that distributivity holds [27, 57, 108, 143–145, 164]. Since this property plays a key role in the analysis of normal routing metrics, related results cannot be applied here where it is not satisfied. Thirdly, for traditional routing metrics, the metric value of one physical link is independent of that of other links. For example, the bandwidth (or delay) of one link will not be affected by the bandwidth of another

link, if these two links do not interfere with each other. However, trust can be passed (propagated) between different users, which means the trust-based routing metrics of different physical links are dependent. Obviously, this dependency will complicate the analysis of trust-based routing. Fourthly, different groups of people may have different rules to establish and handle trust, and therefore, trust metrics are group dependent and non-uniform. When an end-to-end communication runs across multiple groups, more effort needs to be made to model the inter-operation between different trust-based routing protocols. Lastly, by its very nature, trust is a complex and multidimensional phenomenon [54]. Some kinds of trust like reputation are universally acceptable and transitive (social trust), while other kinds of trust is just a personal opinion (personal trust), which is non-transitive and incomparable at all. The breadth of the meaning of trust excludes any possibility to model trust metrics with a single algebraic structure. Multiple ones are needed to model different kinds of trust (i.e., heterogeneity of trust) and their combination to one applicable routing metric is a new topic.

To sum up, the diversity and complexity of trust metrics require a systematic analysis of their properties and the corresponding relationships with routing protocols. Based on a non-classical algebraic structure called bi-monoid, in this chapter we will develop a formal theory to investigate the correctness, optimality, and inter-operativity of trust-based routing protocols for WANETs. The rest of this chapter is organized as follows. In Section 6.2, we first utilize some simplified examples to provide our motivation for developing the formal approach adopted in this chapter. Then, we develop an abstract framework to facilitate our study on trust-based routing in Section 6.3. Next, we develop a non-classical path algebra based on bi-monoid to study indirect trust inference problems in Section 6.4. After that, in Section 6.5 we provide a systematic analysis of the relationship between trust metrics and trust-based routing protocols by identifying the basic algebraic properties that a trust metric must have in order to work correctly and optimally with different generalized distance-vector or link-state routing



(a) A single value as trust metric (b) A pair as trust metric (c) A triple as trust metric (d) Combined routing metric

Figure 6-2. Diversity of trust metrics.

protocols in WANETs. In Section 6.5 we extend our framework to model the interactions between different trust-based routing protocols, and characterize the conditions under which the correctness and optimality of routing operations can be guaranteed in the WANETs where multiple routing protocols coexist or different trust metrics are adopted. Finally, we conclude this chapter in Section 6.7.

6.2 Motivating Examples: Why Do We Need a Formal Study?

In this section, we utilize some simplified examples to provide our motivation for developing this formal approach.

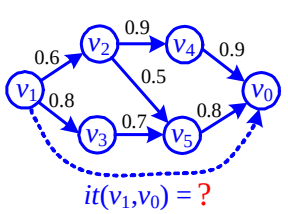
Example 1: diversity of trust metrics. Consider a reputation system based on direct interactions. A node's experience with another node is modeled as a binary event: positive or negative. Evidence $\langle r, s \rangle$ is conceptualized in terms of the numbers of positives (r) and negatives (s). Based on this simple evidence space, various trust metrics have been proposed in the literature, according to different design rationale:

- a trust value in the real interval $[-1, 1]$ (Figure 6-2 (a)); [151]
- a pair $\langle t, c \rangle$ in $[0, 1]^2$, where t and c represent trust and confidence values (Figure 6-2 (b)); [152]
- a triple $\langle b, d, u \rangle$ in $[0, 1]^3$, where b, d and u represent belief, disbelief and uncertainty values, respectively, and $b + d + u = 1$ (Figure 6-2 (c)); [77]
- a trust value can be combined with other QoS requirements (like data rate and delay) to form a combined routing metric (Figure 6-2 (d)). [166]

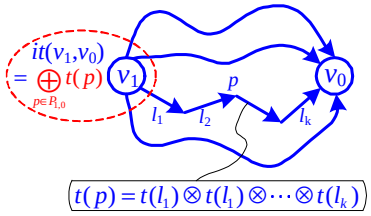
In fact, reputation is just one possible interpretation of trust. In different scenarios, trust may have totally different meanings and accordingly it should be evaluated in different ways [20, 54, 148]. Considering the diversity of emerging WANETs and their applications, it would be foolish to attempt to define a common trust metric. Therefore, more and more trust metrics with different forms and interpretations will emerge. Diversity of trust metrics raises two problems for trust-based routing: (1) It is possible that multiple trust metrics are adopted in one single WANET, therefore we need a generally applicable framework to facilitate our study on the compatibility of different trust metrics. (2) If we do not want to examine trust metrics (which may have infinite possibilities) one by one, we must find a way to abstract all these metrics and identify the key properties related to the correctness and optimality of routing protocols.

Example 2: diversity of operations on trust metrics. Consider an indirect trust inference problem in a reputation system. When two nodes have never directly interacted with each other before, there is no direct trust to be evaluated for them. However, when these two nodes meet (i.e., they are in each other's transmission range), they still need to infer an indirect trust between them based on existing direct trusts in the networks. Take Figure 6-3 (a) as an example, where the number on each link represents the trust value and we want to infer the indirect trust value $it(v_1, v_0)$ from v_1 to v_0 based on direct trust values. There also exist various ways in the literature to fulfil this task (refer to [152] and [54] and the references therein):

- We could choose the strongest path, determined by the path with the highest minimum trust value, and take the lowest trust value on that path as $it(v_1, v_0)$. Based on this inference algorithm, the strongest path is $\langle v_1, v_3, v_5, v_0 \rangle$ with $it(v_1, v_0) = 0.7$.
- We could choose the strongest path, determined by the path with the highest product of all trust values on the path, and take the product of all trust values on that path as $it(v_1, v_0)$. Based on this inference algorithm, the strongest path is $\langle v_1, v_2, v_4, v_0 \rangle$ with $it(v_1, v_0) = 0.49$.



(a) Trust inference problem



(b) Algebraic path problem

$\otimes$	$\oplus$	$it(v_1,v_0)$
min	max	0.70
$\times$	max	0.49
min	ave	0.66

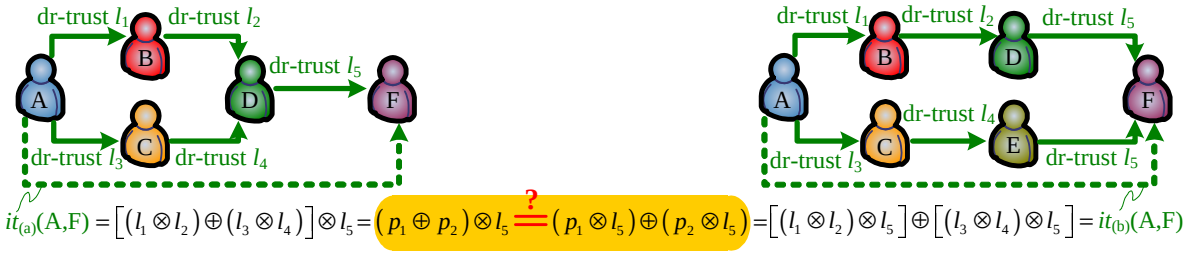
(c) Examples

Figure 6-3. Algebraic path formulation for indirect trust inference problems. In (a), solid lines and dashed lines represent direct and indirect trust relationships, respectively. In (b), solid curves represent trust paths consists of direct trust relationships only.

- $it(v_1, v_0)$ can be calculated as the weighted average of the minima of the trust values along the disjoint paths. The weights in the averaging process are given by v_1's trust in its direct out-neighbors. According to this inference algorithm, $it(v_1, v_0) = 0.66$.

Every trust inference algorithm mentioned above has its own pros and cons. Here, we are not interested in judging their usefulness according to different application scenarios, instead, we are focusing on developing an abstract framework so that we can reason their behavior as a whole. For trust inference problem, we know that we can apply a mathematical tool called path algebra [25, 55, 118]. We can define an operator $\otimes$ to concatenate trust metrics along a path, then we introduce another operator $\oplus$ to aggregate trust metrics across paths (refer to Figure 6-3 (b) for an illustration). When $\oplus$ and $\otimes$ satisfy certain properties, using path algebra, the problem of calculating $it(v_1, v_k)$ can be formulated uniformly as the one given in Figure 6-3 (b). From Figure 6-3 (c), we observe that all trust inference algorithm mentioned above can be included in this formulation with different interpretations of $\otimes$ and $\oplus$.

This example suggests that, instead of examining every possible trust metrics and operations one by one, we can study their behaviors as a whole by developing an algebraic formalism that abstracts trust relationships and also link resources as labels, and models the routing operations as certain operators on the labels. What related to the correctness and optimality of routing protocols are not the contents (the meaning of



(a) Case with dependent direct-trust paths

(b) Case with independent direct-trust paths

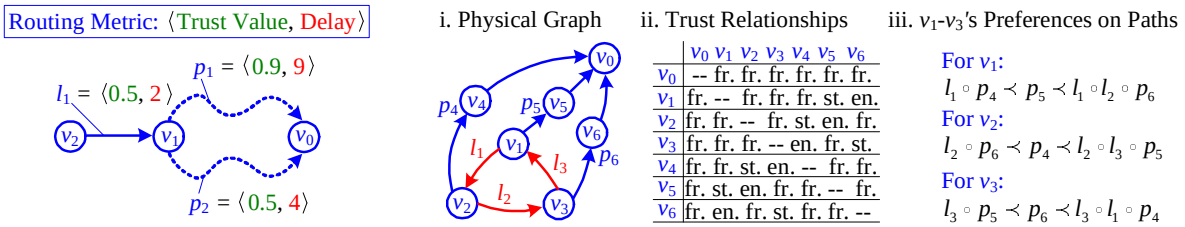
Figure 6-4. Distributivity of trust metrics. Here, solid lines and dashed lines represent direct and indirect trust relationships, respectively. In both cases, we assume that $p_1 = l_1 \otimes l_2$ and $p_2 = l_3 \otimes l_4$.

these trust metrics), but their algebraic properties. Therefore, we need describe, classify and analyze different trust metrics and operators based on their algebraic structures.

Although in this example we indicate that path algebra can be utilized to abstract and analyze trust inference problems, we should also point out that traditional path algebra cannot be directly applied here. The reason is that trust metrics may possess some algebraic properties which do not hold in traditional path algebra. We will illustrate this point in the following example.

Example 3: distributivity of trust metrics. Consider two trust inference problems given in Figure 6-4 (a) and (b). Using path algebra developed in Example 2, indirect trust value from A to E in Figure 6-4 (a) and (b) can be expressed as $it_{(a)}(A, F) = (p_1 \oplus p_2) \otimes l_5$ and $it_{(b)}(A, F) = (p_1 \otimes l_5) \oplus (p_2 \otimes l_5)$, respectively. For traditional path algebra based on semirings, $\otimes$ distributes over $\oplus$, and therefore $it_{(a)}(A, F) = it_{(b)}(A, F)$. The question is whether these two expressions are equal for trust metrics?

We answer this question by examining the physical meanings of two expressions, respectively. For Figure 6-4 (a), F has only one in-neighbor, i.e. D, which means that only D has direct interactions (direct trust) with F before. Therefore, D will act as witness to provide report about F's trustworthiness based on its own experience, which is measured by l_5. Node A will obtain two reports from two independent paths, with both reports claiming that the trustworthiness of F is l_5. The trustworthiness for the first report



(a) Suboptimality of path selection

(b) Forwarding loop problem

Figure 6-5. Examples of routing anomalies in trust-based routing. Here, solid lines and dashed lines represent physical links and physical paths, respectively.

itself (from path $A \rightarrow B \rightarrow D$) is p_1, while for the second report (from path $A \rightarrow C \rightarrow D$) is p_2. For Figure 6-4 (b), A also obtains two reports stating the trustworthiness of F is l_5, and the trustworthiness for reports themselves are p_1 and p_2, respectively. It seems that from the view point of A, there is no difference between two situations. Unfortunately, this is not true. One key observation is that, in Figure 6-4 (a), these two reports are issued by the same nodes D, while in Figure 6-4 (b) these two reports are issued by two different nodes D and E separately. Therefore, although node A collects the same second-hand evidence (two reports) on F in two situations, in Figure 6-4 (a) all these evidences come from a single source D while in Figure 6-4 (b) they come from independent sources D and E. Therefore, F is more trustworthy for A in Figure 6-4 (b). This example shows that distributivity does not hold for trust metrics and therefore we need develop a new path algebra for trust inference problems.

In the next two examples, we show that routing anomalies (suboptimal path selection and forwarding loop) may arise even in very simple network settings in trust-based routing for WANETs. The existence of routing anomalies necessitates the formal proofs of the correctness and optimality of trust-based routing protocols.

Example 4: suboptimum of trust-based routing. Consider the routing problem described in Figure 6-5 (a). Let routing metric be of the form $\langle t, d \rangle$, where t is trust value and d is delay. The metric for a physical path p is $\langle t_p, d_p \rangle$, where t_p is the minimal trust value of physical links along the path and d_p is the delay addition along the path.

Order $\preccurlyeq$ is defined as a lexicographic order: $\langle t_1, d_1\rangle \preccurlyeq \langle t_2, d_2\rangle \Leftrightarrow t_1 > t_2$ or ($t_1 = t_2$ and $d_1 \leq d_2$), where trustworthy paths are preferred, with small delays breaking ties. We say that path $p_1 = \langle t_1, d_1\rangle$ is better than or equal to $p_2 = \langle t_2, d_2\rangle$ if $\langle t_1, d_1\rangle \preccurlyeq \langle t_2, d_2\rangle$. This simple combined trust-related routing metric is proposed in [166] and also appears in Figure 6-2 (d) in Example 1. Here we are only interested in the routing problem with v_0 as the destination in Figure 6-5 (a). For node v_2, path $l_1 \circ p_1$ is not better than $l_1 \circ p_2$, because $l_1 \circ p_1 = \langle 0.5, 11\rangle$, $l_1 \circ p_2 = \langle 0.5, 6\rangle$ and $\langle 0.5, 6\rangle \preccurlyeq \langle 0.5, 11\rangle$. Therefore, for an optimal routing protocol, the selected path from v_2 to v_0 should be $l_1 \circ p_2$. However, node v_1 will always select path p_1 for the destination v_0 because path p_1 is optimal for it. As a consequence, for any hop-by-hop routing protocols, the actually selected path from v_2 to v_0 is $l_1 \circ p_1$, which is not optimal for the source node v_2.

Example 5: forwarding loop in trust-based routing. Consider the routing problem described in Figure 6-5 (b). All possible physical links are described in Figure 6-5 (b)-i, and we take node v_0 as the destination node. There are three types of trust relationships: friends, strangers and enemies, Figure 6-5 (b)-ii describes all trust relationships in a matrix form. Based on above information, for each node v_i, we can rank all possible paths from v_i to v_0 with order relation $\prec$. Figure 6-5 (b)-iii gives path ranking for nodes v_1, v_2 and v_3. Our ranking is based on the relationships between the source node and all intermediate nodes on the path.

For link-state routing, the most preferred paths calculated by v_1, v_2 and v_3 are $l_1 \circ p_4$, $l_2 \circ p_6$ and $l_3 \circ p_5$, respectively. Therefore, the next-hop for v_1 is v_2, for v_2 is v_3, and for v_3 is v_1. A loop $v_1 \to v_2 \to v_3 \to v_1$ appears, and packets will be forwarded in this loop forever.

For distance-vector routing, we take v_1 as an example. v_1 has only two choices for the next-hop: v_2 or v_5. In the first choice, we assume that v_1 takes v_2 as the next-hop. Given v_1's choice, v_3 will take v_6 as the next-hop. Given v_3's choice, v_2 will take v_3 as the next-hop. Given v_2's choice, v_1 should take v_5 as the next-hop, because $p_5 \prec l_1 \circ l_2 \circ p_6$. This contradicts to our assumption that v_1 takes v_2 as the next-hop. In the second

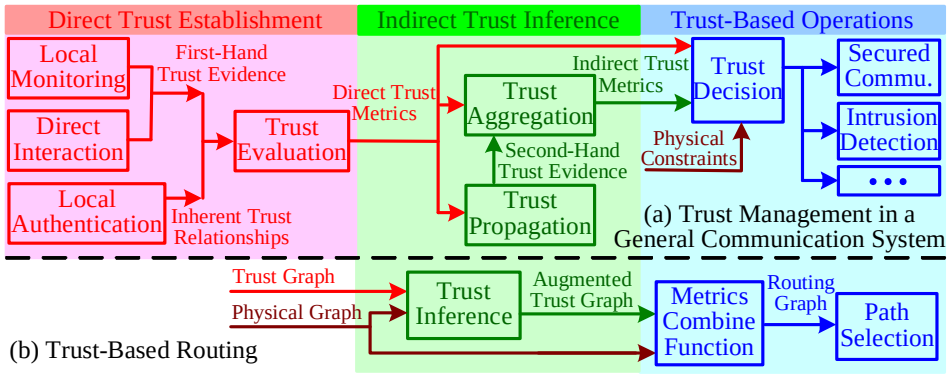


Figure 6-6. System model for (a) trust management in any communication systems and (b) trust-based routing.

choice, we assume that v_1 takes v_5 as the next-hop. Given v_1's choice, v_3 will take v_1 as the next-hop. Given v_3's choice, v_2 will take v_4 as the next-hop. Given v_2's choice, v_1 should take v_2 as the next-hop, because $l_1 \circ p_4 \prec p_5$. This contradicts to our assumption that v_1 takes v_5 as the next-hop. As we can observe that in both choices, we all end up with contradictions. Therefore, node v_1 keeps changing its routing table's configuration. This is called route oscillation, another name for forwarding loop within distance-vector routing.

From these few examples, we can observe that trust metrics are significantly different from traditional routing metrics and hence traditional routing algebra cannot be applicable, which calls for new tools for formal analysis.

6.3 Abstract Framework for Trust Metrics and Trust-Based Routing

6.3.1 System Model for Trust Management

In general, trust management in any communication system can be modeled as a procedure with three sequential phases (refer to Figure 6-6 (a) for an illustration):

Phase 1: direct trust establishment. Direct trust relationships can be obtained from the evidence created by the previous interactions or inherited from the pre-established social relationships in the physical world. In the former case, local monitoring schemes like Watchdog and Pathrater [111] and physical contact scheme [24] have been proposed to collect the first-hand trust evidence. In the latter case, participating node

only needs to verify the other side's identity [168] based on cryptographic operations. Trust evaluation is a mapping, which transforms the trust evidence or inherent trust relationships into direct trust metrics.

Phase 2: indirect trust inference. When the trust relationships are (at least partly) transitive[3] , more trust relationships (or indirect trust) can be derived from the direct trust. First of all, direct trust information need be propagated throughout the network. Any information gossip protocol or broadcast protocol can fulfill this task, and no specific path selection scheme is needed here. We also assume that appropriate cryptographic primitives are available in WANETs in order to provide a secure propagation of direct trust information. Then, based on these second-hand trust evidence, each node can establish new indirect trust relationships with its physical neighbors of which it knows nothing. Trust-based routing is differentiated from traditional routing by introducing trust inference as one of important preprocessing for path selection and packet forwarding, and therefore should be explicitly considered in our modeling. Note that when direct trust relationships are all non-transitive, no indirect trust can be derived from phase 2.

Phase 3: trust-based operations. Direct and indirect trust established in previous two phases will be utilized to support all kinds of operations with security requirements in this phase. For trust-based routing, trust-related routing metrics will be formed (e.g., combined with other QoS requirements) and used as the criteria for selecting most trustworthy paths between any S-D pair.

Phase 1 only concerns with the forming and evaluation of direct trust with neighboring nodes and has no direct relations with routing problem, which is orthogonal and complementary to the research described here and therefore excluded from the rest discussions of this chapter. Based on above discussions, we can identify two

[3] We will define transitive and non-transitive trust in Section 6.3.3.

elementary operations, namely, trust inference and trustworthy path selection for any trust-based routing protocols, which will be investigated here.

6.3.2 Graph Models for WANETs

We utilize graph models to describe physical resources and trust relationships in a WANET. Therefore, we first review some basic concepts applicable to a graph model. For a labeled direct graph $G = (V, E, \omega)$, V is the vertex set, $E \subseteq V \times V$ is the link set, and ω is a function which assigns each link $e \in E$ a label $\omega(e)$. Labels here are the abstraction of routing metrics or trust metrics on links. We also extend this concept to a path, i.e., the label $\omega(p)$ of a path p, which is the metric measuring the whole path. For edge $(i, j) \in E$, we say that node v_j is an out-neighbor of node v_i, and that node v_i is an in-neighbor of node v_j. A path p from v_1 to v_n is denoted by $p(v_1, v_n) = \langle v_1, v_2, \cdots, v_n \rangle$ and $p_{1,n}$ for short. If the last node of path p coincides with the first node of path q, the $p \circ q$ denotes the path formed by the concatenation of p and q. A path is simple or loop-free if all nodes from v_1 to v_n are distinct. If $v_1 = v_n$, then we say $p_{1,n}$ forms a loop.

The physical resources of a WANET are modeled by physical graph $G_H(V, E_H, h)$ where E_H is the set of directed edges representing wireless links[4] . The direct trust relationships are modeled as trust graph $G_T(V, E_T, t)$ where E_T is the set of directed edges representing direct trust relationships. When trust is transitive, we can derive augmented trust graph $G_T^*(V, E, it)$ based on $G_T(V, E_T, t)$ and indirect trust inference scheme, where $E = H \times H$. For a link $(i, j) \in E_H$, the routing metric of that link is

[4] This wireless link can be generalized as any physical link which can flow packets between two nodes. For example, in a WANET with partial infrastructure, this link can also be a wired link. In a delay-tolerant network (DTN), this link can be two wireless links combined with one node movement.

measured by a function $r(i,j)$, i.e.,

$$r(i,j) \triangleq \begin{cases} r\left(p(i,j), t(i,j)\right) & \text{for non-transitive trust} \\ r\left(p(i,j), it(i,j)\right) & \text{for transitive trust} \end{cases},$$

which converts the combination of physical properties and trust relationship into a trust related routing metrics. Graph $G_R(V,E_R,r)$ is called routing graph, because trustworthy path selection and packet forwarding are actually performed on it. When trust metrics are directly used as routing metrics, $r(i,j) = t(i,j)$. Note that, even in this case, trust/routing metrics are still constrained by physical graph, because $E_R = E_H$. Due to the node mobility and limited communication range of wireless communication techniques, $G_T(V,E_T,t)$ may have totally different topological structure from that of $G_H(V,E_H,h)$. Note that this property distinguishes our study from previous work on trust inference or trust-based routing in P2P networks or on-line social networks [54], where the trust graph and the physical graph are assumed to have the same topology.

6.3.3 Formalizing Trust Metric Space

Based on above discussions, trust metrics are just labels on links of labeled direct graphs. We denote the nonempty set of all possible labels as L. In order to define the space of trust-related routing metrics, we need introduce some mathematical structures on the set L, but remember that we also need minimize the number of specifications imposed on the structure over L in order to maximize the generality of our framework.

We first identify the indispensable structures for trust metrics. First of all, we should be able to derive the trust metric of a path from trust metrics on individual links. Therefore, we need introduce operators over the elements of set L to facilitate the combination of link labels (or trust metrics). From Example 2 in Project Description, we have already known that the following two combinations (or operations) over the labels are needed: (1) Concatenation of serial labels with operator $\otimes$; (2) Aggregation of parallel labels with operator $\oplus$. At least these two operations should be closed,

i.e., for all $a, b \in L$, the results of the operations $a \otimes b$ and $a \otimes b$ are also in L. This means that by combining two legal labels we are guaranteed to obtain a legal label too. Obviously, this is an algebraic structure because we define some operations on a set and we require these operations satisfying some constraints/properties. In the following discussions, we will introduce more necessary properties of these two operations in the place whenever needed, and based on these operator's properties, we can get different algebraic structures like semiring and bi-monoid (cf. Section 6.4). We will also introduce new operators (like δ in Section 6.5) for trustworthy path selections.

Secondly, since we need select the most trustworthy path from all physically available paths, we should be able to compare all elements in L. Therefore, we also need introduce some kind of order relation into L. We define a total pre-order $\preccurlyeq$ over L as follows[5] . For two paths or links p and q in a graph, let $\omega(p) \in L$ and $\omega(q) \in L$ denote their labels. If $\omega(p) \preccurlyeq \omega(q)$, we say that p is weakly preferred to q (e.g., path p is at least as trustworthy as path q). If $\omega(p) \preccurlyeq \omega(q)$ and $\omega(q) \preccurlyeq \omega(p)$, then we write $\omega(p) \sim \omega(q)$ and say that p and q are equally preferred. For any pair $\omega(p), \omega(q) \in L$ we have either $\omega(p) \preccurlyeq \omega(q)$ or $\omega(q) \preccurlyeq \omega(p)$. Functions $\max$ and $\min$ are defined with respect to $\preccurlyeq$. Note that $\omega(p) \prec \omega(q)$ mean $\omega(p) \preccurlyeq \omega(q)$ and $\omega(p) \not\sim \omega(q)$, i.e., strictly preferred.

Therefore, we can define the nonempty label set with algebraic structure and order relation described above as the abstract trust metric space.

We distinguish two kinds of trust in this project: transitive and non-transitive trust. For transitive trust, if node v_i has trust $t(i, j)$ in node v_j, and v_j has trust $t(j, k)$ in node v_k, then v_i should have some trust $t(i, k)$ in v_k that is a function of $t(i, j)$ and $t(j, k)$. For non-transitive trust, however, we cannot obtain any conclusion on $t(i, k)$, given $t(i, j)$ and $t(j, k)$. Example 5 in Project Description illustrates this concept. We note that transitive trust (and consequently, indirect trust) is important for any

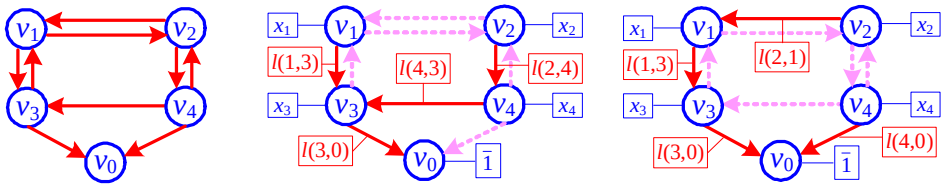
[5] Recall that a pre-order is a reflexive and transitive relation.

trust-based routing to be practical in large-scale dynamic WANETs. Because of the large number of nodes and node mobility, it is impossible for one node to have direct trust with any of its physical neighbors. When the physical link between two nodes is needed to perform multi-hop communications, the trust on this link must be derived beforehand. We also note that trust is not always transitive in real life, and therefore we cannot exclude non-transitive trust from our framework. In this project, we will first study the situation when all trusts in a WANET are transitive, which is called homogeneous trust environment, and then extend our study to the situation when transitive and non-transitive trusts coexist in a WANET, which is called heterogeneous trust environment.

We also distinguish two situations when the whole WANET is modeled by one trust metric space (and accordingly with one trust-based routing protocol) or by multiple trust metric spaces (and accordingly with multiple trust-based routing protocols), which are called uniform trust environment and diverse trust environment, respectively. We will first study the simpler situation, i.e., uniform trust environment in Section 6.4 and Section 6.5. In Section 6.6, we will study the inter-operation problems introduced by multiple trust metric spaces.

6.3.4 Formalizing Routing Protocols

Given the routing graph, for any hop-by-hop routing protocol, its main task is to generate and maintain a path between each S-D pair with the desirable properties (defined by routing metrics). Because for each destination, every source will have in its routing table the next hop to reach that destination, a spanning tree rooted at each destination is defined implicitly by the set of routing tables residing in a network. We call this spanning tree as in-tree with the root node as the given destination. Therefore, any routing protocol can be abstracted as the collection of rules and procedures to generate an in-tree for a given destination.



(a) Routing Graph (b) In-Tree IT_1 (c) In-Tree IT_2

Figure 6-7. Exemplary in-trees for root node v_0 (destination).

We illustrate this concept by the following example[6] . Consider a network modeled by the routing graph $G(E, E_L, L)$ given in Figure 6-7 (a), with the destination node v_0. This network has many in-trees whose root is v_0; two of those in-trees, IT_1 and IT_2, are shown in Figure 6-7 (b) and (c), respectively. In each of these in-trees, the node label x_i of node v_i can be expressed using an concatenation operator $\otimes$ as follows based on the structures and link labels of the given in-tree.

1. For the root node v_0, $x_0 = \bar{1}$ (a special label which will be explained a little later).
2. If node v_i is not the root and node v_j is the parent of node v_i in the in-tree and $l(i,j)$ is the label of link (i,j), then $x_i = l(i,j) \otimes x_j$.

For example, the labels of all nodes in in-trees IT_1 (cf. Figure 6-7 (b)) and IT_2 (cf. Figure 6-7 (c)) are given in Table 6-1.

Obviously, node label is related to the choice of in-trees, and therefore, is routing-protocol dependent. Node label can be interpreted in the following two ways:

1. Individually, label x_i represents the routing table configuration for destination v_0 at node v_i. For example, in IT_1, $x_2 = l(2,4) \otimes x_4$ means that at node v_2 the out-link for v_0 will be link $(2,4)$ with the next-hop node v_4.
2. Given other nodes' configurations x_j $(j \neq i)$, label x_i also represents the chosen path from v_i to destination v_0. For example, in IT_1:

[6] We restrict ourselves to unicasts in this chapter. Our discussion below is restricted to one destination node v_0 scenarios. However, it is straightforward to be extended to the case with arbitrary number of destination nodes (i.e., one in-tree for each destination node).

Table 6-1. Node labels in Figure 6-7.

For IT_1 in Figure 6-7 (b)	For IT_2 in Figure 6-7 (c)
$x_0 = \bar{1}$	$x_0 = \bar{1}$
$x_1 = l(1,3) \otimes x_3$	$x_1 = l(1,3) \otimes x_3$
$x_2 = l(2,4) \otimes x_4$	$x_2 = l(2,1) \otimes x_1$
$x_3 = l(3,0) \otimes x_0$	$x_3 = l(3,0) \otimes x_0$
$x_4 = l(4,3) \otimes x_3$	$x_4 = l(4,0) \otimes x_0$

$$
\begin{aligned}
x_2 &= l(2,4) \otimes x_4 \\
&= l(2,4) \otimes (l(4,3) \otimes x_3) \text{ (given } x_4\text{)} \\
&= l(2,4) \otimes (l(4,3) \otimes (l(3,0) \otimes \bar{1})) \text{ (given } x_3 \text{ and } x_0\text{)} \ *,
\end{aligned}
$$

which means the path from v_2 to v_0 chosen by in-tree IT_1 will be $p_{2,0} = \langle v_2, v_4, v_3, v_0 \rangle$. Here, x_2 in $(*)$ form equals to path label of $p_{2,0}$, which is denoted as $l(p_{2,0})$.

The existing hop-by-hop routing protocols in WANETs can be divided into two categories according to their different path calculating approaches: namely, link-state routing and distance-vector routing

In the link-state approach to routing, each node broadcasts updates of its local topology information (link state) to the rest of the network. These broadcasts can be periodic or event-driven. By putting the updates together, each node is able to reconstruct the routing graph $G_R(V, E_R, r)$ for the entire network. Given $G_R(V, E_R, r)$, a node can then construct its routing tables appropriately by running Dijkstra's shortest path algorithm (when taking the distance as the routing metric). For the general routing metrics, a generalized Dijkstra's algorithm is used. Given the destination v_0, each node v_i will calculate the most preferred path $p^*_{i,0}$ on $G_R(V, E_R, r)$, according to $r(p^*_{i,0}) = \min \{r(p_{i,0}) \mid \forall p_{i,0} \in \mathcal{P}_{i,0}\}$, where $\mathcal{P}_{i,0}$ is the set of all paths from v_i to v_0. When v_j is the next-hop node on path $p^*_{i,0}$, we have $x_i = r(i,j) \otimes x_j$ where $v_j \in N_i$. Exemplary routing protocols for WANETs in the literature which fall in this category include LQSR, HSR, OLSR and HSLS [131].

In the distance-vector approach to routing, neighboring nodes exchange (advertise) vectors of distances with each other. Each entry in a distance-vector corresponds to a particular destination and contains the current distance estimate of the shortest path

from the source to the corresponding destination. For the general routing metrics, it means that each node v_i only knows its out-neighbors[7] N_i, its out-going links, and its out-neighbors' node labels. Node v_i will calculate its own node label x_i (which has one-to-one mapping with the entry in routing table for v_0 and the selected path $p_{i,0}$) using a generalized Bellman-Ford algorithm: $x_i = \min\{r(i,j) \otimes x_j \mid \forall v_j \in N_i\}$. Exemplary routing protocols for WANETs in the literature which fall in this category include AODV and DSDV [131].

Based on the above discussion, we can naturally introduce our definitions on the correctness and optimality of any routing protocol $\mathcal{R}$ as follows:

Definition 6.1. ***[$\mathcal{R}$-Correctness]*** *A (trust-based) routing protocol $\mathcal{R}$ is said to be correct, if given any $G_R(V,E_R,r)$, x_i's ($i = 1, 2, \cdots, n$) calculated by $\mathcal{R}$ form an in-tree.*

Definition 6.2. ***[$\mathcal{R}$-Optimality]*** *A (trust-based) routing protocol $\mathcal{R}$ is said to be optimal, if given any $G_R(V,E_R,r)$, x_i's ($i = 1, 2, \cdots, n$) calculated by $\mathcal{R}$ satisfy*

1. *x_i's form an in-tree and*
2. *$\forall v_i \in V : x_i = \min\{ r(p_{i,0}) \mid \forall p_{i,0} \in \mathcal{P}_{i,0}\}$.*

Based on the definition of in-trees, a correct routing protocol $\mathcal{R}$ will not create any routing loops, i.e., $\mathcal{R}$ always converges to loop-free states. For an optimal routing protocol, for every node v_i ($i \neq 0$), data packets will be forwarded along the most preferred path $p^*_{i,0}$ among all existing paths from v_i to v_0.

Note that in our routing protocol abstraction, some details such as when to calculate the next hop (the difference between reactive and proactive routing) are ignored, since they are irrelevant to the theme of our study here, i.e., the correctness and optimality of routing protocols. Also we exclude source routing since for trust-based routing it is

[7] We assume that for directed link (i,j), data packets can flow from v_i to v_j while signaling routing packets may be sent in the opposite direction.

impractical to assume relay nodes will forward packets according to the path assigned by the source.

6.4 Path Algebra for Indirect Trust Inference

In this section, we develop a non-classical path algebra based on bi-monoid to study indirect trust inference problems.

6.4.1 Algebraic Foundations

The general frameworks used in this and nxet sections are based on the algebraic structure of semigroups and monoids. Thus, we first briefly review some relevant results.[8]

A semigroup $(S, \oplus)$ is a non-empty set S with a binary operator $\oplus$ such that (for all $a, b, c \in S$)

- $a \oplus b \in S$ ($\oplus$-Closure),
- $(a \oplus b) \oplus c = a \oplus (b \oplus c)$ ($\oplus$-Associativity).

Moreover, a semigroup $(S, \oplus)$ is called (for all $a, b \in S$)

- commutative when $a \oplus b = b \oplus a$ ($\oplus$-Commutativity),
- idempotent when $a \oplus a = a$ ($\oplus$-Idempotency),
- selective when $a \oplus b = a$ or b ($\oplus$-Selectivity).

Note that a selective semigroup must be idempotent, but the converse is not true.

A semigroup $(S, \oplus)$ may have some special elements:

- $\varepsilon \in S$ is an identity if $\forall a \in S : \varepsilon \oplus a = a \oplus \varepsilon = a$,
- $\sigma \in S$ is an annihilator if $\forall a \in S : \sigma \oplus a = a \oplus \sigma = \sigma$,
- $a^{-1} \in S$ is an inverse of $a \in S$, if $a \oplus a^{-1} = a^{-1} \oplus a = \varepsilon$.

[8] Refer to [25, 55, 56, 118] for a more complete survey of the issues exposed here.

Note that ε, σ and a^{-1} defined above are unique if they exist. A semigroup is a monoid if it has an identity. A monoid is a group if $\forall a \in S : a^{-1}$ exists.

A bi-semigroup is an algebraic structure $(S, \oplus, \otimes)$ with two binary operators $\oplus$ and $\otimes$ over the set S which satisfy

- $(a \oplus b) \oplus c = a \oplus (b \oplus c)$ ($\oplus$-Associativity),
- $(a \otimes b) \otimes c = a \otimes (b \otimes c)$ ($\otimes$-Associativity),

that is, when both $(S, \oplus)$ and $(S, \otimes)$ are semigroups.

For a commutative monoid $(S, \oplus)$, it is always possible to introduce a pre-order relation over S, denoted $\preccurlyeq_{\oplus}$, as:

$$\forall a, b \in S : a \preccurlyeq_{\oplus} b \Leftrightarrow \exists c \in S : a = b \oplus c.$$

We call this relation as canonical pre-order, since the identity and associativity of $\oplus$ ensure that $\preccurlyeq_{\oplus}$ is indeed a pre-order. Note that $\oplus$-commutativity ensures that the following definition of $\preccurlyeq_{\oplus}$ are equivalent:

$$\exists c \in S : a = b \oplus c \Leftrightarrow \exists c \in S : a = c \oplus b.$$

The canonical pre-order $\preccurlyeq_{\oplus}$ has the following properties:

- If ε ($\oplus$-identity) exists, then $\forall a \in S : a \preccurlyeq_{\oplus} \varepsilon$.
- If σ ($\oplus$-annihilator) exists, then $\forall a \in S : \sigma \preccurlyeq_{\oplus} a$.

6.4.2 Trust Inference Problem Formalization

After trust propagation subphase, every node will know the trust graph. If the trust is transitive, the functionality of trust inference is to calculate the indirect trust based on the trust graph which describes direct trust relationships. In this subsection, we first formalize this procedure as follows.

Given the trust graph $G_T(V, E_T, t)$, where $t : E_T \mapsto T$ is a function which assigns each edge $e \in E_T$ a label $t(e) \in T$ (i.e., the direct trust metric on e), we need to compute

the indirect trust metric $it(v_i, v_j)$ for every $(v_i, v_j) \in V \times V$. We first introduce two operations over the labels:

- Concatenation of serial labels with operator $\otimes$,
- Aggregation two parallel labels with operator $\oplus$.

Mathematically, $\otimes, \oplus : T \times T \mapsto T$ are two functions which combine two labels into a new one. Therefore, the label of a nontrivial path $p_{1,k} = \langle v_1, v_2, \cdots, v_k \rangle$ (i.e., the trust metric of path $p_{1,k}$) of G_T, denoted $t(p_{1,k})$, is given by

$$t(p_{1,k}) \triangleq (\cdots((t_{1,2} \otimes t_{2,3}) \otimes t_{3,4}) \cdots) \otimes t_{k-1,k},^{9}$$

and $t(p_i) \triangleq \bar{1}$ for trivial path $p_i = \langle v_i \rangle$.

A solution to an indirect trust inference problem is a function $it : V \times V \mapsto T$ such that $it(v_i, v_i) \triangleq \bar{1}$ and for all $i \neq j$, $it(v_i, v_j) \triangleq \bigoplus_{p \in \mathcal{P}(i,j)} t(p)$, where $\mathcal{P}(i,j)$ is the set of all paths from v_i to v_j.

In this chapter, we solve this problem within the following algebraic structure, called bi-monoid:

Definition 6.3. ***[Bi-Monoid]*** *Given a trust graph G_T, we define the non-classic path algebra over G_T as an algebraic structure $(T, \oplus, \otimes, \bar{0}, \bar{1})$, where*

- *$(T, \oplus, \bar{0})$ is a commutative monoid with $\bar{0}$ as its identity,*
- *$(T, \otimes, \bar{1})$ is a monoid with $\bar{1}$ as its identity,*
- *$\oplus$-identity $\bar{0}$ is also an annihilator for $\otimes$.*

We call this algebraic structure $(T, \oplus, \otimes)$ as bi-monoid.

From Section 6.4.1, we know that there exists a canonical pre-order $\preccurlyeq_\oplus$ over T. We indicate it as $\preccurlyeq$ for short.

[9] When $\otimes$ is commutative, it equals to $t(p_{1,k}) \triangleq t_{1,2} \otimes t_{2,3} \otimes \cdots \otimes t_{k-1,k}$.

In previous work like [152], trust inference problems are solved within the algebraic structure called semiring:

Definition 6.4. ***[Semiring]*** *A bi-monoid* $(T, \oplus, \otimes, \bar{0}, \bar{1})$ *is a semiring if* $\otimes$ *distributes over* $\oplus$*, i.e., for all* $a, b, c \in T$*:*

- $(a \oplus b) \otimes c = (a \otimes c) \oplus (b \otimes c)$ *(Left Distribution),*
- $c \otimes (a \oplus b) = (c \otimes a) \oplus (c \otimes b)$ *(Right Distribution).*

6.4.3 Verification of the Bi-Monoid Properties

Here we give the physical meanings of the constraints/ properties we impose on the algebraic structure of trust metrics in Def. 6.3 and explain why these properties should be held for all trust metrics in practice. Remember that we need to keep the property set as small as possible in order to leave more room for trust metric design.

The associativity for both $\otimes$ and $\oplus$ operators allows the incremental calculation of trust metrics: If one more label needs to be aggregated into the current "total," then it can be done in one step, without having to recall all labels that were aggregated for the current total. The same goes for concatenation. Commutativity for aggregation $\oplus$ makes irrelevant the order in which labels are taken into account (i.e., which one is first, which one is second, etc.).

The $\bar{0}$ element (identity for $\oplus$, annihilator for $\otimes$) corresponds to nonexistent trust relations between nodes. The rationale is that if $\bar{0}$ is encountered along a path, then the whole path "through" this relation should have label equal to $\bar{0}$. Also, such paths should be ignored in $\oplus$-aggregation.

The $\bar{1}$ element (identity for $\otimes$) corresponds to the most trustworthy relations between nodes. This can also be seen as the trust relation of a node with itself (i.e., trivial path in G_T). If a path is extended with a link of label $\bar{1}$, the path label should remain the same.

The key difference between our work and previous work is that our model does not impose distributivity on trust metrics.

Next, we introduce a new property of trust metrics, which is not included in our bi-monoid model (i.e., Def. 6.3), but we believe it should hold in practice, since any path with label $\bar{1}$ is most preferred.

Definition 6.5. ***[Absorptivity]*** *A trust graph $G_T(V,E_T,t)$ is said to be absorptive if for every simple directed cycle $c = \langle v_1, v_2, \cdots, v_k, v_1\rangle$ in G_T, we have $\bar{1} \preccurlyeq t(c)$.*

Absorptive graph is a generalization of the absence of a negative cycle in a graph with real numbers as edge weights. It plays an important role to make sure the solution of trust inference problem exists.

6.4.4 Solving Path Algebraic Problems

(1) Solving the Problem with Semirings: We first recall how to solve a trust inference problem within the framework of semiring. The operations $\oplus$ and $\otimes$ can be extended in the usual way to matrices built from the elements of the set T. Let $\mathcal{M}_n(T)$ denote the set of all $n \times n$ matrices over T, and for $\mathbf{A} \in \mathcal{M}_n(T)$ let $\mathbf{A}_{ij}$ denote the (i,j)-entry of $\mathbf{A}$. For all $\mathbf{A}, \mathbf{B} \in \mathcal{M}_n(T)$, we define $\mathbf{A} \oplus \mathbf{B}$ and $\mathbf{A} \otimes \mathbf{B}$ by $(\mathbf{A} \oplus \mathbf{B})_{ij} \triangleq \mathbf{A}_{ij} \oplus \mathbf{A}_{ij}$ and $(\mathbf{A} \otimes \mathbf{B})_{ij} \triangleq \bigoplus_{k=1}^{n} (\mathbf{A}_{ik} \otimes \mathbf{B}_{kj})$.

The $\oplus$-identity matrix $\bar{\mathbf{0}}$ and $\otimes$-identity matrix $\bar{\mathbf{1}}$ are given by:

$$\bar{\mathbf{0}}_{ij} \triangleq \bar{0} \text{ and } \bar{\mathbf{1}}_{ij} \triangleq \begin{cases} \bar{1} & \text{if } i = j, \\ \bar{0} & \text{otherwise.} \end{cases}$$

Then $(\mathcal{M}_n(T), \oplus, \otimes)$ form another semiring.

The adjacency matrix $\mathbf{A}$ of the trust graph $G_T(V,E_T,t)$ is

$$\mathbf{A}_{ij} \triangleq \begin{cases} t(i,j) & \text{if } (i,j) \in E_T, \\ \bar{0} & \text{otherwise.} \end{cases}$$

Define $\mathbf{A}^{(k)}$ as

$$\mathbf{A}^{(k)} \triangleq \bar{\mathbf{1}} \oplus \mathbf{A} \oplus \cdots \oplus \mathbf{A}^k, \text{ where } \mathbf{A}^k \triangleq \mathbf{A} \otimes \mathbf{A} \otimes \cdots \otimes \mathbf{A} \ (k \text{ times}).$$

Let $\mathcal{P}(i,j)$, $\mathcal{P}^k(i,j)$ and $\mathcal{P}^{(k)}(i,j)$ be the set of all paths in G_T from i to j, the set of paths from i to j with length k and the set of paths from i to j with length at most k,

respectively. Obviously, $\mathcal{P}^k(i,j) \subseteq \mathcal{P}^{(k)}(i,j) \subseteq \mathcal{P}(i,j)$. The following connection between matrix powers and paths of a certain length is well known:

$$\left(\mathbf{A}^k\right)_{ij} = \bigoplus_{p \in \mathcal{P}^k(i,j)} t(p), \tag{6–1}$$

Note that the proof of Eq. (6–1) relies on the (left) distribution rule. From Eq. (6–1), we directly obtain

$$\left(\mathbf{A}^{(k)}\right)_{ij} = \bigoplus_{p \in \mathcal{P}^{(k)}(i,j)} t(p). \tag{6–2}$$

If the trust graph $G_T(V, E_T, t)$ is absorptive, then we only need to consider simple paths (i.e., no repetitions of nodes along the path is allowed). In the graph G_T with $|V| = n$, no path length is larger than $n-1$, which means that $\mathcal{P}^{(n-1)}(i,j) = \mathcal{P}(i,j)$. Therefore $\mathbf{A}^{(n-1)} = \mathbf{A}^{(n-1+k)}$ for any $k \geq 0$. From Eq. (6–2), we obtain the solution

$$it(v_i, v_j) = \bigoplus_{p \in \mathcal{P}(i,j)} t(p) = \left(\mathbf{A}^{(n-1)}\right)_{ij}. \tag{6–3}$$

Many efficient algorithms are available in the literature to compute $\mathbf{A}^k$, see [25, 55, 56, 118] and the references therein.

(2) Eliminating Distributivity: Our way to solve the trust inference problem without distributivity is based on the grouping function g introduced in [96]. The basic idea is that by utilizing the grouping function g, we first convert the problem in a bi-monoid $\mathcal{BM}$ into the problem in a semiring $\mathcal{C}$. Then, we mapping back labels computed in $\mathcal{C}$ to labels in $\mathcal{BM}$.

Let $M(T)$ be the set of all countable multisets that are composed of elements in T. Mathematically the grouping function $g : M(T) \mapsto M(T)$ has the following properties:

1. For $M_1, M_2 \in M(T)$, let $M_1 \otimes M_2$ be the multiset such that $M_1 \otimes M_2 \triangleq \{t_1 \otimes t_2 \mid t_1 \in M_1 \text{ and } t_2 \in M_2\}$. Then, $g(M_1 \otimes M_2) = g(g(M_1) \otimes g(M_2))$;
2. For all $M_i \in M(T)$, where $i \in I$ and I is a countable index set, we have $g\left\{\bigcup_{i \in I} M_i\right\} = g\left\{\bigcup_{i \in I} g(M_i)\right\}$;
3. If $M \in M(T)$ then $\oplus(M) = \oplus g(M)$.

Intuitively, if M is a set of path labels, then g groups labels together as long as this does not violate the distributivity. Property 1) states that the grouping process is compatible with $\otimes$, i.e., grouping before a trust evaluation does not change the result of the evaluation. Property 2) requires a natural commutativity property of the grouping process. Finally, Property 3) states that the grouping process is compatible with $\oplus$.

Based on the grouping function g, we can always turn a $\mathcal{BM} = (T, \oplus, \otimes, \bar{0}, \bar{1})$ into $\mathcal{C} = (\hat{T}, \hat{\oplus}, \hat{\otimes}, \bar{0}', \bar{1}')$ as follows:

- $\hat{T} \triangleq g(M(T))$;
- $\forall \hat{M}_1, \hat{M}_2 \in \hat{T} : \hat{M}_1 \hat{\otimes} \hat{M}_2 \triangleq g\left(\hat{M}_1 \otimes \hat{M}_2\right)$;
- $\forall \hat{M} \in M(\hat{T}) : \hat{\oplus}\hat{M} \triangleq g\left\{\bigcup \hat{M}\right\}$;
- $\bar{0}' = g(0)$ and $\bar{1}' = g(1)$.

Then, it has been proved in [96] that the solution in $\mathcal{BM}$ is:

$$it(v_i, v_j) = \bigoplus \left(\hat{\bigoplus_{p \in \mathcal{P}(i,j)}} \hat{t}(p) \right) \tag{6–4}$$

Note that the computation of $\hat{\bigoplus}\hat{t}(p)$ in Eq. (6–4) can be performed like in semirings, i.e. Eq. (6–3).

It is easy to show that the trust inference problem has a solution in $\mathcal{BM}$ if and only if it has a solution in $\mathcal{C}$ defined above and $G_T(V, E_T, \hat{t})$ is absorptive if and only if $G_T(V, E_T, t)$ is absorptive. Therefore, the condition that there exists a solution for $\mathcal{BM}$ is that the trust graph $G_T(V, E_T, t)$ is absorptive.

Given the solution $it(v_i, v_j)$, we will have a complete graph $G_T^*(V, E, it)$, where $E = T \times T$, and it is a function which assigns each edge $e \in E$ an label $it(e) \in T$ (i.e., indirect trust metric). Graph G_T^* is called augmented trust graph.

6.5 Routing Algebra for Uniform Trust Environment

In this section we develop a non-classic routing algebra to study the correctness and optimality of routing protocols under the uniform trust environment, i.e., we assume

all nodes in a WANET establish and handle trust in the same way. We will relax this assumption in Section 6.6.

6.5.1 Non-Classic Routing Algebra for Trust-Based Routing

In the physical graph $G_H(V, E_H, h)$, function $h : E_H \mapsto H$ assigns each edge $(i,j) \in E_H$ a physical label $h(i,j) \in H$ which describes the physical properties of that link. We define a special physical label $\phi \in H$, which corresponds to "no physical link" in G_H. After trust inference process, we can combine the physical graph with the augmented trust graph $G_T^*(V, E, it)$ to obtain the routing graph $G_R(V, E_R, r)$, where $E_R = E_H$ and $r : H \times T \mapsto R$ is a function which assigns each edge $(i,j) \in E_R$ a label $r(i,j) \in R$ which combines the physical label and trust label on that link, i.e., $r(i,j) = r\,(h(i,j), it(i,j))$. Note that, when trust is non-transitive, we directly use trust graphs to combine with physical graphs, and obtain routing graphs, i.e., in this case we assume $G_T^* = G_T$.

Given the augmented trust graph $G_T^*(V, E, it)$ and routing graph $G_R(V, E_R, r)$, the main task of trust-based routing is to find a path from each node $v_i \in V - \{v_0\}$ to the destination v_0. In what follows, we develop a non-classic routing algebra to formalize this procedure.

Definition 6.6. ***[Routing Algebra]*** *Given the routing graph $G_R(V, E_R, r)$, we define the non-classic routing algebra over G_R as an algebraic structure $\mathcal{A} = (R, \oplus, \otimes, \bar{0}, \bar{1}, \delta)$, where*

- $(R, \oplus, \bar{0})$ *is a commutative monoid with $\bar{0}$ as its identity,*
- $(R, \otimes, \bar{1})$ *is a monoid with $\bar{1}$ as its identity,*
- *$\oplus$-identity $\bar{0}$ is also an annihilator for $\otimes$.*
- *$\delta : R \times R \mapsto R$ is a function which combines labels on the flow graph of a physical path in G_R into a new one.*

Before we proceed, some comments on Def. 6.6 seems in order.

- Do not confuse $\bar{0}$ and $\bar{1}$ in $\mathcal{A}$ with the ones in $\mathcal{BM}$ of Def. 6.3. Here $\bar{0}$ and $\bar{1}$ are special elements in R not in T. Also note that the operands of $\oplus$ and $\otimes$ are

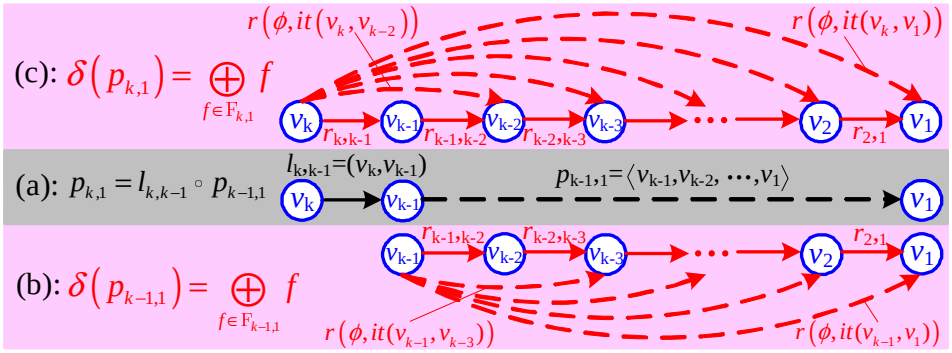


Figure 6-8. Flow graphs, δ function and trust evaluation on paths.

elements in R, and therefore $\oplus$ and $\otimes$ here may have totally different meanings compared with their counterparts in $\mathcal{BM}$.

- We can also introduce a canonical pre-order $\preccurlyeq_{\oplus}$ over R, as what we have done for $\mathcal{BM}$. Also we indicate it as $\preccurlyeq$ for short from now on. $\bar{0}$ is also called the least preferred label since $\forall a \in R - \{\bar{0}\} : a \prec \bar{0}$ and $\bar{1}$ is called the most preferred label since $\forall a \in R - \{\bar{1}\} : \bar{1} \prec a$.

- Function δ can be interpreted as a combined operator of $\oplus$ and $\otimes$, and its operand is a special structure called flow graph, which will be explained in details as follows.

For a physical path $p_{k,1} = \langle v_k, v_{k-1}, \cdots, v_2, v_1 \rangle$ in G_R, we define the flow graph $\mathcal{F}_{k,1}$ of that path as a labeled directed graph (refer to Figure 6-8 (c) for an illustration) such that

- The vertex set of $\mathcal{F}_{k,1}$ is $V_{1,k} = \{v_1, v_2, \cdots, v_k\}$.
- The edge set of $\mathcal{F}_{k,1}$ is $E_R(\mathcal{F}_{k,1}) \cup E(\mathcal{F}_{k,1})$, where
  - $E_R(\mathcal{F}_{k,1}) \triangleq \{(v_k, v_{k-1}), (v_{k-1}, v_{k-2}), \cdots, (v_2, v_1)\}$ (denoted by solid lines in Figure 6-8 (c)).
  - $E(\mathcal{F}_{k,1}) \triangleq \{(v_k, v_{k-2}), (v_k, v_{k-3}), \cdots, (v_k, v_1)\}$ (denoted by dashed lines in Figure 6-8 (c)).
- The label on each edge of $\mathcal{F}_{k,1}$ is defined by
  - for edge $(v_i, v_j) \in E_R(\mathcal{F}_{k,1})$, the label will be $r_{i,j}$, i.e., the same as that in the routing graph;

– for edge $(v_k, v_i) \in E(\mathcal{F}_{k,1})$ ($i = 1, \cdots, k-2$), the label will be $r(\phi, it(v_k, v_i))$, which is the combination of "no physical link" with the indirect trust $it(v_k, v_i)$ between two nodes.

Obviously, $E_R(\mathcal{F}_{k,1}) \subseteq E_R$ and $E(\mathcal{F}_{k,1}) \subset E$.

Flow graph $\mathcal{F}_{k,1}$ for the physical path $p_{k,1}$ includes all information that can be utilized to evaluate the physical path $p_{k,1}$ from the view point of the source node v_k. It includes the qualities of physical links which consist of path $p_{k,1}$ as well as the trust relationships of the source v_k with intermediate nodes on that physical path. For traditional routing metrics (i.e., $h(i,j)$ in G_H), a path metric can be simply calculated from the physical link metrics. For example, for path $p_{k,1} = \langle v_k, v_{k-1}, \cdots, v_2, v_1 \rangle$ in G_H, we have $h(p_{k,1}) = h(l_{k,k-1} \circ p_{k-1,1}) = h(l_{k,k-1}) \otimes h(p_{k-1,1})$, where $l_{k,k-1}$ represents link $(k, k-1)$. For trust related metrics, we can observe from Figure 6-8 that in general, $\delta(p_{k,1}) = \delta(l_{k,k-1} \circ p_{k-1,1}) \neq \delta(l_{k,k-1}) \otimes \delta(p_{k-1,1})$. However, we can utilize $\mathcal{BM}$ introduced in Section 6.4 to calculate $\delta(p_{k,1})$. Note that for flow graph $\mathcal{F}_{k,1}$, $(R, \oplus, \otimes, \bar{0}, \bar{1})$ forms a bi-monoid over R. Let $F_{k,1}$ be the set of all paths from v_k to v_1 in $\mathcal{F}_{k,1}$, $\delta(p_{k,1})$ can be calculated as $\delta(p_{k,1}) = \bigoplus_{f \in F_{k,1}} f$.

6.5.2 Conditions for Correct and Optimal Routing

We consider the following properties of function δ, which play an important role in guaranteeing the correctness and optimality of trust-based routing protocols.

Given a routing graph $G_R(V, E_R, r)$, if a path p (or a link l) exists in G_R, we write $p \in G_R$ (or $l \in G_R$) with a slight abuse of notation. We define:

1. Routing algebra $\mathcal{A}$ is strictly δ-left-monotonic, if for all link $l \in G_R$ and path $p \in G_R$, if $l \circ p \in G_R$, $\delta(l) \neq \bar{0}$ and $\delta(p) \neq \bar{0}$, then $\delta(p) \prec \delta(l \circ p)$.

2. Routing algebra $\mathcal{A}$ is strictly δ-right-monotonic, if for all link $l \in G_R$ and path $p \in G_R$, if $p \circ l \in G_R$, $\delta(l) \neq \bar{0}$ and $\delta(p) \neq \bar{0}$, then $\delta(p) \prec \delta(p \circ l)$.

3. Routing algebra $\mathcal{A}$ is δ-right-isotonic, if for all link $l \in G_R$ and paths $p, q \in G_R$, if $l \circ p, l \circ q \in G_R$ and $\delta(p) \preccurlyeq \delta(q)$, then $\delta(l \circ p) \preccurlyeq \delta(l \circ q)$. $\mathcal{A}$ is strictly δ-right-isotonic if we replace $\preccurlyeq$ by $\prec$ in above statement.

4. Routing algebra $\mathcal{A}$ is δ-left-isotonic, if for all link $l \in G_R$ and paths $p, q \in G_R$, if $p \circ l, q \circ l \in G_R$ and $\delta(p) \preccurlyeq \delta(q)$, then $\delta(p \circ l) \preccurlyeq \delta(q \circ l)$. $\mathcal{A}$ is strictly δ-left-isotonic if we replace $\preccurlyeq$ by $\prec$ in above statement.

Based on the properties we defined for δ function, we can summarize our main results as follows:

Theorem 6.1. *A link-state routing protocol is guaranteed to be correct, if an only if (1) the trust graph is absorptive; and (2) routing algebra $\mathcal{A}$ is δ-right-monotonic, δ-right-isotonic and strictly δ-left-isotonic. A link-state routing protocol is guaranteed to be optimal, if and only if (1) the trust graph is absorptive; and (2) routing algebra $\mathcal{A}$ is δ-right-monotonic, δ-right-isotonic and strictly δ-left-isotonic.*

Theorem 6.2. *A distance-vector routing protocol is guaranteed to be correct, if an only if (1) the trust graph is absorptive; and (2) routing algebra $\mathcal{A}$ is δ-left-monotonic. A distance-vector routing protocol is guaranteed to be optimal, if and only if (1) the trust graph is absorptive; and (2) routing algebra $\mathcal{A}$ is δ-left-isotonic and δ-left-monotonic.*

The detailed proofs of these results are omitted here. Here, we first compare our results with Sobrinho's classic routing algebra [143, 145]. Although our results are also characterized by monotonicity and isotonicity of a operator (δ function here), our δ function is totally different with its counterpart in Sobrinho's routing algebra (i.e., $\otimes$). The difference is technical. Our δ function is related to monoid endomorphisms [56], and therefore, is not included by Sobrinho's routing algebra, which is based on ordered semirings.

6.5.3 Illustrating Examples

Here, we utilize some concrete examples to explain how to use our abstract results described in previous subsection. We already give an example in Example 4 Section 6.2 where the routing protocol is not optimal. It is easy to check that it violates the isotonic property. In what follows, we will utilize Example 5 in Project Description to investigate the case which violates the monotonic property.

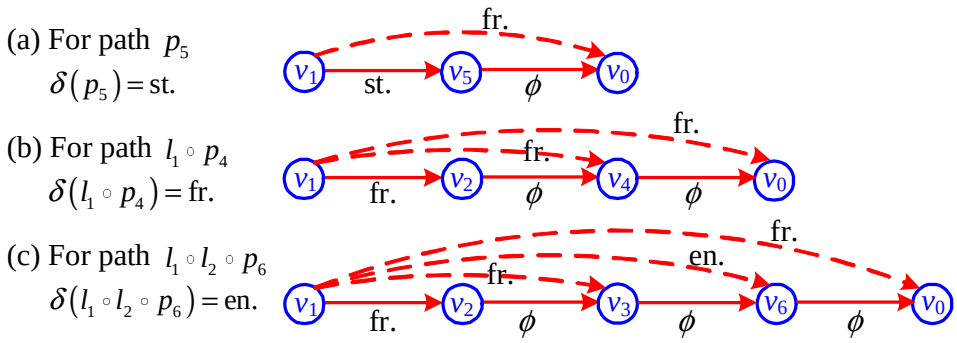


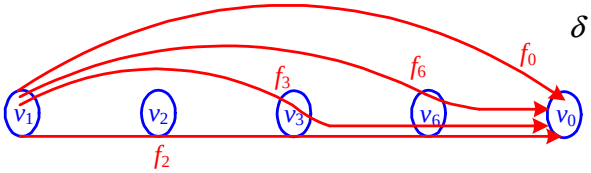
Figure 6-9. Flow graphs for the physical paths from node v_1 to node v_0 in Example 5 in Section 6.2.

We first exhibit the way to formalize this problem with the routing algebra proposed in Def. 6.6. We make the items in $\mathcal{A}$ more concrete as the following:

For the routing algebra $\mathcal{A} = (R, \oplus, \otimes, \bar{0}, \bar{1}, \delta)$, we define

- $R = \{\text{fr., st., en., } \phi\}$ with $\bar{0} = \text{en.}$ and $\bar{1} = \text{fr.}$,
- fr. $\prec$ st. $\prec$ en. $\prec \phi$,
- for all $a, b \in R$, $a \otimes b = \begin{cases} a & \text{when } a \prec b \\ b & \text{when } b \prec a \end{cases}$,
- for all $a, b \in R$, $a \oplus b = \begin{cases} a & \text{when } b \prec a \\ b & \text{when } a \prec b \end{cases}$,
- $\delta : R \times R \mapsto R$ is a function which combines labels on the flow graph of a physical path in G_R into a new one with operators $\otimes$ and $\oplus$ defined above.

Figure 6-9 shows flow graphs for three physical paths from node v_1 to the destination v_0 in Example 5 in Project Description. Here, we introduce a new label ϕ to indicate the links which will not affect the calculation of the trustworthiness of a path. The intuition here is that the trustworthiness of a path only involves the trust relationships between the source node and intermediate nodes on the path, and depends on the least trustworthy intermediate node. Figure 6-10 gives an example of utilizing operators $\otimes$ and $\oplus$ defined above to calculate δ function for the physical path given in Figure 6-9 (c).



$$\delta(l_1 \circ l_2 \circ p_6) = \bigoplus_{f \in F_{l_1 \circ l_2 \circ p_6}} f$$
$$= f_2 \oplus f_3 \oplus f_6 \oplus f_0$$
$$f_6 = \text{en.} \otimes \phi = \text{en.}$$

Figure 6-10. Calculating δ function for the physical path given in Figure 6-9 (c).

It is easy to check that the specification given above leads to the path ranking described in Example 5 in Project Description, which agrees with our intuition on the trustworthy paths. Also our formalization guarantees that for each flow graph $\mathcal{F}$, $(R, \oplus, \otimes, \bar{0}, \bar{1})$ forms a bi-monoid over R. Therefore, we can utilize $\mathcal{BM}$ introduced in Section 6.4 to calculate δ functions. Note that, for Sobrinho's classic routing algebra [143, 145], only labels on physical links are involved in the formalization. Therefore, only hop-by-hop trust relationships can be remained in Sobrinho's classic routing algebra, which are not sufficient for the calculation of the trustworthiness of a whole physical path. Obviously, the introduction of flow graphs in Def. 6.6 makes it possible to include all trust relationships in the formalization. Based on these information, we can easily calculate the trustworthiness of a whole physical path with δ functions.

Next, with the help of this framework, it is easy to check that for both link-state and distance-vector routing approaches, no in-tree can form and therefore the correctness of routing protocols cannot be guaranteed.

6.6 Routing Algebra for Group-Based Trust Environment

6.6.1 Motivating Example

Consider a disaster recovery scenario, the local police force may need to coordinate with fire fighters, military forces, and medical crews by sharing information and communicating with each other regardless of the particular networking protocols that each group uses. See Figure 6-11 for an illustration.

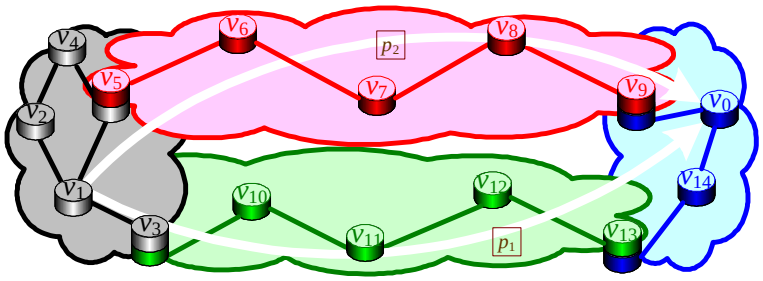


Figure 6-11. Trust in multiple groups.

Obviously, different groups (indicated by different colors in Figure 6-11) like police force, military force, medical crews etc. may have different rules to evaluate and handle trust. Therefore, different trust metrics will be adopted and combined with different trust-based routing protocols. Now node v_1 as a police want to communicate with node v_0, a doctor, through the most trustworthy path. Multiple questions will naturally arise for this task. For example, how node v_1 evaluate the trustworthiness of path p_1 and p_2, since links along each path are measured by different trust metrics? How to compare p_1 with p_2? What kinds of operation rules the nodes connecting different groups (like v_3, v_5, v_9 and v_{13} in Figure 6-11) need to follow in order to make sure there will be no loops in hop-by-hop routing? Or under what conditions the path formed by the next-hop selections of all intermediate nodes is the most trustworthy path as defined by the source node v_1? All these questions call for development of a unified framework to enable the analysis of end-to-end communications over heterogeneous WANETs governed by distinct trust metrics.

In this section, we consider a group-based trust environment, where multiple groups coexist in a WANET. Each group i can be modeled as one routing algebra $\mathcal{A}_i = (R_i, \oplus_i, \otimes_i, \bar{0}_i, \bar{1}_i, \delta_i)$ defined in Section 6.5. The problem here is how to perform end-to-end trust-based routing across multiple groups. To be more focused, we make the following simplifications here:

1. We assume each group is in a homogeneous trust environment, i.e., there exist uniformed rules for the direct trust establishment and indirect trust inference in

each group, and these procedures are independent among different groups. Interactions between different groups only happen in the path selection phase.

2. We assume all groups use the same kind of routing protocols, either distance-vector or link-state routing protocols. The problems emerge from the inter-operations between two kinds of protocols are universal, which are irrelevant to trust metrics, i.e., the theme of this chapter.

In previous section, we use routing algebra to study how to guarantee correct routing and optimal path selection in one group. Our key observation here is that the problems involved in multi-group communications are all related to conversions of trust-based routing metrics between different groups. Therefore, based on previous results, the problems of correct and optimal routing in multi-group environment can be more specifically reformulated as the following: what conditions must the conversions between different routing algebra satisfy in order to guarantee the correct and optimal end-to-end trust-based routing?

6.6.2 Problem Formalization

We first formalize the notion of conversions between different routing algebras.

Definition 6.7. ***[Conversion Function]*** *Given $k+1$ routing algebras $\mathcal{A}_i = (R_i, \oplus_i, \otimes_i, \bar{0}_i, \bar{1}_i, \delta_i)$ for $i = 0, 1, \cdots, k$ as defined in Def. 6.6. Routing algebra $\mathcal{A}_0$ is called the host algebra, if all paths across multiple groups will be measured using the labels in R_0. Correspondingly, routing algebra $\mathcal{A}_j$ ($j \neq 0$) is called the guest algebra, since the label of a path p in group $\mathcal{A}_j$ will be converted to that in the host algebra before path p can be utilized by other groups. We define two conversion functions between $\mathcal{A}_0$ and $\mathcal{A}_j$ as:*

- $\beta_{0\to j} : R_0 \mapsto R_j$,
- $\beta_{j\to 0} : R_j \mapsto R_0$.

Next, we show how to model basic inter-operations between host algebra/group $\mathcal{A}_0$ and guest algebra $\mathcal{A}_1$ by utilizing conversion functions. We only consider two groups here. It is straightforward to extend our discussion to more groups. Refer to Figure 6-12 for an illustration. Here, node v_m belongs to both groups, and acts as bridge router

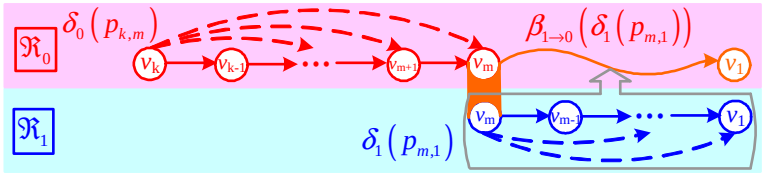


Figure 6-12. Concatenation of paths from different groups.

to connect two groups. Node v_m finds a path, say $p_{m,1}$, to the designation node v_1 in group $\mathcal{A}_1$. Let the label calculated by v_m for path $p_{m,1}$ be $\delta_1(p_{m,1})$. In the distance-vector routing, node v_m needs to advertise the information about path $p_{m,1}$ to its neighbor, say v_{m+1} in group $\mathcal{A}_0$. The label $\delta_1(p_{m,1})$ cannot be directly used because for v_{m+1} label $\delta_1(p_{m,1})$ belongs to different trust metric set R_1, which cannot be understood and further processed by v_{m+1}. Therefore, v_m need first convert $\delta_1(p_{m,1})$ into $\beta_{1\to0}(\delta_1(p_{m,1}))$, i.e., the trust metric used in $\mathcal{R}_0$. Note that, we assume the trust inference process is performed in each group independently. Therefore, node v_{m+1} has no indirect relationships with nodes in group $\mathcal{A}_1$ except v_m. The label of path $p_{m+1,1} = (v_{m+1}, v_m) \circ p_{m,1}$ can be simply calculated as $\delta_0(p_{m+1,1}) = r_{m+1,m} \otimes_0 \beta_{1\to0}(\delta_1(p_{m,1}))$, where $r_{m+1,m} \in R_0$ is the label of link (v_{m+1}, v_m). For link-state routing, source node v_k need calculate the label (i.e., trust metric) $\delta_0(p_{k,1}) \in R_0$ for the whole path $p_{k,1} = p_{k,m} \circ p_{m,1}$. This path can be separated into two parts: subpath $p_{k,m}$ with label $\delta_0(p_{k,m}) \in R_0$ and subpath $p_{m,1}$ with label $\delta_1(p_{m,1}) \in R_1$. Since two labels belong to different trust metric sets, they cannot be directly compared or combined. By utilizing $\beta_{1\to0}$, we can simply compute $\delta_0(p_{k,1}) = \delta_0(p_{k,m}) \otimes_0 \beta_{1\to0}(\delta_1(p_{m,1}))$.

Two important observations can be made through this example. (1) Although there are three operations defined in each routing algebra $\mathcal{A}_i$, i.e., $\oplus_i$, $\otimes_i$ and δ_i, only $\otimes_i$ involves in path label calculations across multiple groups. Therefore, the interactions between conversion functions and $\otimes_i$ operators will play a key role to determine the properties of the whole system (with multiple groups). (2) If we abstract each path contained in one group (like paths $p_{k,m}$ and $p_{m,1}$ in Figure 6-12) as a generalized link,

from the view point of group $\mathcal{A}_0$, conversion function $\beta_{1\to 0}$ assigns each generalized link from other groups a new (generalized) link label. The principles established for one group obviously apply here. Conversion functions can not be arbitrary: in order to guarantee the correctness and optimality of trust-based routing across multiple groups, some constraints on them must be imposed. To characterize those constraints is what we will do next.

6.6.3 Properties of Conversion Functions

Conversion function pair $(\beta_{j\to 0}, \beta_{0\to j})$ represents relationships between host algebra $\mathcal{A}_0$ and guest algebra A_j. Here we first consider the following properties of conversion function pair, which will be used to analyze the correctness and optimality of routing across multiple groups. Recall that for every routing algebra $\mathcal{A}_i$, a canonical pre-order $\preccurlyeq_i$ can be naturally introduced from $\oplus_i$. We define:

(1) Function pair $(\beta_{j\to 0}, \beta_{0\to j})$ is guest-order-preserved if

- $\forall r_a, r_b \in R_0$, if $r_a \preccurlyeq_0 r_b$ we have $\beta_{0\to j}(r_a) \preccurlyeq_j \beta_{0\to j}(r_b)$.

(2) Function pair $(\beta_{j\to 0}, \beta_{0\to j})$ is host-order-preserved if

- $\forall r_a, r_b \in R_j$, if $r_a \prec_j r_b$ we have $\beta_{j\to 0}(r_a) \prec_0 \beta_{j\to 0}(r_b)$.
- $\forall r \in R_0$, $r \preccurlyeq_0 \beta_{j\to 0}(\beta_{0\to j}(r))$.

(3) Function pair $(\beta_{j\to 0}, \beta_{0\to j})$ distributes over $\otimes$ if

- $\beta_{j\to 0}$ is bijective and $\beta_{0\to j} = \beta_{j\to 0}^{-1}$,
- $\forall r_a, r_b \in R_j$, $\beta_{j\to 0}(r_a \otimes_j r_b) = \beta_{j\to 0}(r_a) \otimes_0 \beta_{j\to 0}(r_b)$.
- $\forall r_a, r_b \in R_0$, $\beta_{0\to j}(r_a \otimes_0 r_b) = \beta_{0\to j}(r_a) \otimes_j \beta_{0\to j}(r_b)$.

To facilitate our analysis, we define a universal routing algebra $\mathcal{A} = (R, \oplus, \otimes, \bar{0}, \bar{1}, \delta, \preccurlyeq)$ upon $\mathcal{A}_0$ and $\mathcal{A}_j$, where

- $R = R_0 \cup R_1 \cup \cdots \cup R_k$
- $\otimes : R \times R \mapsto R$ is a function such that $\forall a \in R_i, b \in R_j$:

$$a \otimes b \triangleq \begin{cases} a \otimes_j b & \text{if } i = j, \\ \beta_{i\to j}(a) \otimes_j b & \text{otherwise.} \end{cases}$$

As we discussed before, $\oplus_i$ and δ_i never involve in path calculations across multiple groups, which means operantes of $\oplus$ or δ are always the same. Therefore, we define:

- $a \oplus b = a \oplus_i b$ if $a, b \in R_i$,
- $\delta(p) = \delta_i(p)$ if the whole path p is in group $\mathcal{A}_i$.

For $\mathcal{A}$, $\preccurlyeq$ cannot be introduced from $\oplus$, and therefore should be defined independently:

- $\preccurlyeq$ is an order relation over R such that $\forall a \in R_i, b \in R_j$:

$$a \preccurlyeq b \triangleq \begin{cases} a \preccurlyeq_j b & \text{if } i = j, \\ \beta_{i\to 0}(a) \preccurlyeq_0 \beta_{j\to 0}(b) & \text{otherwise.} \end{cases}$$

Given $\preccurlyeq$, $\bar{0}$ and $\bar{1}$ is defined as follows:

- $\bar{0} \in R$ such that $\forall a \in R - \{\bar{0}\}, a \prec \bar{0}$,
- $\bar{1} \in R$ such that $\forall a \in R - \{\bar{1}\}, \bar{1} \prec a$.

Based on above discussions, we have the following results:

Lemma 9. *Order relation $\preccurlyeq$ defined in universal routing algebra $\mathcal{A}$ is a total pre-order over R if all pairs $(\beta_{j\to 0}, \beta_{0\to j})$ are host-order-preserved.*

Proof: A relation $\preccurlyeq$ on a nonempty set R is

- reflexive, if $\forall a \in R : a \preccurlyeq a$;
- transitive if $\forall a, b, c \in R : a \preccurlyeq b$ and $b \preccurlyeq c \Rightarrow a \preccurlyeq c$;
- total, if $\forall a, b \in R : a \preccurlyeq b$ or $b \preccurlyeq a$.

Recall that a total pre-order is a reflexive, transitive and total relation. In what follows, we prove that the order relation $\preccurlyeq$ defined in universal routing algebra $\mathcal{A}$ has these three properties, respectively.

[Reflexivity] For an arbitrary $a \in R$, because $R = R_0 \cup R_1 \cup \cdots \cup R_k$, there exists an $i \in [0, k]$ such that $a \in R_i$. By the definition of $\preccurlyeq$, $a \preccurlyeq b \triangleq a \preccurlyeq_i b$ when $a, b \in R_i$. Recall that $\preccurlyeq_i$ is a total pre-order over R_i for all $i \in [0, k]$ (i.e., $\preccurlyeq_i$ is reflexive), therefore, $a \preccurlyeq a$ for all $a \in R$.

[Transitivity] For arbitrary $a, b, c \in R$, because $R = R_0 \cup R_1 \cup \cdots \cup R_k$, there exists $i, j, l \in [0, k]$ such that $a \in R_i$, $b \in R_j$ and $c \in R_l$. There are five different cases according to the relationships between i, j and l. Here, we just address the case that i, j, l are all distinct. Other cases are simpler and can be reasoned in a similar way. By the definition of $\preccurlyeq$ and host-order-preservation, $a \preccurlyeq b$ means that $\beta_{i\to 0}(a) \preccurlyeq_0 \beta_{j\to 0}(b)$ when $i \neq j$ and $b \preccurlyeq c$ means that $\beta_{j\to 0}(b) \preccurlyeq_0 \beta_{l\to 0}(c)$ when $j \neq l$. Because $\preccurlyeq_0$ is a total pre-order over R_0 (i.e., $\preccurlyeq_0$ is transitive), we obtain $\beta_{i\to 0}(a) \preccurlyeq_0 \beta_{l\to 0}(c)$, which implies that $a \preccurlyeq c$ for all $a, b, c \in R$.

[Totality] For arbitrary $a, b \in R$, because $R = R_0 \cup R_1 \cup \cdots \cup R_k$, there exists $i, j \in [0, k]$ such that $a \in R_i$ and $b \in R_j$. When $i = j$, since $\preccurlyeq_i$ is total, we have either $a \preccurlyeq_i b$ or $b \preccurlyeq_i a$. By the definition of $\preccurlyeq$, this implies that $a \preccurlyeq b$ or $b \preccurlyeq a$. When $i \neq j$, since $\preccurlyeq_0$ is total and related conversion functions are host-order-preserved, we have either $\beta_{i\to 0}(a) \preccurlyeq_0 \beta_{j\to 0}(b)$ or $\beta_{j\to 0}(b) \preccurlyeq_0 \beta_{i\to 0}(a)$. By the definition of $\preccurlyeq$, this implies that $a \preccurlyeq b$ or $b \preccurlyeq a$. □

6.6.4 Conditions for Correct and Optimal Routing

We first analyze distance-vector routing protocols in multi-group environment:

Theorem 6.3. *Given $k+1$ routing algebras $\mathcal{A}_i$ ($i = 0, \cdots, k$) and conversion function pairs $(\beta_{j\to 0}, \beta_{0\to j})$ ($j = 1, \cdots, k$), a distance-vector routing protocol $\mathcal{R}$ is guaranteed to be correct across multiple groups if (1) $\mathcal{R}$ is correct in each routing algebra; and (2) all pairs $(\beta_{j\to 0}, \beta_{0\to j})$ are host-order-preserved. A distance-vector routing protocol $\mathcal{R}$ is guaranteed to be optimal across multiple groups if (1) $\mathcal{R}$ is optimal in each routing algebra; and (2) all pairs $(\beta_{j\to 0}, \beta_{0\to j})$ are host-order-preserved and guest-order-preserved.*

Proof: **[Part 1: Correctness]** To prove the correctness of $\mathcal{R}$ across multiple groups, the key point here is to show that Condition (2) is sufficient to preserve strict δ-monotonicity across the algebras. We assume that a node find a path p with label $\delta_i(p) \in R_i$ from algebra $\mathcal{A}_i$, and the path is extended to an link l with a label $r_j(l) \in R_j$. With the assumption that all routing algebras are strictly δ-monotonic and Condition (2)

holds, we need prove the extended path $l \circ p$ has a strictly lower preference than the initial path. When $i = j$, i.e., the initial path is extended into the same routing algebra $\mathcal{A}_i$, the proof is trivial. Therefore, we focus on the case $i \neq j$. Since $\mathcal{A}_j$ is strictly δ-monotonic, $\delta_j(p) \prec_j \delta_j(l \circ p)$, which can be wrote as $\beta_{0\to j}(\beta_{i\to 0}(\delta_i(p))) \prec_j r_j(l) \otimes_j \beta_{0\to j}(\beta_{i\to 0}(\delta_i(p)))$. Then by the host-order-preserving property, we have $\beta_{j\to 0}(\beta_{0\to j}(\beta_{i\to 0}(\delta_i(p)))) \prec_0 \beta_{j\to 0}(r_j(l) \otimes_j \beta_{0\to j}(\beta_{i\to 0}(\delta_i(p))))$ and $\beta_{i\to 0}(\delta_i(p)) \preccurlyeq_0 \beta_{j\to 0}(\beta_{0\to j}(\beta_{i\to 0}(\delta_i(p))))$. By the definition of $\otimes$, we obtain $\beta_{i\to 0}(\delta_i(p)) \prec_0 \beta_{j\to 0}(r_j(l) \otimes \delta_i(p))$. Finally, by definition of $\preccurlyeq$, we obtain $\delta_i(p) \prec r_j(l) \otimes \delta_i(p)$.

[Part 2: Optimality] We need show that Condition (2) guarantees the preservation of the δ-right-isotonicity property across the multiple groups, i.e., for two paths $p \in R_i$ and $q \in R_j$, if $\delta_i(p) \preccurlyeq \delta_j(q)$, then for $l \in R_k$, $\delta_k(l \circ p) \preccurlyeq_k \delta_k(l \circ q)$. In the universal algebra $\mathcal{A}$, this is equivalent to $\delta_k(l) \otimes \delta_i(p) \preccurlyeq \delta_k(l) \otimes \delta_j(q)$. There are five different cases according to the relationships between i, j and k. Here, we just address the case that i, j, k are all distinct. Other cases are simpler and can be reasoned in a similar way. From $\delta_i(p) \preccurlyeq \delta_j(q)$, we have $\beta_{i\to 0}(\delta_i(p)) \preccurlyeq_0 \beta_{j\to 0}(\delta_j(q))$, therefore $\beta_{0\to k}(\beta_{i\to 0}(\delta_i(p))) \preccurlyeq_k \beta_{0\to k}(\beta_{j\to 0}(\delta_j(q)))$. For link l: $\delta_k(l) \otimes_k \beta_{0\to k}(\beta_{i\to 0}(\delta_i(p))) \preccurlyeq_k \delta_k(l) \otimes_k \beta_{0\to k}(\beta_{j\to 0}(\delta_j(q)))$. Therefore, $\delta_k(l) \otimes \delta_i(p) \preccurlyeq \delta_k(l) \otimes \delta_j(q)$. □

Theorem 6.4. *Given $k+1$ routing algebras $\mathcal{A}_i$ ($i = 0, \cdots, k$) and conversion function pairs $(\beta_{j\to 0}, \beta_{0\to j})$ ($j = 1, \cdots, k$), a link-state routing protocol $\mathcal{R}$ is guaranteed to be correct and optimal across multiple groups if (1) $\mathcal{R}$ is correct and optimal in each routing algebra; and (2) all pairs $(\beta_{j\to 0}, \beta_{0\to j})$ are host-order-preserved, guest-order-preserved and distributive.*

Proof: Based on the proof of Theorem 6.4, here we just need show that given one more condition, i.e., distributivity, we can guarantee (1) the isotonicity property across the algebras and (2) the associativity of $\otimes$ in universal algebra $\mathcal{A}$.

[Part 1: Left-isotonicity] We already show the left-isotonicity in the proof of theorem 6.4. Here, we just need prove for three paths p, q, l where $\delta_i(p) \in R_i$, $\delta_j(q) \in R_j$

and $\delta_k(l) \in R_k$, if $\delta_i(p) \preccurlyeq \delta_j(q)$, then $\delta_i(p) \otimes \delta_k(l) \preccurlyeq \delta_j(q) \otimes \delta_k(l)$. We focus on the case when i, j, k are all distinct. From $\delta_i(p) \preccurlyeq \delta_j(q)$, we have $\beta_{i\to 0}(\delta_i(p)) \preccurlyeq_0 \beta_{j\to 0}(\delta_j(q))$. From $\mathcal{A}_0$'s left-isotonicity, we have $\beta_{i\to 0}(\delta_i(p)) \otimes_0 \beta_{k\to 0}(\delta_k(l)) \preccurlyeq_0 \beta_{j\to 0}(\delta_j(q)) \otimes_0 \beta_{k\to 0}(\delta_k(l))$. Because conversion function pairs are distributive, we obtain $\beta_{i\to 0}(\delta_i(p) \otimes_i \beta_{0\to i}(\beta_{k\to 0}(\delta_k(l)))) \preccurlyeq_0 \beta_{j\to 0}(\delta_j(q) \otimes_j \beta_{0\to j}(\beta_{k\to 0}(\delta_k(l))))$. By the definition of $\otimes$, we obtain $\delta_i(p) \otimes \delta_k(l) \preccurlyeq \delta_j(q) \otimes \delta_k(l)$.

[Part 2: Associativity of $\otimes$] For three paths p, q, l where $\delta_i(p) \in R_i$, $\delta_j(q) \in R_j$ and $\delta_k(l) \in R_k$, we focus on the case when i, j, k are all distinct. We have

$$
\begin{aligned}
& (\delta_i(p) \otimes \delta_j(q)) \otimes \delta_k(l) \\
= \; & (\delta_i(p) \otimes_i \beta_{0\to i}(\beta_{j\to 0}(\delta_j(q)))) \otimes \delta_k(l) \\
= \; & (\delta_i(p) \otimes_i \beta_{0\to i}(\beta_{j\to 0}(\delta_j(q)))) \otimes_i \beta_{0\to i}(\beta_{k\to 0}(\delta_k(l))) \\
= \; & \delta_i(p) \otimes_i (\beta_{0\to i}(\beta_{j\to 0}(\delta_j(q)))) \otimes_i \beta_{0\to i}(\beta_{k\to 0}(\delta_k(l)))) \\
& \delta_i(p) \otimes (\delta_j(q) \otimes \delta_k(l)) \\
= \; & \delta_i(p) \otimes (\delta_j(q) \otimes_j \beta_{0\to j}(\beta_{k\to 0}(\delta_k(l)))) \\
= \; & \delta_i(p) \otimes_i \beta_{0\to i}(\beta_{j\to 0}((\delta_j(q) \otimes_j \beta_{0\to j}(\beta_{k\to 0}(\delta_k(l)))))) \\
= \; & \delta_i(p) \otimes_i (\beta_{0\to i}(\beta_{j\to 0}(\delta_j(q)))) \otimes_i \beta_{0\to i}(\beta_{k\to 0}(\delta_k(l))))
\end{aligned}
$$

Therefore, $(\delta_i(p) \otimes \delta_j(q)) \otimes \delta_k(l) = \delta_i(p) \otimes (\delta_j(q) \otimes \delta_k(l))$. □

6.7 Chapter Summary

In this chapter, we develop a formal framework and theory to investigate the correctness, optimality, inter-operativity of trust-based routing protocols for WANETs. Our results obtained here can be extended in two ways. (1) For indirect trust inference problems, we only consider the situation when all trusts in a WANET are transitive. When transitive and non-transitive trust coexist in a WANET, a new algebraic structure for a combined trust metric is needed and consequently new algorithm should be designed to infer indirect trust under non-transitive trust constraints. (2) From routing's point of view, in our framework we only consider topology-based routing protocols. We

can extend our study to location-based routing like geographic routing, which is popular for WANETs. Also in this chapter we restrict ourselves to unicast routing. Obviously, the trust metrics for multicast, broadcast, and anycast are totally different from that for unicast, and the concept of path selection will be replaced by tree selection. Therefore, these topics should be further investigated.

CHAPTER 7
SECURE NETWORK PERFORMANCE OF LARGE-SCALE WANETS

7.1 Chapter Overview

The growth of modern communication networks, such as the Internet and wireless cellular systems, over the last decade has surpassed many expectations. Indeed, going back in time to the origins of these networks, it would have been hard to imagine the importance and scale to which these networks have developed. Now, projecting into the future, we strongly believe that this trend will continue, if not accelerate. Hence, the communication devices and protocols of today must be capable of operating with the same efficiency in the very large-scale networks of the future. This highlights the need for asymptotic analysis on a network and its corresponding protocol design, which characterizes the asymptotic behaviors of network performance as its size n grows. This is especially the case for wireless ad hoc networks (WANETs) which offer communications over a shared wireless channel without any pre-existing infrastructure, since more effort needs to be made to harmonize the behavior of different participants and manage distributed network resources to support end-to-end (e2e) communication demands compared to the network with infrastructure. Obviously, this kind of unavoidable coordination overhead, which may be tolerable in small-scale networks, is possible to become dominant factors in large-scale networks and should be quantitatively analyzed.

Since both throughput and delay are important network performance metrics, significant effort in the last few years has been devoted to understanding the scaling laws on throughput and delay and their relationship in WANETs. In their seminal work, Gupta and Kumar [60] show that the per-flow throughput capacity for static WANETs scales as $\Theta(1/\sqrt{n \log n})$ (refer to Appendix-B for the standard asymptotic notation used throughout the chapter) under the assumption that nodes with common transmission range are randomly distributed. Note that this work [60] implicitly uses a fluid model

for establishing throughput scaling. Later work by Kulkarni and Viswannath [90] consolidates the result in [60] with an explicit constant-packet-size model. Following the same methodology, the corresponding delay of $\Theta(\sqrt{n/\log n})$ and the complete throughput-delay tradeoffs of static WANETs are first obtained in [48]. Recently, with the percolation theory, Franceschetti et al. [46] show that the per-flow throughput of $\Theta(1/\sqrt{n})$ is achievable if each node can adjust its transmission range through power control. The throughput-delay tradeoffs under different mobility models are also studied in the literature (e.g., [11, 48–50, 100, 101, 120, 140, 154]).

A drawback common to all the above results is the neglect of security requirements, which are receiving growing attention in recent years because many large-scale WANETs are expected to be deployed in hostile scenarios such as military and homeland security operations [177]. It is known that security always comes with a price, as securing communications against the adversary typically consume more network resources in terms of bandwidth and/or hardware capacities. This price may be tolerable in small-scale WANETs, but it may dominate the consumption of scarce network resources in large-scale WANETs. This situation makes the investigation of throughput-delay tradeoffs with security requirements in large-scale WANETs an important open challenge.

Although security requirements in WANETs are application-dependent, in this chapter we focus on the most critical and fundamental one which reflects the distinct nature of WANETs and enables analytical tractability, that is, we require that wireless communications should operate on secure links whenever necessary. A WANET can be informally visualized as a group of wireless communication devices/nodes, held by users coming together spontaneously to form a network for a common purpose (e.g., emergency response). Some keying materials for primary security associations (SAs), which we will formally define later (cf. Section 7.2.1), are already pre-configured in communication devices based on the trust relationships among the persons involved.

The problem is how to exploit those primary SAs to provide secure communications for arbitrary node pairs when needed. Neighbor authentication or securing the physical link, which provides hop-by-hop security, is the first step for providing e2e secure communications in all kinds of networks. This is especially crucial for WANETs since every node needs to act as a router to forward packets for others. If the node cannot authenticate its neighbors[1] , how can it trust them to handle its packet correctly? Obviously, neighboring nodes with primary SAs can authenticate each other directly with pre-configured keying materials, and the physical links between them can be secured accordingly. Since the number and the distribution of primary SAs are determined by the embedded social network (e.g., trust relationship) of users, a node may not have primary SAs with any of its neighbors. In fact, the probability that a node shares a primary SA with any other node, i.e., p_f, will be very small in practice when the node population n in the WANET increases. In this case, if the physical link still need be secured, secure link augmentation (SLA) operations[2] are required and performed with the help of physically connected common friends of two end nodes of this physical link.

When $p_f < 1$, there is in general network performance degradation because we require all communications operate on secure links, and some network resources, i.e., the unsecured links cannot be utilized compared to WANETs without secure requirements. Although we can obtain more derived secure links with SLA, network

[1] We say two nodes are friends when they have a primary SA. We say two nodes are neighbors if their Euclidean distance is no greater than the transmission range r_n. There is a physical link between any two neighboring nodes, and this link can be secured with the primary SA if the neighboring nodes are also friends. We call this kind of secure links as primary secure links. A link can also be secured with the help of other authenticated neighbors; we call this kind of secure links derived secure links.

[2] SLA here means the procedure of securing a physical link between two neighboring nodes which are not friends. A detailed description of SLA operations is given in Scheme 2 in Section 7.5.1.

resources consumed by SLA is another kind of security cost which should be taken into consideration. Therefore, it is natural to ask: what is the price of security (performance degradation) we have to pay in WANETs? Can we design a protocol to achieve the optimal secure network performance or minimize the price of security? In this chapter, we answer these questions with rigorous analysis based on reasonable assumptions on WANETs. We formally characterize the tradeoffs between key pre-distribution related to p_f and secure network performance. Our results show that the minimal price of security with SLA is strictly smaller than that without SLA, which theoretically necessitates SLA operations in WANETs with security requirements. We also design two schemes to achieve the minimal price of security with or without SLA, respectively. Furthermore, these schemes provide several important insights on protocol design for secure communications in WANETs as follows: (1) It is unnecessary and even harmful to think that in order to achieve the minimal price of security, we have to obtain as many derived secure links as possible. In fact, the physical links need be secured with SLA are few and should be carefully selected. (2) Our schemes show that it is possible to construct the secure backbone and select the physical links that need be secured in a totally distributed fashion with negligible communication overhead, and this "secure infrastructure" is unrelated to source-destination (S-D) pairs and can be reused again and again. (3) Although in general secure network performance degrades with p_f, with or without SLA, one important exception we find is that when p_f is $\Omega\left(1/\log n\right)$, the secure throughput remains at the Gupta and Kumar bound of $\Theta\left(1/\sqrt{n\log n}\right)$ packets/timeslot, wherein no security requirements are enforced on WANETs. This implies that even when p_f goes to zero as the network size becomes arbitrarily large, it is still possible to build throughput-order-optimal secure WANETs, which is of practical interest since in many practical large-scale WANETs, p_f is very small.

7.2 Background and Related Work

The impact of security requirements on the performance of WANETs is largely untouched in the literature with only a few exceptions [14, 168]. In this section, we first review some keying schemes and secure operations related to the fulfillment of our security requirements and then present recent results on secure connectivity and throughput, respectively. We compare these results with ours obtained in this chapter and point out our own contributions.

7.2.1 On Pre-Distribution of Keying Materials/SAs

When we say two nodes have a primary SA, we mean that two nodes trust each other in the sense that either a symmetric key is shared between them or they know each others' authentic public keys. We further assume that SAs are always symmetric because trust relationship is symmetric in nature [39, 168]. The concept of SA here is closely related to the topic of pre-distributed key establishment and management in the security protocol design [23, 26, 38, 71]. In what follows, we summarize some representative schemes proposed in the literature and demonstrate that the parameter p_f is a good abstraction of trust relationships among network nodes regardless of the implementation details of keying schemes.

Eschenauer and Gligor's Key Pool Scheme [38]: Before the nodes are deployed, an off-line trust authority (TA) will provide a large key pool of size P. Each node randomly picks k different keys from this key pool. Therefore, two neighboring nodes have a primary SA if they share at least one common key in the key pool with probability p_f, which is given as

$$p_f = 1 - \frac{\binom{P-k}{k}}{\binom{P}{k}} = 1 - \frac{((P-k)!)^2}{(P-2k)!\,P!}, \tag{7–1}$$

where the second equality holds for $P > 2k$.

Chan, Perrig and Song's Random-Pairwise Key Scheme [26]: Each node identity (ID) is paired with f other randomly selected distinct node IDs, and a pairwise key is pre-generated for each pair of nodes. The key is stored in both nodes' key ring along

with the ID of the other node that knows the key. Therefore, the probability p_f of two selected nodes sharing a primary SA is directly given by

$$p_f = f/n\,. \tag{7–2}$$

Hubaux, Čapkun and Buttyán's Self-Organized Public-Key Scheme [23, 71]: Like in PGP [182], each node's public and private keys are created by the node itself. Unlike in PGP, where certificates of the public keys are mainly stored in centralized certificate repositories, certificates in this scheme are stored and distributed by the nodes in a fully self-organized manner. For simplicity, we assume that each node has f friends and it has already stored the certificates of its friends' public keys. Therefore, the probability p_f that a node can directly authenticate one of its neighbor is also given by Eq. (7–2).

Multiple Trust Authority Scheme [23, 71]: In this scheme, there are P secure domains. In each secure domain, there exists one off-line trust authority, which creates public-private key pairs for each node belonging to its domain. Each node belongs to k randomly selected domains. Therefore, two nodes have a primary SA if they belong to the same domain, with probability p_f, which is given in Eq. (7–1).

Note that the processes of neighbor authentication and pairwise key establishment based on primary SAs and the ways to secure more physical links with SLA have been presented in a unified approach in our previous work (cf. [168, Section II.B]), which is omitted here due to space constraints. Here, we just emphasize that we assume a homogeneous trust model, i.e., each pair of nodes has a primary SA with the same probability p_f, and p_f's are pairwise independent for each pair of nodes.

To sum up, we have the following observations from the schemes discussed above. First, we need keep p_f as small as possible. A larger p_f requires more memory space for storing keying materials in each node, and when that node is compromised, the revealed keys will have a larger impact on network security. Therefore, with the same

network performance, we are interested in the scheme with the minimal p_f. Second, all these schemes assume homogeneous and independent trust relationships among network nodes, i.e., every node pair has the same probability p_f of sharing a primary SA which happens independently of other node pairs. Although these two properties are not necessarily valid in all practical situations (cf. [168, Section II.A.2]), we adhere to these assumptions throughout the chapter for the analytical tractability.

7.2.2 On Secure Connectivity

Secure connectivity here refers to the requirement that there should exist a secure path connecting arbitrary node pairs[3] , which indicates the availability of secure communications. Primary results analyzing secure connectivity have been presented in [38] based on an approximation model for sensor networks. More precise analysis are given in [36, 70]. These results suffer from two main drawbacks. First, they assume that in order to achieve the secure connectivity, the network should at least be connected with primary secure links, which is not necessarily the case. Second, they only study the case under certain requirement on p_f for a given $r_n = \Theta(\sqrt{\log n/n})$, the common transmission range.

In our previous work [168], we overcome these limitations by giving a thorough study on r_n-p_f tradeoffs with the secure connectivity constraint under the assumptions that n nodes are randomly distributed in a unit area and that primary SAs are pre-distributed as described above. Our main results are as follows:

- The network is securely connected without SLA or with one-hop SLA when $p_f \cdot n \cdot \pi r_n^2 = \Omega(\log n)$;

[3] A secure path consists of consecutive secure links. Of course this requirement is reasonable only when the S-D pair is in the same trust domain and there exists at least one physical path connecting the S-D pair. Therefore, it is necessary to have $p_f = \Omega(\log n/n)$ and $r_n = \Omega(\sqrt{\log n/n})$ (cf. [168, Section II.C]). In what follows, we always assume that it is the case.

- The network is securely connected with k_n-hop SLA when $p_f \cdot n \cdot \pi r_n^2 \geq c$, where $c = \Theta(1)$ and $k_n = O(\log n)$;
- It is impossible for the network to be securely connected when $p_f \cdot n \cdot \pi r_n^2 < c$ for the routing-security dependency loop problem, where $c = \Theta(1)$.

We want to keep r_n and p_f as small as possible. However, the above results show that we cannot minimize both to maintain secure connectivity. There are tradeoffs between r_n and p_f under the secure-connectivity constraint, but we can achieve a better r_n-p_f tradeoff when multi-hop SLA is utilized.

The problem left here is that secure connectivity alone is not a good performance metric for secure WANETs. From the above results, if we only consider secure connectivity, then for the situation of $p_f < 1$, we can trivially achieve the secure connectivity by taking a larger r_n. However, this will lead to a dramatic reduction in achievable throughput (cf. Section 7.6). Therefore, when we say there exists a secure path for a S-D pair, we must also ask what the secure throughput can be supported. Otherwise, the existence of secure paths is meaningless because the throughput that can be supported may be very low. In this chapter, we continue our investigation on characterizing those network performance metrics for secure WANETs.

7.2.3 On Secure Throughput

Recently, Bhandari and Vaidya [14] suggest that their techniques developed for studying the capacity of multi-channel WANETs with random (c, f) assignment [13] can be utilized to analyze the secure throughput with the key pool scheme [38]. For multi-channel WANETs with random (c, f) assignment, there are c channels of equal bandwidth available. Each node can only work on a subset of f channels which is pre-assigned from c channels randomly. It can be mapped into secure WANETs with each node randomly selecting f keys from a key pool of size c. The ability of two neighboring nodes switching on a common channel can be viewed as having a common key to secure their physical link. Based on this idea, they obtain the secure throughput of $\Theta\left(\sqrt{\frac{p_f}{n \log n}}\right)$ for $p_f = \Omega(1/\log n)$. To the best of our knowledge, this is the only work

contributing to this topic. Our work is done concurrently with and independently of the work in [14] and differentiates itself from [14] as follows.

First of all, it is worth noting that there are some fundamental differences between multi-channel WANETs and secure WANETs. If two neighboring nodes in a multi-channel WANET do not share a common assigned channel, there exists no physical link between them. In contrast, if two neighboring nodes in a secure WANET do not share a common key, it only means that they cannot establish a primary secure link. The physical link still exists and can be secured and utilized if they can find a connected common friend to help them. Also note that, if two concurrent transmission pairs in a multi-channel WANET use different channels, they will not interfere with each other. By comparison, even two concurrent transmission pairs in a secure WANET use different keys to secure their transmissions, it is still possible for them to interfere with each other. By taking these differences into consideration, we show that, when SLA is utilized, a secure throughput of $\Theta(1/\sqrt{n \cdot \log n})$ is achievable, which is much higher than the result in [14] for $p_f = \Omega(1/\log n)$.

Second, we adopt different models more suitable for secure WANETs. Following previous works, e.g., [49, 60, 100, 101, 140, 154], the results in [14] and [13] are implicitly based on the fluid model, in which the packets are allowed to be arbitrarily small as $n \to \infty$. In contrast, we follow [48, 50, 90] and explicitly assume the constant-packet-size model, where the packet size remains constant, i.e., does not scale down with n. Although the analysis of the constant-packet-size model is much harder than that of the fluid model [50], we still prefer the former since in reality the packet size does not change when more nodes are added to the network. Furthermore, for a WANET with secure requirements, each packet includes a message authentication code of at least constant size for cryptographic operations. This security overhead on the packet level can be ignored asymptotically only with the constant-packet-size model.

The adoption of the constant-packet-size model also facilitate our analysis on packet delays [48, 50, 90].

Finally, we utilize different techniques to derive more general results. We demonstrate how to take advantage of the considerable similarity between our problem and existing work on parallel computing and leverage the results on faulty arrays to obtain scaling laws on secure throughput and delay. Our results actually apply to all possible p_f's when $p_f = \Omega(\log n/n)$.

7.3 System Assumptions and Main Results

7.3.1 Random Network Model of WANETs

7.3.1.1 Node distribution

We are mainly interested in static WANETs or networks with slow mobility, in which the round-trip time (RTT) of a packet between any S-D pair is much smaller than the time scale of network topology changes. We do not consider the WANETs with rapid topology changes because in this case the overhead of maintaining end-to-end paths will dominate wireless transmissions, while in this chapter we focus on the overhead introduced by security requirements and its impact on data transmissions.

We model the node positions as a random point process as follows. Let $\{X_1, X_2, \cdots\}$ be independent and uniformly distributed random points on a bounded region A in the plane. Given a positive integer n, the point process $\{X_1, X_2, \cdots, X_n\}$ is referred to as the uniform n-point process on A and denoted by $\mathcal{X}_n$. Given a positive number $\lambda = \frac{n}{|A|}$, let $Po(\lambda)$ be a Poisson random variable with parameter λ, independent of $\{X_1, X_2, \cdots\}$. Then it can be shown that the point process $\{X_1, X_2, \cdots, X_{Po(\lambda)}\}$ is a Poisson point process with mean λ on A [130, Section 1.7, p.18] and is denoted by $\mathcal{P}_\lambda$. It is assumed throughout the chapter that for any n or λ, the random point processes $\mathcal{X}_n$ and $\mathcal{P}_\lambda$ are coupled in this manner. $\mathcal{V}_n$ is shorthand either for $\mathcal{X}_n$ or $\mathcal{P}_\lambda$. Recall that $\mathcal{P}_\lambda$ is characterized by the following spatial independence property : if $A_1, A_2, \cdots, A_m$ are arbitrarily disjoint regions of A, then the numbers of points in $\mathcal{P}_\lambda$ on $A_1, A_2, \cdots, A_m$

are mutually independent Poisson random variables with mean $\lambda|A_1|, \lambda|A_2|, \cdots, \lambda|A_m|$, respectively. Because of this extreme independence property, it is often easier to work with $\mathcal{P}_\lambda$ rather than $\mathcal{X}_n$. Therefore, we shall often start by proving limit theorems about $\mathcal{P}_\lambda$ as $\lambda \to \infty$ and then deduce results for $\mathcal{X}_n$ from these. The rationale behind this de-Poissonization technique (cf. [130, Section 2.5, p.37]) is that, given that there are exactly k points of $\mathcal{P}_\lambda$ in a region A, these k points are independently and uniformly distributed in A. Thus, $\mathcal{X}_n$ can be well approximated by $\mathcal{P}_\lambda$ as n or λ tends to infinity. Note that the results obtained in this chapter apply to both $\mathcal{X}_n$ and $\mathcal{P}_\lambda$ (i.e., $\mathcal{V}_n$).

We further assume that A is a torus[4] with a unit area and take $\lambda = n$ as $|A| = 1$, which corresponds to the dense network model [48, 60] because the area is fixed and the density of nodes increases with the network size n. Another possible model that can be used to study the asymptotic behavior of large-scale WANETs is to keep the node density λ as a constant and let the area of A increase linearly with n, which corresponds to the extended network model [34, 114]. In this chapter we concentrate on the dense network model just for fair comparisons, as most known results about WANETs without secure requirements are based on this model [48, 50, 60, 90]. We note that, however, our results can also be applied to the extended network model by utilizing the scaling technique introduced in [114, Section 2.2, p.28].

7.3.1.2 Interference models

We adopt the following two widely used models [60] to describe the necessary and sufficient condition for the successful reception of a transmission over one hop. In what follows, we assume that time is slotted for packetized transmissions and that only $O(1)$ packets can be transmitted per timeslot, i.e., our analysis is explicitly

[4] We assume the torus to avoid border effects, which otherwise complicates the analysis. We note, however, that the results in the chapter hold for square, disk or any other shapes of practical interests.

based on the constant-packet-size model. A transmitter sends data at a constant rate of W packets/ timeslot for a successful transmission, and zero for an unsuccessful transmission, where $W = O(1)$.

Protocol Model: We assume that all nodes use a common range r_n for their transmissions, and a transmission from node i to node j is successful if and only if $d_{ij} \leq r_n$ and $d_{kj} \geq (1+\Delta)r_n$ for any other simultaneous transmitter, say node k. Here, d_{ij} is the distance between nodes i and j, and Δ is a positive constant independent of n.

Physical Model: We assume that all nodes use a common power P_n for their transmissions, a transmission from node i to node j is successful if and only if for a concurrent transmitter set $\mathcal{S}$, we have the signal to interference plus noise ratio (SINR) at receiver j, denoted as $SINR_{ij}$, satisfying

$$SINR_{ij} = \frac{P_n \cdot G_{ij}}{N_0 + \sum_{k \in \mathcal{S} \setminus \{i\}} P_n \cdot G_{kj}} \geq \beta.$$

Here, β is the SINR threshold, N_0 represents the ambient noise, and G_{ij} denotes the link gain on link $i \to j$. We use $G_{ij} = d_{ij}^{-\alpha}$ for simplicity, where $\alpha > 2$ is the path loss exponent.

We mainly focus on the protocol model in this chapter for a cleaner presentation of the key ideas. We also show that the same results on secure WANETs can be obtained under the physical model in Appendix-D.

7.3.1.3 Traffic pattern

Similar to previous works [48–50, 90], we consider the uniform-permutation traffic pattern, i.e., there are n flows/sessions and each node is a source node for only one unicast session and a destination node for another unicast session. Suppose that the source node $i \in \{1, \cdots, n\}$ has data intended for destination node $d(i)$, and then $d(1), d(2), \cdots, d(n)$ is a random permutation of $1, 2, \cdots, n$, where $d(i) \neq i$ for all i.

7.3.2 Network Performance Metrics

7.3.2.1 (Secure) throughput

A per-flow throughput τ is said to be feasible/achievable if every node can send at least at a rate of τ packets/ timeslot to its chosen destination. We denote by $T(n)$, the maximum feasible throughput as the throughput capacity for the network. When security requirements are enforced, we define secure throughput as the maximum throughput that can be supported on secure paths for all S-D pairs. Note that when SLA is utilized, the traffic overhead for constructing secure paths will be excluded from the total traffic on secure paths, so secure throughput is only measured as the data rate achieved on the application layer.

7.3.2.2 (Secure) delay

The delay of a packet is the time it takes the packet to reach the destination after it leaves the source. We do not take the queueing delay at the source into account, since our interest is in the network delay. We are interested in the expectation of the average packet delay over all S-D pairs and all random network configurations, which is denoted as $D(n)$ throughout the chapter. Note that for secure WANETs, the secure delay is measured only on secure paths. If SLA is utilized, the time required to construct the secure path for the packet going through this path will be calculated as a part of secure delay of that packet.

7.3.2.3 The price for security

The loss on the secure throughput or the increase on the secure delay compared to WANETs without secure requirements will be defined as the price for security.

7.3.3 Main Results of Our Work

The goal of this chapter is to study the impact of r_n and p_f on the secure throughput and delay of random networks defined in Section 7.3.1. The following results hold with

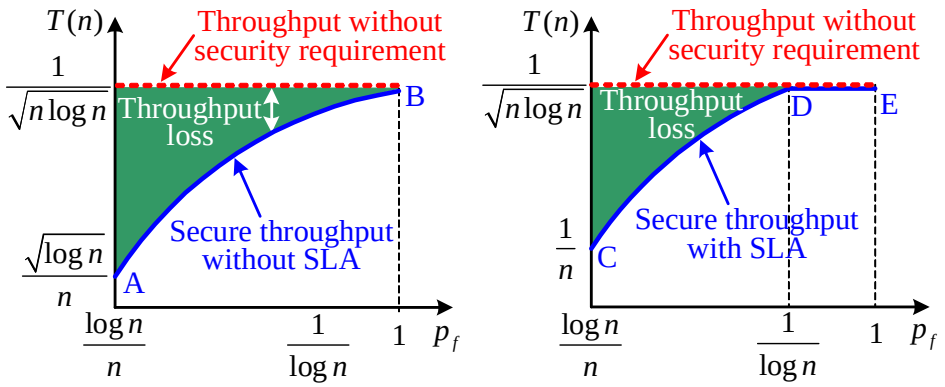


Figure 7-1. Impact of security requirements on throughput scaling in random networks. The shaded area represents throughput loss due to secure requirements. The scales of the axes are in terms of the orders in n.

high probability (w.h.p.)[5] when the network size $n \to \infty$. Here we only consider the situation when $p_f = \Omega(\log n/n)$ and $r_n = \Omega(\sqrt{\log n/n})$ (cf. Footnote 3).

Theorem 7.1. *When $p_f = \Omega(\log n/n)$, the secure throughput without SLA is $T(n) = \Theta\left(\sqrt{\frac{p_f}{n \cdot \log n}}\right)$ packets/timeslot (segment A-B in Figure 7-1), and the secure delay is $D(n) = \Theta\left(\sqrt{\frac{n \cdot p_f}{\log n}}\right)$.*

Theorem 7.2. *(i) When $p_f = \Omega(\log n/n)$ and also $p_f = O(1/\log n)$, the secure throughput with SLA is $T(n) = \Theta(\sqrt{p_f/n})$ packets/timeslot (segment C-D in Figure 7-1), and the corresponding delay is $D(n) = \Theta(\sqrt{n/p_f})$; (ii) When $p_f = \Omega(1/\log n)$, the secure throughput with SLA is $T(n) = \Theta(1/\sqrt{n \cdot \log n})$ packets/timeslot (segment D-E in Figure 7-1), and the corresponding delay is $D(n) = \Theta(\sqrt{n/\log n})$.*

Comparing Theorem 7.1 with Theorem 7.2, we can conclude that, SLA is necessary because in general it can increase the achievable throughput as a factor of $\Theta(\sqrt{\log n})$. However, it does not mean that we should try to secure all physical links. Remember that SLA also incurs extra communication overhead, and the scheme we design to achieve the throughput in Theorem 7.2 shows that we need carefully choose

[5] Here w.h.p. refers to a probability at least $1 - \epsilon(n)$, for a function $\epsilon(n)$ going to 0 with $n \to \infty$.

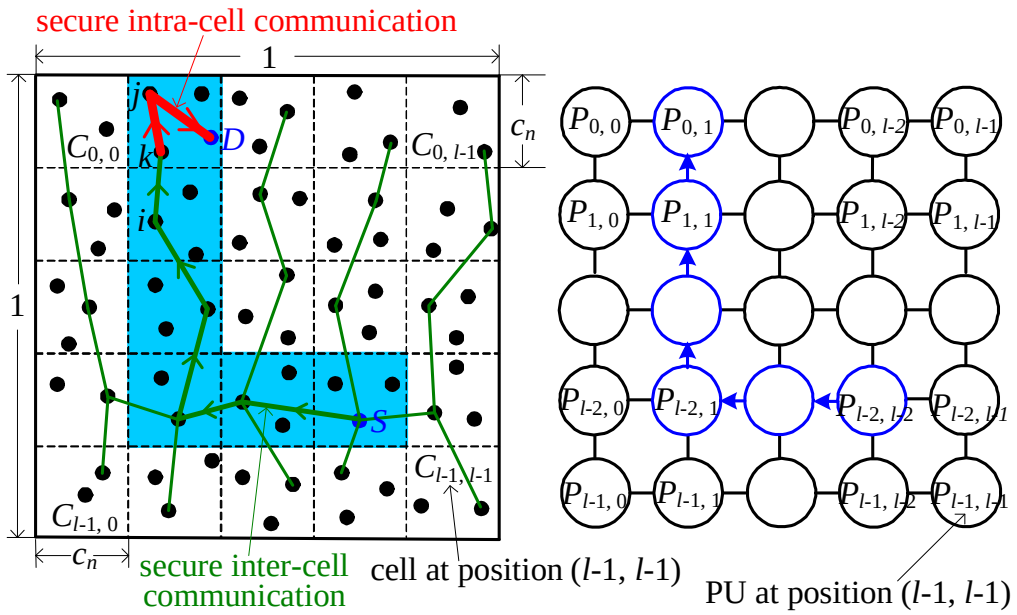
links to be secured with the help of friends in order to guarantee that the benefits from SLA always exceed its costs (cf. Section 7.5).

In order to calculate the price of security, we recall the results on network performance of WANETs without security requirements [48, 50, 90], which can be summarized as the following Theorem.

Theorem 7.3. *The throughput capacity of WANETs without security requirements is* $T(n) = \Theta(1/\sqrt{n \cdot \log n})$ *packets/timeslot (dashed lines in Figure 7-1), and the corresponding delay is* $D(n) = \Theta(\sqrt{n/\log n})$. [6]

From Theorem 7.3, we can find that the price of security mainly exhibits in the loss of throughput. Figure 7-1 gives an illustration of the comparison on throughput capacity with or without security requirements. It is worth noting that when p_f is $\Omega(1/\log n)$, security comes with no price in asymptotic sense, i.e., secure network performance remains on the same order compared to the networks without security requirements (cf. Theorem 7.2 (ii)). We believe that this result is quite important because it provides valuable insight on the desirable operating points that balance security and efficiency concerns. We need to minimize p_f in order to reduce the memory size for keying materials and mitigate the impact of nodes being compromised. Our result implies that even when p_f goes to zero as the network size becomes arbitrarily large, as long as $p_f = \Omega(1/\log n)$, it is still possible to secure a large-scale WANET with negligible overhead. Our results also show that security requirements in general will not increase

[6] The throughput capacity of $\Theta(1/\sqrt{n \cdot \log n})$ was firstly proved by Gupta and Kumar in [60], but their analysis is based on the fluid model. The same result was obtained by Kulkarni and Viswannath [90] through the constant packet size model. El Gamal et al. [48, 50] further improved this result by giving bounds on $D(n)$. Note that recently Franceschetti et al. [46] showed that the $\Theta(1/\sqrt{n})$ throughput capacity is achievable if we relax the assumption that all nodes use the same r_n. Here we still use $\Theta(1/\sqrt{n \cdot \log n})$ bound on throughput because our trust model is a homogeneous one, and for a fair comparison, we also assume the random network model is homogeneous, i.e., all nodes have the same r_n.



(a) Dividing unit torus (b) Corresponding $l \times l$ array

Figure 7-2. The secure communication scheme without SLA.

the e2e delay. This can be intuitively explained as follows: in order to keep the secure connectivity, which is the primary requirement for secure services, the negative effect of a smaller p_f should be compensated by a larger r_n, which will effectively decrease the number of hops a packet need travel and thus the e2e delay.

7.4 Network Performance without SLA

We now present a parameterized secure communication scheme without SLA and analyze its performance. Our theoretical results in Theorem 7.4 confirm that the bounds given in Theorem 7.1 are achievable and tight.

7.4.1 Scheme Description

Scheme 1: the secure communication scheme without SLA

(1) Torus Partition: Divide the unit torus into a set of regular cells, each of side length $c_n = \sqrt{\frac{c_1 \cdot \log n}{n \cdot p_f}}$, where c_1 is a constant. See Figure 7-2 (a) for an illustration.

(2) Setting Transmission Range: Set $r_n = \sqrt{5}\, c_n$, which guarantees that each node can directly communicate with any node in the same cell or in the immediate vertical and horizontal neighboring cells.

(3) Routing: Packets are delivered from the source to the destination in two phases. First, they are forwarded along the cells in the row that contains the source cell until they reach the column that contains the destination cell. In the second phase, packets are forwarded along the cells in the same column to their destination. The L-shaped curve connecting the source and destination as described above is called S-D routes (shaded area in Figure 7-2 (a)).

(4) Cell Scheduling: A cellular time-division multi-access (TDMA) transmission scheme is used, in which each cell becomes active, i.e., its nodes can transmit successfully to nodes in the same cell or in neighboring cells, at regularly scheduled timeslots (cf. Proposition 7.1).

(5) Packet Transmission Scheduling: Each packet will have a timestamp t_b denoting the timeslot the packet was transmitted by the source. When a cell becomes active, it will select the packet with the smallest t_b in the cell to transmit. If there are ties, choose the packet from the S-D pair i which maximizes $(t_b + i) \bmod n$. Note that only one packet is transmitted per timeslot per cell.[7] Our packet transmission scheduling scheme will treat the packets from different sessions equally and prefer the oldest packet in each session.

(6) Secure Inter/ Intra-Cell Transmission: All the packets will be transmitted on primary secure links. When a node need transmit a packet to its neighboring cell, it always transmits the packet to one of its friends in that cell (secure inter-cell transmission); otherwise, it will drop the packet. When a node need transmit a packet to the node in the same cell and they are not friends, the node will find one of their common friends in the same cell as the relay node (secure intra-cell transmission). If it cannot find one, it will drop the packet.

Here we give some primary results on the scheme described above. We first recall the following result on the property of cell scheduling in Step (4), which is widely known now [90].

Proposition 7.1. *Under the protocol model, there exists an interference-free schedule such that each cell becomes active regularly once in K^2 timeslots without interfering with any other simultaneously transmitting cell. Here K depends only on Δ and is independent of n.*

See Figure C-1 for an example of this interference-free schedule.

[7] Each S-D pair is identified by the source node's ID.

Next, we show that the probability that the scheduled packet is dropped in Step (6) in Scheme 1 approaches to zero as $n \to \infty$. This claim is true due to the following lemma.

Lemma 10. *In Scheme 1, we can always find a constant c_1 such that (i) each cell contains $\Theta(\log n/p_f)$ nodes w.h.p.; (ii) given an arbitrary node i, each cell contains $\Theta(\log n)$ friends of i w.h.p.; (iii) given two arbitrary nodes i and j in the same cell, either they are friends or they have at least one common friend in that cell w.h.p.*

Proof. See Appendix-C. □

Lemma 10 shows that, for any source node S (Figure 7-2 (a)), it can always find a friend in each neighboring cells w.h.p. If multiple friends are available in a cell, S randomly chooses one and defines this friend as its secure relay in this cell. With the routing rule in Step (3), this friend-finding procedure (i.e., each secure relay find its own friends in the following neighboring cells) is continued until there is a secure backbone (regular solid lines in Figure 7-2 (a)) spanning all cells. Based on Lemma 10 (ii), every node can construct its own secure backbone[8] as described above. Therefore, each packet can follow this secure backbone until it reaches the secure backbone node k in the same cell as the destination node D through secure inter-cell transmissions. If k is a friend of D, it can directly transmit the packet to D; otherwise, it need find a common friend j to relay the packet, which is guaranteed to happen w.h.p. according to Lemma 10 (iii). Therefore, there are at most two secure intra-cell transmissions (bold solid lines in Figure 7-2 (a)) for each packet w.h.p.

[8] Note that the source node does not need to construct this secure backbone beforehand. It will emerge gradually with the data flow and extend to a new cell when the source node's packet goes through that cell. Here we just show that there exists such a secure backbone for each node to facilitate the analysis in Section 7.4.2.

7.4.2 Performance Analysis of Scheme 1

Note that Scheme 1 only operates on primary secure links. These links will be established after direct neighbor authentication operations, which only need local broadcasts. Obviously the overhead and the delay incurred here is negligible compared to multi-hop data communications (cf. [168, Section II.B]). Therefore, in what follows, we only concentrate on the throughput and delay in the data delivery phase.

Our analysis on Scheme 1 mainly relies on some well-known results about 2-dimensional (2-d) arrays [80, 94], which have been extensively studied in the parallel and distributed computing research community. Therefore, we first review some related definitions and results.

A 2-d $l \times l$ array consists of $L = l^2$ processors or processing units (PUs) arranged in a 2-d $l \times l$ grid. Each PU is connected to its four neighbors via point-to-point wired communication links. In the MIMD (Multiple Instruction Multiple Data) mode, the PUs perform routing in a series of synchronous timeslots. During each timeslot, a PU may send one packet to its neighbors along each of the (up to four) links incident on it. A PU may also receive one packet along each of its incident links during a timeslot. The PUs can be indicated by their coordinates within the array; the PU at position (i, j), $0 \leq i, j \leq l$, is denoted $P_{i,j}$. Here position $(0, 0)$ lies in the upper-left corner (refer to Figure 7-2 (b) for an illustration). A torus is an array with so-called wrap-around links, which connect $P_{i,0}$ with $P_{i,l-1}$ and $P_{0,j}$ with $P_{l-1,j}$. Throughout this chapter, all results about an array can be extended to the corresponding torus, so we do not distinguish tori from arrays hereafter and simply call them arrays[9] for simplicity. An h-h routing problem on 2-d arrays refers to the scenarios that each PU is the source and destination of exactly h packets.

[9] Also notice that, throughout this chapter we only consider 2-d arrays within the MIMD mode.

Lemma 11. *([94] and [80]) h-h routing on $l \times l$ arrays can be performed deterministically in $h \cdot l/2 + O(h^{5/6} \cdot l^{2/3})$ timeslots, with average packet delay $\Theta(l)$.*

We now point out the correspondence between Scheme 1 and the optimal communication scheme for the 2-d array. Let

$$l = \left\lceil \frac{1}{c_n} \right\rceil = \Theta\left(\sqrt{\frac{n \cdot p_f}{\log n}}\right). \tag{7–3}$$

Cell $C_{i,j}$ in Figure 7-2 (a) corresponds to PU $P_{i,j}$ in Figure 7-2 (b) for $0 \leq i, j \leq l$. Without loss of generality, we assume that each source node has only one packet in our WANET model, so there are $\Theta(\log n/p_f)$ packets generated in each cell based on Lemma 10 (i). By letting

$$h = \Theta(\log n/p_f), \tag{7–4}$$

we have formed a correspondence in the traffic pattern between our WANET model and the array, i.e., we associate the h packets generated in a PU with the packets of the nodes contained in the corresponding cell. Routing and scheduling algorithms used by the array to achieve the performance given in Lemma 11 are the same as the schemes we described in Step (3) and (5), respectively. In fact, Scheme 1 simulates the optimal communication scheme for the array by requiring the scheduled node in cell $C_{i,j}$ to perform the communication operation performed by PU $P_{i,j}$ of the array.

Next, we discuss the difference between our WANET model and the array. Note that each PU can transmit and receive up to 4 packets in each timeslot, while in our cell scheduling scheme in Step (4), each cell will be scheduled to be active once in K^2 timeslots (cf. Proposition 7.1). Therefore, compared to the array, the scheme performed in WANET will have a slowdown by no more than a factor of $4K^2$.

Therefore, we make a correspondence between the performance of secure inter-cell communications with that of communications between neighboring PUs. Based on Lemma 11, we can conclude that the total number of timeslots needed to deliver n

packets (one for each source node) to their destination nodes' cells equal to

$$\Theta\left(4K^2\frac{hl}{2}\right) \overset{K=\Theta(1)}{=} \Theta\left(hl\right) \overset{(7--3),(7--4)}{=} \Theta\left(\sqrt{\frac{n\cdot\log n}{p_f}}\right). \tag{7–5}$$

We have already shown in Section 7.4.1 that each packet only needs at most 2 secure intra-cell transmissions, corresponding to $2K^2$ timeslots. Therefore, the total number of timeslots needed to deliver n packets to their destination nodes, denoted as $\varphi(n)$, can still be expressed in Eq. (7–5). Since only one packet has been delivered for each node, we obtain per-node throughput as $\tau = 1/\varphi(n)$ packets/timeslot. Since we assume the constant-packet-size model and that one packet can be transmitted in each timeslot, we have $T(n) = 1/\varphi(n)$ packets/timeslot. From Lemma 11 and Eq. (7–3), we can directly obtain $D(n) = \Theta(\sqrt{n\cdot p_f/\log n})$ timeslots.

Based on the above analysis, we have in fact given a constructive lower bound on $T(n)$ and upper bound on $D(n)$ without SLA as follows.

Theorem 7.4. *When $p_f = \Omega(\log n/n)$, the secure throughput without SLA is $T(n) = \Omega\left(\sqrt{\frac{p_f}{n\cdot\log n}}\right)$ packets/timeslot and the corresponding delay is $D(n) = O\left(\sqrt{\frac{n\cdot p_f}{\log n}}\right)$ timeslots.*

In particular, when $p_f = 1$ or without any security requirement, we can obtain Gupta and Kumar's result [60]. El Gamal et al. [48, 50] reprove their result under the constant-packet-size model with complicated analysis on a discrete-time queuing network. Here, we follow Kulkarni and Viswannath's methodology [90] to avoid these complicated queueing analysis by exploiting the similarity between the cell-based network model and the array. Therefore, our proof is desirable in its simplicity. Moreover, it provides some necessary background for understanding our more complicated scheme designed with SLA.

7.5 Network Performance with SLA

In this section, we analyze achievable secure network performance when SLA is allowed. We first present the following schemes to achieve the performance bounds in Theorem 7.2.

7.5.1 Scheme Description

As a prelude to describing the scheme, we review the SLA operations defined in our previous work [168]. When $p_f \cdot n \cdot \pi r_n^2 \geq c_2$ for some constant c_2, the network consisting of nodes and primary secure links (modeled as primary secure graph) is in the percolated phase, i.e., most nodes are connected by a secure backbone (also called the giant cluster in percolation theory) with primary secure links. There are still $p_i \cdot n$ nodes disconnected from the giant cluster, where $0 < p_i < 1$ is a constant only depending on c_2; we call all these nodes isolated nodes, though their degrees in the physical graph may be larger than 1. Take node i in Figure 7-3 as an example. When $r_n = \Theta(1/\sqrt{np_f})$ and $r_n = \Omega(1/\sqrt{n \log n})$ (or more precisely, when r_n is set as in Step (2) of Scheme 3‘), even if node i is isolated, w.h.p. there exists at least one node in its transmission range, e.g., node j, belonging to the secure backbone. Also note that when $p_f = \Omega(\log n/n)$, w.h.p. node i has at least one friend, e.g., node k, in the secure backbone (cf. Lemma 12 for a justification of these two statements). Therefore, with the help from node j and k, we can perform multi-hop SLA as the following.

Scheme 2: the multi-hop SLA scheme

(1) Isolated node i first sends a secure connection request (SEC-REQ) message to one of its neighboring node j in the secure backbone.

(2) Node j will forward this SEC-REQ message to one of its neighbors in the secure backbone, as long as the latter never receives this message.

(3) When the receiver, say, node k, receives the SEC-REQ message, it will check whether node i is its friend. If it is not the case, node k will forward the SEC-REQ message as described in Step (2); otherwise, it will send back a secure connection approval (SEC-APV) message to the sender. This process will continue until node j receives the SEC-APV message.

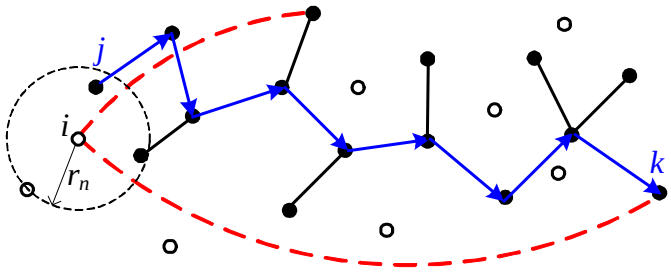


Figure 7-3. Multi-hop SLA operations. Here solid lines and dashed lines represent primary secure links and primary SAs, respectively. Solid and open points represent nodes on the giant cluster (secure backbone) or not, respectively. Node i is isolated and needs to be connected with the secure backbone with the help of its friend, e.g., node k.

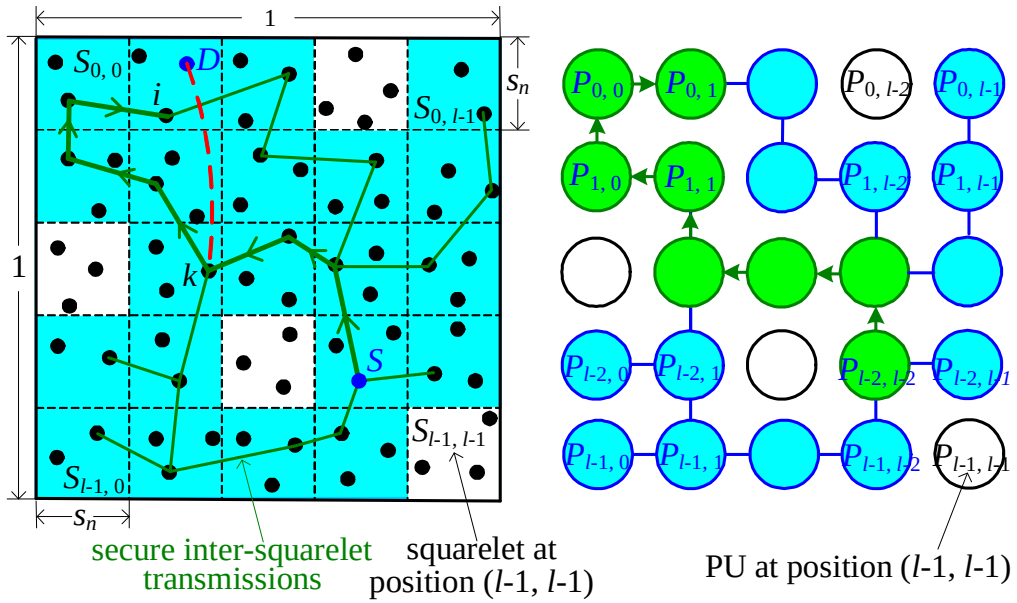
(4) Nodes i and j mutually authenticate each other and secure the physical link $i \leftrightarrow j$.

One important property of Scheme 2 we obtained in [168] is that node k is $O(\log n)$ hops away from node j. Or put it in another way, in order to find a friend of node i in the secure backbone, we need visit $O(\log n)$ nodes w.h.p.

Also note that one prerequisite of Scheme 2 is that every node should know whether it is an isolated node or a node in the secure backbone. This necessitates a secure network partition detection algorithm performed in each node to decide its role in the primary secure graph. Our previous research [168] shows that in the percolated phase, isolated nodes only form clusters with size $O(1)$ even when $n \to \infty$. Therefore, each node can send a probe message, which will be forwarded only through primary secure links. If the probe message can only go through $O(1)$ hops, w.h.p. the node is isolated. The associated overhead for the secure network partition detection is on the same order of direct neighbor authentication, as they both require communications within $O(1)$ hops, which can be ignored as compared to the network-wide multi-hop communications.

Next, we present a primary secure communication scheme for S-D pairs on the secure backbone, and our task is to deliver a packet from the source node to the squarelet in which the destination node dwells.

Scheme 3‘: secure communications on the secure backbone



(a) Dividing unit torus (b) Corresponding $l \times l$ array

Figure 7-4. Secure communication scheme 3‘ with SLA. In (a), solid lines and dashed lines represent primary secure links and primary SAs, respectively. The shaded area represents the squarelets that can be covered by the secure backbone of source node S. In (b), solid points and open points represent PUs on and off the giant cluster, respectively,

(1) Torus Partition: Divide the unit torus into a set of regular squarelets, each of side length

$$s_n = \begin{cases} \sqrt{\frac{c_3 \log n}{n}} & \text{if } p_f = \Omega\left(\frac{1}{\log n}\right), \\ \sqrt{\frac{c_3}{n p_f}} & \text{otherwise,} \end{cases} \tag{7–6}$$

where c_3 is a constant. Note that a cell used in Scheme 1 is much larger than a squarelet in general. In fact a cell contains $\Theta(\log n)$ squarelets when $p_f = o(1/\log n)$ (refer to Figure 7-4 (a) for an illustration).

(2) Setting Transmission Range: Set $r_n = \sqrt{5}\, s_n$, which guarantees that nodes in neighboring squarelets can communicate directly. Also notice that this r_n guarantees that the primary secure graph is in the percolated phase w.h.p. (this is a direct consequence of Theorem 2 in our previous work [168]).

(3) Squarelet and Packet Transmission Scheduling: The squarelet scheduling and the packet transmission scheduling in each active squarelet are the same as the cell scheduling in Step (4) of Scheme 1, and the packet scheduling in each active cell described in Step (5) of Scheme 1, respectively.

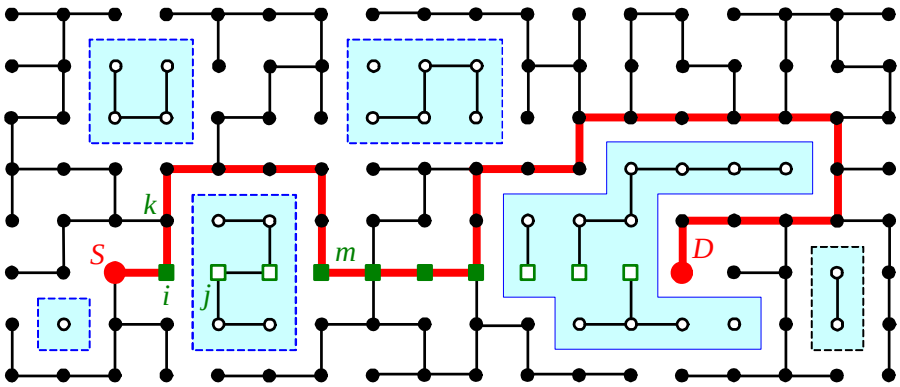


Figure 7-5. Routing scheme on the percolated grid. Here solid lines represent secure links between neighboring PUs/squarelets. Solid and open points represent PUs on and off the giant cluster (secure backbone), respectively. The shaded area represent the finite clusters of PUs disconnected from the secure backbone, and dashed lines represent borders of these clusters.

(4) Secure Inter-Squarelet Transmission: All the packets will be transmitted on primary secure links crossing neighboring squarelets. In other words, when a node needs to transmit a packet to its neighboring squarelet, it always transmits the packet to one of its friends in that squarelet. As what we have done in Scheme 1, we can establish a correspondence between our squarelet system and the $l \times l$ array by setting $l = \lceil 1/s_n \rceil$. See Figure 7-4 for an illustration. Here, two neighboring PUs will have a link in the array, if the packet holder in the corresponding squarelet can find a friend in another squarelet. Therefore, the array we obtained in Figure 7-4 (b) is a faulty array (which will be defined more precisely soon) with link failures, where a failure indicates there is no friend in a neighboring squarelet.

(5) Routing: Since here we only consider the inter-squarelet communications, routing between squarelets is equivalent to that operating on PUs. Therefore, we use the faulty array as an example for a cleaner presentation. Note that the corresponding array is also percolated, and we can guarantee that there exists a path connecting the S-D PUs if the corresponding S-D node pairs are on the secure backbone. See Figure 7-5 for an illustration. We first fix one shortest path of length k connecting the S-D PUs in the array without faulty links, which consists of square nodes in Figure 7-5. Our routing algorithm attempts to follow this shortest path until it encounters a failure link, e.g., at node i. At this point, we simply "circumnavigates" the cluster of isolated nodes (the shaded area) that blocks the path, until either the destination PU is reached, or the algorithm is back onto the original shortest path to it (e.g., reach node m). Since the average size of the cluster of isolated nodes is a constant w.h.p., the path length of route (bold lines in Figure 7-5) found by our scheme is $O(k)$ on average [112].

We next summarize some basic results about torus partition in Scheme 3‘ in the following lemma.

Lemma 12. *In Scheme* 3‘*, we can always find a constant* c_3 *such that (i) each squarelet contains* $\Theta\left(ns_n^2\right)$ *or* $\Omega\left(\log n\right)$ *nodes w.h.p.; (ii) each squarelet w.h.p. contains* $\Theta\left((1-p_i)ns_n^2\right)$ *and* $\Theta\left(p_i ns_n^2\right)$ *nodes on and off the secure backbone, respectively, where* $0 < p_i < 1$ *is a constant; (iii) given an arbitrary nodes* i*, each squarelet contains at least one friend of* i *with probability* p_I*, independently of each other, where* p_I *is a constant; (iv) given an arbitrary nodes* i*, there exists at least one node in node* i*'s transmission range which belongs to the secure backbone and node* i *has at least one friend in the secure backbone w.h.p.*

Proof. See Appendix-C. □

Based on above discussions, we now give the complete description of our scheme supporting secure communications between all S-D pairs as the following:

Scheme 3: the secure communication scheme with SLA

- Phase 1 - primary secure link establishment and secure network partition detection: After this phase, each node finds its neighboring friends and knows whether it is on the secure backbone.
- Phase 2 - connecting isolated nodes to the secure backbone: After this phase, each isolated node will connect to a node on the secure backbone with a derived secure link. We further require that each isolated node only connects to a secure backbone node in the same squarelet. For each squarelet, this phase consists of the following three steps:

  (1) We first select a secure backbone node for each isolated node in the same squarelet. Given a squarelet, denote *NI* and *NS* as the node set of isolated nodes and secure backbone nodes in that squarelet, respectively. For $u \in NI$, $\psi(u) = v$ means that we select node $v \in NS$ for node u. Then we choose v in such a way that for all $v \in NS$ we have $|\{\omega \in NI : \psi(\omega) = v\}| \leq \lceil |NI|/|NS| \rceil$.

  (2) See Figure 7-3 for an example. When node $j \in NS$ is selected for node $i \in NI$, we call node j as the deputy of node i. We will run Scheme 2 for node i. Note that Scheme 2 has two multi-hop communications on the secure backbone. One is from node j to node k, and the other is from node k to node j. These two communications can both be implemented with Scheme 3‘.

(3) Isolated nodes transmit the packets generated by themselves to their deputies, respectively.

- Phase 3 - secure-backbone communication:
  After Phase 2, all the packets generated by sources are redistributed on secure-backbone nodes only, and then we can utilize Scheme 3‘ to deliver these packets from the source nodes or deputies to their corresponding destination squarelets (green lines from S to i in Figure 7-4 (a)). More precisely, if the destination node is also on the secure backbone, we define the squarelet in which it dwells as the destination squarelet. Otherwise, we define one of the closest squarelets to the destination node, which is also covered by the secure backbone, as the destination squarelet. Note that the destination squarelet is always covered by the transmission range of the destination node, which is guaranteed by our torus partition in Scheme 3‘.

- Phase 4 - last-hop delivery: See Figure 7-4 (a) for an example. If node i is a friend of the destination node D, then i can directly transmit the packet to D. Otherwise, we need to secure link $i \rightarrow D$. This can be done by utilizing Scheme 2 again: we find a friend of D, say node k, and then secure the link $i \rightarrow D$ with the help of k. As described in Step (2) of Phase 2, we need use Scheme 3‘ twice to fulfill this operation.

7.5.2 Performance Analysis of Scheme 3

We first analyze the performance of Scheme 3‘. Our analysis mainly relies on the following results on faulty arrays. A q-faulty array refers to the array in which each link may fail independently with some probability bounded above by a fixed value q.

Lemma 13. *There exists a scheme for a q-faulty $l \times l$ array to solve the 1-1 routing problem in $\Theta(l)$ timeslots with probability $1 - 1/l$ when q is small enough. Note that for faulty arrays, we are required to route packets on live links, and we only need route packets among all PUs connected by live paths.*

Remark: Mathies proves Lemma 13 in [112] for $q < 0.5$. It is trivial to extend Lemma 13 to the h-h routing problem: in the same way we can perform the h-h routing on a q-faulty $l \times l$ array within $\Theta(h \cdot l)$ timeslots with the average packet delay of $\Theta(l)$ w.h.p. Compared to Lemma 11, the result here shows that when $q < 0.5$, the running time for a q-faulty array is almost the same as if there were no faults in the array links (up to constant factors), if a routing scheme similar to the one we described in Step (5) in Scheme 3‘ is adopted. It is trivial to find the correspondence between our Scheme 3‘

and the $(1-p_l)$-faulty array. Therefore, the above results can be leveraged to analyze Scheme 3‘ when $p_l > 0.5$, which can be easily achieved by tuning the parameter c_2 mentioned in Section 7.5.1. Following the same argument given in Section 7.4.2, we obtain the following result.

Corollary 7.1. *When each node on the secure backbone has at most $O(1)$ packets, Scheme 3‘ can deliver all these packets within $\Theta(n \cdot s_n)$ timeslots with the average packet delay of $\Theta(1/s_n)$ w.h.p.*

Proof. This can be directly obtained from Lemmas 12 and 13 . □

We now analyze the performance of Scheme 3. Note that Phase 1 and Steps (1) and (3) in Phase 2 only need local broadcasts, which will be dominated by other phases involving Scheme 3‘. We thus ignore them in our asymptotic analysis.

Step (1) in Phase 2 guarantees that every secure-backbone node will act as the deputy for $\Theta(|NI|/|NS|)$ isolated nodes. From Lemma 12 (ii), we know that it is equal to $\Theta(1)$. Therefore, every secure-backbone node only need handle $\Theta(1)$ SEC-REQ or SEC-APV messages. Then the performance of Scheme 3‘ used in Step (2) of Phase 2 can be bounded as in Corollary 7.1. For the same reason, the network performance in Phase 4 is also bounded as in Corollary 7.1. From Steps (1) and (3) of Phase 2, we can guarantee that each secure-backbone node only holds $\Theta(1)$ packets at the beginning of Phase 3, assuming that each source node only generates one packet. Therefore, we can apply Corollary 7.1 again to Phase 3. To sum up, the performance of Scheme 3 is on the same order of that of Scheme 3‘, which is characterized by Corollary 7.1. By substituting Eq. (7–6) into Corollary 7.1 and following the argument given in Section 7.4.2, we can obtain the bounds on the secure throughput and delay.

Based on the above analysis, we have in fact obtained a constructive lower bound on $T(n)$ and upper bound on $D(n)$ with SLA as follows.

Theorem 7.5. *(i) When $p_f = \Omega(\log n/n)$ and also $p_f = O(1/\log n)$, the secure throughput with SLA is $T(n) = \Omega(\sqrt{p_f/n})$ packets/timeslot and the corresponding*

delay is $D(n) = O(\sqrt{n/p_f})$ *timeslots; (ii) When* $p_f = \Omega(1/\log n)$*, the secure throughput with SLA is* $T(n) = \Omega(1/\sqrt{n \cdot \log n})$ *packets/timeslot and the corresponding delay is* $D(n) = O(\sqrt{n/\log n})$ *timeslots.*

7.6 Optimality of Our Schemes

In this section, we present upper bounds on the secure throughput with or without SLA. The corresponding lower bounds on the e2e delay will also be obtained. Since the upper bounds derived here match the constructive lower bounds obtained in Section 7.4 and 7.5, we complete the proof of the Theorems 7.1 and 7.2 of this chapter under the protocol model. The results in this section also show that the schemes we designed in Section 7.4 and 7.5 are optimal at least in the order sense. Note that we defer the proofs of Theorems 7.1 and 7.2 under the physical model to Appendix-D.

7.6.1 Upper Bounds on Secure Throughputs

The secure throughput of random networks defined in Section 7.3.1 is limited by the following three constraints. The maximum feasible throughput satisfying all these constraints is an upper bound on the secure throughput. While there may be other constraints under secure throughput as well, the constraints we consider here are sufficient to provide tight bounds, as the upper bounds obtained here match the constructive lower bounds provided in Section 7.4 and 7.5.

Physical-Connectivity Constraint: We first need to make sure that the network is physically connected, which constrains r_n as $r_n = \Omega(\sqrt{\log n/n})$ [59, 130].

Secure-Connectivity Constraint: The throughput of secure WANETs is constrained by the need to ensure that the network is securely connected, so that every S-D pair can communicate through at least one secure path. Our previous work [168] quantifies this constraint as follows (cf. Section 7.2.2):

- $r_n = \Omega\left(\sqrt{\frac{\log n}{n \cdot p_f}}\right)$ without SLA;
- $r_n = \Omega(1/\sqrt{n \cdot p_f})$ with SLA.

Interference Constraint: The secure throughput is also constrained by interference. Since the wireless channel is a shared medium, under the protocol model, two nodes simultaneously receiving a packet from different transmitters must be separated by enough distance. This implies a constraint on the maximum number of simultaneous transmissions in torus A. We characterize this constraint with the following lemma.

Lemma 14. *The interference constraint requires that* $T(n) \leq \frac{c_3}{n \cdot r_n}$*, where* c_3 *is a constant.*

Proof. We first consider the case when $p_f = 1$. Let $\bar{L}$ be the expected distance between S-D pairs within the unit-area torus, and then $\bar{L} = \Theta(1)$ w.h.p. (cf. [90, Claim 3.1 (3)]). Thus on average each packet need traverse at least $\Theta\left(\frac{\bar{L}}{r_n}\right)$ hops to reach the destination. Since each node generates packets at rate $T(n)$, this means that the packets per timeslot being transmitted by the whole network is at least $nT(n)\frac{\bar{L}}{r_n}$. Under the protocol model, each transmission "consumes" area, i.e., disks of radius $\frac{\Delta}{2}r_n$ around every transmitter should be disjoint [60]. Since the area "consumed" is bounded above by the total area $|A| = 1$, the maximum number of feasible simultaneous transmissions is no more than $\frac{4}{\pi\Delta^2 r_n^2}$. Hence we have the constraint,

$$nT(n)\frac{\bar{L}}{r_n} \leq W\frac{4}{\pi\Delta^2 r_n^2} \Rightarrow T(n) \leq \frac{c_1}{n \cdot r_n}.$$

The throughput of network when $p_f = 1$ is at least as large as the throughput of the network when $p_f < 1$ (this is trivially true, by not using unsecured physical links), so $\frac{c_1}{n \cdot r_n}$ is also an upper bound for $T(n)$ when $p_f \leq 1$. □

By combining the above constraints, we obtain the following theorem on the upper bounds on the secure throughput:

Theorem 7.6. *(i) When* $p_f = \Omega(\log n/n)$*, the secure throughput without SLA is* $T(n) = O\left(\sqrt{\frac{p_f}{n \cdot \log n}}\right)$ *packets/timeslot; (ii) When* $p_f = \Omega(\log n/n)$ *and also* $p_f = O(1/\log n)$*, the secure throughput with SLA is* $T(n) = O(\sqrt{p_f/n})$ *packets/timeslot; (iii) When* $p_f = O(1)$ *and also* $p_f = \Omega(1/\log n)$*, the secure throughput with SLA is* $T(n) = O(1/\sqrt{n \cdot \log n})$ *packets/timeslot.*

7.6.2 Lower Bounds on Secure Delays

Lower bounds on secure delays can be analyzed in a similar fashion. The only thing we need to do is to replace the interference constraint with the following path-length constraint.

Path-Length Constraint: Since only a single packet can be transmitted per cell per timeslot, the e2e delay is lower bounded by the number of hops on the path. Let $\overline{L}$ be the expected distance between S-D pairs. We then have $D(n) \geq \frac{\overline{L}}{r_n}$. If we require that the packet is always transmitted through the secure path, $D(n)$ is even larger, therefore, $D(n) = \Omega(1/r_n)$.

By combining the above constraint with physical and secure connectivity constraints, we obtain the following theorem for the lower bounds on the secure delay:

Theorem 7.7. *(i) When* $p_f = \Omega(\log n/n)$*, the secure delay without SLA is* $D(n) = \Omega\left(\sqrt{\frac{n \cdot p_f}{\log n}}\right)$ *timeslots; (ii) When* $p_f = \Omega(\log n/n)$ *and also* $p_f = O(1/\log n)$*, the secure delay with SLA is* $D(n) = \Omega(\sqrt{n/p_f})$ *timeslots; (iii) When* $p_f = O(1)$ *and also* $p_f = \Omega(1/\log n)$*, the secure delay with SLA is* $D(n) = \Omega(\sqrt{n/\log n})$ *timeslots.*

7.7 Chapter Summary

In this chapter, based on a general random network model, the asymptotic behaviors of secure throughput and delay with the common transmission range r_n and the probability p_f of neighboring nodes having a primary security association, are quantified when the network size n is sufficiently large. The costs and benefits of secure link augmentation operations on the secure network performance are also analyzed.

CHAPTER 8
CONCLUSION AND FUTURE DIRECTIONS

8.1 Dissertation Summary

In this dissertation, we have studied several challenging and fundamental issues related to the network performance and security of wireless ad hoc networks. The main contributions of this dissertation can be summarized as follows.

After a wireless sensor network (WSN) is deployed, sensor nodes are usually left unattended for a long period of time. There is an inevitable devolution of the connected coverage of the WSN due to battery exhaustion of sensor nodes, intended physical destruction attacks on sensor nodes, unpredictable node movement by physical means like wind, and so on. It is, therefore, critical that the base station (BS) learn in real time how well the WSN performs the given sensing task (i.e., what is the current connected coverage) under a dynamically changing network topology. In this dissertation, we propose a coverage inference protocol (CIP) which can provide the BS an accurate and in-time measurement of the current connected coverage in an energy-efficient way. Especially, we show that the scheme called BOND which our CIP requires to be implemented on each sensor node enables each node to locally self-detect whether it is a boundary node with the minimal communication and computational overhead. The BOND can also be exploited to seamlessly integrate multiple functionalities with low overhead. Moreover, we devise extensions to CIP which can tolerate location errors and actively predict the change of the connected coverage based on residual energy of sensor nodes.

Network heterogeneity is a certainty for today's wireless networks. There are two kinds of heterogeneity: first, the distribution of wireless users/devices in the physical space is non-homogeneous; second, wireless devices are likely to have widely varying radio ranges (e.g., cellular/WiMax, WiFi, Zigbee). For a heterogeneous ad hoc network, where short-range wireless links and long-range wireless links (shortcuts) coexist,

how to design efficient decentralized routing protocols with local information is an open problem in the literature. In this dissertation, we show that the long-range links are not necessarily helpful for decentralized routing. Sometimes, short paths with long-range links exist in the network, however, the decentralized routing schemes with only local information cannot find them. We prove that the necessary and sufficient condition for localized routing (e.g., greedy geographic routing) to be efficient is that the probability of a long-range link being present from node u to v should be inversely proportional to the number of nodes which are closer to u than v is. Our result shows that it is the distribution pattern instead of total number of shortcuts that affects the navigability of geometric networks. In most cases, the number of shortcuts is proportional to the cost of the system. Therefore, we need to be very careful when planning the network, due to the possibility that more shortcuts may lead to worse network performance.

This dissertation also investigates the problem of how much benefit network coding can contribute to the network performance in terms of throughput, delay, and storage requirements for mobile ad hoc networks (MANETs), compared to when only replication, storage and forwarding are allowed in relay nodes. We characterize the throughput-delay-storage tradeoffs under different node mobility patterns, i.e., i.i.d. and random walk mobility, with and without network coding. Our results show that when random linear coding instead of replication is used in MANETs, an order improvement on the scaling laws of MANETs can be achieved. Note that previous work showed that network coding could only provide constant improvement on the throughput of static wireless networks. Our work thus differentiates MANETs from static wireless networks by the role network coding plays.

For a multi-hop wireless network (MWN) consisting of mobile nodes controlled by independent self-interested users, incentive mechanism is essential for motivating mobile nodes to cooperate and forward packets for each other. Existing solutions such as barter based, virtual-currency based and reputation based schemes are either less

effective or incur high implementation costs, and therefore do not fit well with the unique requirements of MWNs. In this dissertation, we propose a novel and promising incentive paradigm, Controlled Coded packets as virtual Commodity Currency (**C4**), to induce cooperative behaviors in MWNs. In our **C4**, through introducing several techniques from network coding, coded information packets are utilized as a new kind of virtual currency to facilitate packet/service exchanges among self-interested nodes in a MWN. Since the virtual currency implemented in this way also carries useful data information, it is the counterpart of the so-called commodity currency in the physical world, and the overhead brought by **C4** is extremely small compared to traditional schemes. We theoretically show that **C4** is perfectly efficient to support MWNs with broadcast and multicast traffics. For pure unicast communications, by adjusting the grouping parameter, our **C4** provides a systematic way to smoothly trade incentive effectiveness for implementation cost, and traditional barter based and virtual-currency based schemes are just two extreme cases of **C4**. We also show that when our **C4** is combined with the social network formed by mobile users in the MWN, the implementation costs can be further reduced without sacrificing incentive effectiveness.

In the last few years, trust has been identified as the underlying and the most effective mechanism to secure routing protocols in WANETs and accordingly many trust metrics and trust-based (or secure-aware) routing protocols have been proposed for WANETs. Unfortunately, the issues like algebraic properties and compatibilities of these trust metrics are largely untouched in the literature. The correctness, optimality and efficiency of these trust-based routing protocols have been analyzed by informal means only. It is well-known that informal arguments can be prone to errors, and are scenario-specific. Therefore, there is a strong need for a more rigorous and generally applicable framework and theory to deepen our understanding of the fundamental rules governing all possible trust-based routing protocols, facilitate our formal evaluation and comparisons on existing trust-based routing protocols, and provide guidelines for

designing new trust-based routing protocols. In this dissertation, we develop a formal theory to investigate the correctness, optimality, and inter-operativity of trust-based routing protocols for WANETs. We first propose a formal model to abstract the key algebraic properties of trust-related routing metrics and identify the common elements, like trust evaluation, indirect trust inference and trustworthy path selection, in all trust-based routing protocols. Next, we develop a non-classical path algebra based on bi-monoid to study indirect trust inference problems. We then provide a systematic analysis of the relationship between trust metrics and trust-based routing protocols by identifying the basic algebraic properties that a trust metric must have in order to work correctly and optimally with different generalized distance-vector or link-state routing protocols in WANETs. Moreover, we extend our framework to model the interactions between different trust-based routing protocols, and characterize the conditions under which the correctness and optimality of routing operations can be guaranteed in WANETs where multiple routing protocols coexist or different trust metrics are adopted. The proposed research provides a new methodology for the formal analysis of wireless network security and accelerates the evaluation, design and real deployment of trust-based routing protocols.

Security always comes with a price in terms of performance degradation, which should be carefully quantified. This is especially the case for wireless ad hoc networks (WANETs) which offer communications over a shared wireless channel without any pre-existing infrastructure. Forming end-to-end secure paths in such WANETs is more challenging than in conventional networks due to the lack of central authorities, and its impact on network performance is largely untouched in the literature. In this dissertation, based on a general random network model, the asymptotic behaviors of secure throughput and delay with the common transmission range r_n and the probability p_f of neighboring nodes having a primary security association are quantified when the network size n is sufficiently large. The costs and benefits of secure-link-augmentation

operations on the secure throughput and delay are also analyzed. In general, security has a cost: since we require all the communications operate on secure links, there is a degradation in the network performance when $p_f < 1$. However, one important exception is that when p_f is $\Omega(1/\log n)$, the secure throughput remains at the Gupta and Kumar bound of $\Theta\left(1/\sqrt{n \log n}\right)$ packets/timeslot, wherein no security requirements are enforced on WANETs. This implies that even when the p_f goes to zero as the network size becomes arbitrarily large, it is still possible to build throughput-order-optimal secure WANETs, which is of practical interest since in many practical large-scale WANETs, p_f is very small.

8.2 Future Directions

The engineering and study of large-scale complex networks will be a major focus of scientific research in the 21st century, particularly in the areas of communications, sociology, biology, and cognitive science. It also poses a great variety of important challenges and open problems. In the future, I will continue to focus my research on developing theories and methodologies to better understand the behavior of large complex networks, and engineer them in a more efficient and secure way. My current research on wireless networks and network security offers an initial validation of these methodologies. In the short term, I plan to also investigate other complex networks, such as online social networks; and in the long term, I would like to extend my methodologies to a broader scope, the so called Network Science[1] , a new and emerging scientific discipline that examines the interconnections among diverse physical or engineered networks, information networks, biological networks, cognitive and semantic networks, and social networks. Research in this field seeks to discover common principles, algorithms and tools that govern network behavior.

[1] See: http://en.wikipedia.org/wiki/Network_science

Social Networking and Its Applications in Networking Systems

Recently, many online social networking (OSN) applications such as Facebook, MySpace and Twitter emerge as new ways to connect people. These applications already attract a tremendous number of users, and their communities are expected to grow much larger in the near future. Many interesting questions arise under the new contexts of these innovative applications. For example, what are the fundamental characteristics of the social graphs representing user social connections? How does a computer virus spread on such a graph? As users may lose interests in a small community, what is the critical size of a user community to ensure the sustainability and growth of this community? I would like to investigate these problems from a theoretical perspective, considering in particular the uncertainty among social connections of users. The ultimate goal is to derive useful insights towards designing effective algorithms to improve the utility of social networking applications. More specifically, I am particularly interested in designing effective learning algorithms based on user social connections to recommend new contacts with common interests, display advertisements, and sell products to interested users, which are crucially important to commercial success.

Furthermore, social networking information, as a complementary channel to traditional networking characteristics, is potentially useful to improve the performance of other networking systems. For example, most current delay tolerant networks (DTNs) only utilize mobile connectivity information to select routes and deliver data. On the other hand, social networking provides indications of the inter-meeting time interval among users as they meet friends more frequently than strangers. Hence, such information is helpful to facilitate data routing in DTNs. In addition, social connections naturally infer trusts among friends and may be useful to reduce security overhead. They can also help develop collaboration efforts among different nodes to improve and optimize network utilities. I am interested in exploring innovative ways and efficient

algorithms to apply social networking information to improve existing networking systems.

Network Science and Network Algorithm Design

My future research in Network Science will be focused on the fundamental aspects of problems lying at the heart of a large number of complex networks, from WANETs to biological, or social networks. My study will also be pertinent to the design and analysis of algorithms for these complex networks. The originality of my approach is based on the following methodology and concerns:

First of all, instead of well-known properties of complex networks, such as small world or scale free, my study will be built upon the new discoveries of the network properties, such as bounded growth rate, low doubling dimension, minor excluding or hyperbolic metrics. These network properties are very novel. For instance, the notion of doubling dimension has been only very recently introduced with the objective of tacking hard problems such as TSP. Notions like minor excluding, or hyperbolic metrics have been seldomly considered for the design and analysis of network algorithms, whereas it is strongly believed that they are critical in this context. The main advantages of the aforementioned notions are their generality and powerfulness. They apply to a large class of networks, and enable efficient solutions for hard problems. (For example, any network with bounded doubling dimension is navigable, i.e., greedy routing performs in poly-logarithmic number of steps).

Secondly, I will tackle some network problems in contexts where the network is only implicitly and/or partially known. The concept of implicit knowledge aims at capturing the fact that the amount of knowledge each node of a distributed network has about any aspect of the topology (from the full knowledge of the whole structure or global information such as number of nodes, to local information such as neighborhood topology) is inherently limited. Implicit knowledge is modeled by an oracle, the complexity of which is measured in terms of the amount of information (i.e., the number

of bits) given to the entities or nodes composing the networks. The concept of partial knowledge aims at capturing the fact that nodes need not know all their connections. This is modeled by the probe-complexity of the problems, i.e., the minimum amount of probing that nodes must perform to solve a problem.

APPENDIX A
NETWORK TOPOLOGIES USED IN PERFORMANCE EVALUATIONS

Consider the situation where sensor nodes are independently and randomly placed in the ROI. Such a random initial deployment is required when individual sensor placement is infeasible and is desirable when priori knowledge of the ROI and the monitored target is limited or not available. In this cases, it is widely accepted in the literature [59] that the locations of sensors can be modelled by a 2-D homogeneous Spatial Poisson Point Process (SPPP) with density λ.

Definition A.1. ***[homogeneous SPPP]*** A homogeneous SPPP with density λ can be defined by the following two properties: First, for any measurable subset of A_l with area B,

$$\Pr\{\textit{finding } i \textit{ nodes in the region of area } B\} = \frac{(\lambda B)^i e^{-\lambda B}}{i!}.$$

Second, the number of nodes in disjoint (non-overlapping) area are independent random variables.

Each node is expected to have $k = \pi r_c^2 \lambda$ neighbors on average, and the expected number of nodes in A_l is given by $n = \lambda \cdot A_l$. When each node fails independently and uniformly with probability p. It has been shown that functional nodes still form a homogeneous SPPP with density $\lambda' = (1-p)\,\lambda$ [150]. In this case, the network can be uniquely identified by the current node density λ (or equivalently k).

Note that by tuning parameter k (or λ), we can get different network topologies corresponding to different coverage patterns. Four situations get great interests in the literature (see Figure A-1 for an illumination) [130]:

(a) When $k > 4\log n + 4\log\log n$, the ROI is almost fully covered. There are no boundary nodes when border information is available (cf. Figure A-1 (a)).

(b) When $k > \log n$, the whole network is connected. The number of interior nodes is larger than that of boundary nodes (cf. Figure A-1 (b)).

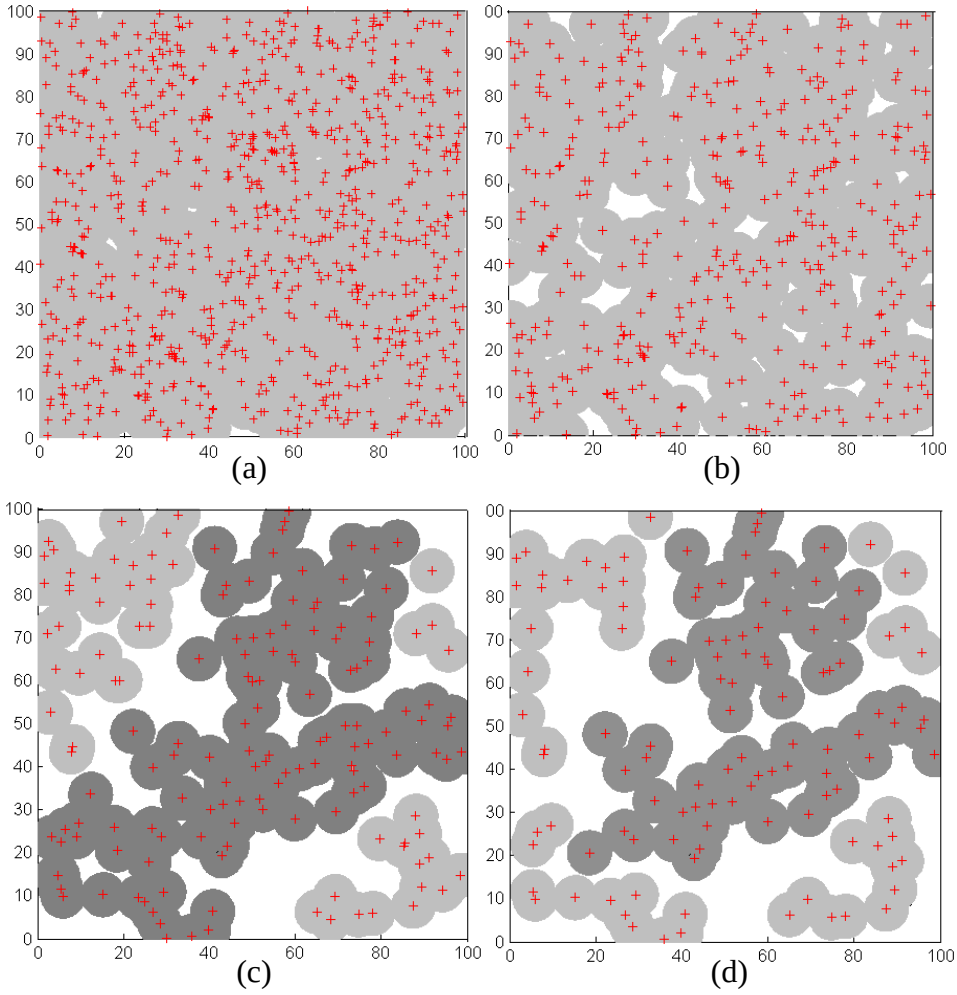


Figure A-1. Some network topologies used in our performance evaluation. (a) $k = 40$. (b) $k = 15$. (c) $k = 5$. (d) $k = 4$. In all situations, the position of the BS is $(50, 50)$. Shaded area represents coverage of sensors. Notice that in (c) and (d) only darkly shaded area represents connected coverage needed to be measured.

(c) When $k > 4.5$, the network is percolated. There only one big cluster, and the number of interior nodes is smaller than that of boundary nodes (cf. Figure A-1 (c)).

(d) When $k < 4.5$, the network is subcritical (collapsed), and consists of many small clusters (cf. Figure A-1 (d)).

Therefore, in our evaluation we focus on the cases when

$$4.5 < k \leq 4 \log (n) + 4 \log \log (n) . \tag{A–1}$$

It can be found that, although we only use homogeneous SPPPs to generate the network topologies, when the node density is smaller than the critical value for the connectivity, the network topologies will become very irregular due to the disconnectedness.

In our simulations, we use NS-2 and assume $r_s = 5$ units, the data size for position representation is 32 bits, the energy consumed to transmit and receive one bit is 0.8μJ/bit and 0.6μJ/bit, respectively. Sensor nodes are distributed in a square ROI with side $l = 100$ units. In the network initial deployment phase, in order to ensure coverage, total number of nodes deployed in the ROI is 1000 (corresponding to $k = 40$). The MAC protocol used in our simulation is 802.11.

APPENDIX B
ASYMPTOTIC NOTATION

We use the following standard notation throughout the dissertation. For two nonnegative functions $f(\cdot)$ and $g(\cdot)$:

(i) $f(n) = O(g(n))$ means that there exists a constant c and an integer N such that $f(n) \leq c \cdot g(n)$ for $n > N$ (i.e., asymptotic upper bound);

(ii) $f(n) = o(g(n))$ means that $\lim_{n\to\infty} f(n)/g(n) = 0$ (i.e., asymptotic insignificance);

(iii) $f(n) = \Omega(g(n))$ means that there exists a constant c and an integer N such that $f(n) \geq c \cdot g(n)$ for $n > N$ (i.e., asymptotic lower bound);

(iv) $f(n) = \omega(g(n))$ means that $\lim_{n\to\infty} f(n)/g(n) = \infty$ (i.e., asymptotic dominance);

(v) $f(n) = \Theta(g(n))$ means that $f(n) = O(g(n))$ and $g(n) = O(f(n))$ (i.e., asymptotic tight bound).

APPENDIX C
SOME RESULTS ABOUT TORUS PARTITIONS IN SCHEME 1 AND 3'

As a prelude, we first establish the following Chernoff bound [61] for a Poisson random variable X of parameter λ.

Lemma 15. *Let X be a Poisson random variable of parameter λ, we have*

$$\boldsymbol{Pr}[X \geq a] \leq \frac{e^{-\lambda}(e\lambda)^a}{a^a}, \text{ for } a > \lambda \tag{C–1}$$

$$\text{and } \boldsymbol{Pr}[X \leq a] \leq \frac{e^{-\lambda}(e\lambda)^a}{a^a}, \text{ for } a < \lambda. \tag{C–2}$$

For $0 < \delta < 1$, Chernoff bounds given in Eq. (C–1) and Eq. (C–2) can be combined and simplified to

$$\boldsymbol{Pr}\left[\,|X - \lambda| \geq \delta\lambda\,\right] < 2\, e^{-\delta^2\lambda/2}. \tag{C–3}$$

Proof. Note that for any random variable $X \geq 0$, and constants $a, t \geq 0$, we have $X \geq a$ if and only if $e^{tX} \geq e^{ta}$. So by Markov's inequality, we have

$$\mathbf{Pr}[X \geq a] \leq \frac{\mathbf{E}\left[e^{tX}\right]}{e^{ta}}.$$

For a Poisson random variable X, we have

$$\begin{aligned}
\mathbf{E}\left[e^{tX}\right] &= \sum_{k\in\mathbb{N}} \frac{e^{tk}e^{-\lambda}\lambda^k}{k!} \\
&= e^{-\lambda}\sum_{k\in\mathbb{N}} \frac{\left(\lambda e^t\right)^k}{k!} \\
&= e^{-\lambda}e^{\lambda e^t} = e^{\lambda(e^t-1)}.
\end{aligned}$$

Therefore, we have $\mathbf{Pr}[X \geq a] \leq e^{\lambda(e^t-1)}e^{-ta} = e^{\lambda(e^t-1)-ta}$. For $a > \lambda$, we choose $t = \log(a/\lambda) > 0$ and obtain Eq. (C–1). Following a similar approach, we can obtain Eq. (C–2) for $a < \lambda$ by choosing $t = \log(a/\lambda) < 0$.

By substituting $a = (1+\delta)\lambda$ into Eq. (C–1), we obtain

$$\mathbf{Pr}[X \geq (1+\delta)\lambda] \leq \left(\frac{e^\delta}{(1+\delta)^{(1+\delta)}}\right)^\lambda < e^{-\delta^2\lambda/4}. \tag{C–4}$$

By substituting $a = (1-\delta)\lambda$ into Eq. (C–2), we obtain

$$\mathbf{Pr}[X \leq (1-\delta)\lambda] \leq \left(\frac{e^{-\delta}}{(1-\delta)^{(1-\delta)}}\right)^{\lambda} < e^{-\delta^2\lambda/2}. \tag{C–5}$$

Therefore, we can obtain Eq. (C–3) by combining Eqs. (C–4) and (C–5). □

Then we prove Lemma 10 for Scheme 1 in Section 7.4.1 and Lemma 12 for Scheme 3‘ in Section 7.5.1. We first show that these two lemmas hold when node positions follow the Poisson point process, i.e., $\mathcal{P}_n$.

Proof of Lemma 10 with $\mathcal{P}_n$. (i) Based on the description of Scheme 1 in Section 7.4.1, we know that there are $m = \left\lceil \frac{1}{c_n^2} \right\rceil = \frac{np_f}{c_1 \log n}$ cells, and the number of nodes in each cell is a Poisson random variable X with parameter $\lambda = nc_n^2 = c_1 \log n / p_f$, where c_1 is a constant and $p_f = \Omega(\log n/n)$. For $0 < \delta < 1$, let A_n be the event that there is at least one cell with more than $(1+\delta)\lambda$ or less than $(1-\delta)\lambda$ nodes. By the union bound and Eq. (C–3) in Lemma 15, we have

$$\begin{aligned} \mathbf{Pr}[A_n] &\leq m\mathbf{Pr}[\,|X-\lambda| \geq \delta\lambda\,] \\ &< \frac{2\,np_f}{c_1 \log n}\left(\frac{1}{n}\right)^{\frac{c_1\delta^2}{2p_f}} \to 0 \end{aligned}$$

as n tends to infinity for any $c_1 \geq 4/\delta^2$. Therefore, each cell contains $\Theta(\lambda) = \Theta(\log n/p_f)$ nodes w.h.p.

(ii) Given an arbitrary node i in a particular cell, the number of i's friends in that cell is a Poisson random variable Y of parameter $\lambda' = p_f\lambda = c_1 \log n$. For $0 < \delta < 1$, by the Chernoff bound in Eq. (C–3), we have

$$\mathbf{Pr}[\,|Y-\lambda'| \geq \delta\lambda'\,] < 2\left(\frac{1}{n}\right)^{c_1\delta^2/2} < 2\left(\frac{1}{n}\right)^{2}$$

for any $c_1 \geq 4/\delta^2$. Applying union bound over all $m \leq n$ cells in the network, the probability that this happens in any cell is at most $2/n$, which tends to zero as n tends to infinity. Therefore, each cell contains $\Theta(\lambda') = \Theta(\log n)$ friends of node i w.h.p.

(iii) Considering an arbitrary cell. All nodes in this cell and primary secure links between these nodes form a subgraph, which can be modeled as an Erdös-Rényi random graph [16, 74]. From the above proof, we know that the number of nodes in each cell is $\Theta(\log n/p_f)$ and that the average node degree in this subgraph is $\Theta(\log n)$, which is larger than the logarithm of the number of nodes in the cell, given that $p_f = \Omega(\log n/n)$. Therefore, by the properties of the Erdös-Rényi random graph [16, 74], this subgraph is connected, i.e., there exists a secure path connecting arbitrary node pairs in the cell. Because all nodes in the cell are in the transmission range of each other, to find this secure path only needs one-hop local communications, which can be ignored compared to the multi-hop data communications. □

Proof of Lemma 12 with $\mathcal{P}_n$. (i) Based on the description of Scheme 3‘ in Section 7.5.1, we know that when $p_f = o(1/\log n)$, there are $m = 1/s_n^2 = \frac{np_f}{c_3}$ squarelets, and the number of nodes in each squarelet is a Poisson random variable X of parameter $\lambda = ns_n^2 = c_3/p_f$, where c_3 is a constant and $p_f = \Omega(\log n/n)$ and $p_f = o(1/\log n)$. For $0 < \delta < 1$, let A_n be the event that there is at least one squarelet with more than $(1+\delta)\lambda$ or less than $(1-\delta)\lambda$ nodes. By the union bound and Eq. (C–3) in Lemma 15, we have

$$\begin{aligned} \mathbf{Pr}[A_n] &\leq m\mathbf{Pr}[\,|X-\lambda| \geq \delta\lambda\,] \\ &< \frac{2\,np_f}{c_3}\left(\frac{1}{n}\right)^{\frac{c_3\delta^2}{2}} \to 0 \end{aligned}$$

as n tends to infinity for any $c_3 \geq 4/\delta^2$. Therefore, each squarelet contains $\Theta(\lambda) = \Theta(ns_n^2)$ nodes w.h.p.

When $p_f = \Omega(1/\log n)$, the proof of Lemma 12 (i) is straightforward and is omitted here due to space constraints.

(ii) Suppose there are Z nodes in the network, where Z is a Poisson random variable of parameter n. From our previous work [168], we know that $(1-p_i)\cdot|Z|$ nodes (called backbone nodes) are connected by a secure backbone (also called the *giant cluster* in the percolation literature) with primary secure links. There are still $p_i \cdot |Z|$

nodes (called isolated nodes) disconnected from the giant cluster, where $0 < p_i < 1$ is a constant only depending on parameter c_3 in Scheme 3‘. From the randomness of the construction of the network model, we know that all backbone nodes or isolated nodes are also uniformly distributed in the unit torus. Therefore, following the similar argument in the proof of Lemma 12 (i), we can prove that each squarelet w.h.p. contains $\Theta((1-p_i)ns_n^2)$ backbone nodes and $\Theta(p_i ns_n^2)$ isolated nodes.

(iii) Because the number of nodes in each squarelet is a Poisson random variable independent of that in any other squarelet, and p_f's between different node pairs are also independent. The event that a squarelet contains at least one friend of a given node i is independent of that in any other squarelet. Next we show that this event happens with probability p_l, which is lower bounded by a constant. Recall that in Scheme 3‘, when $p_f = \Omega(1/\log n)$, the number of nodes in each squarelet, i.e. $|X|$, is lower bounded by $(1-\delta)c_3 \log n$. We thus have

$$\begin{aligned} p_l &= 1-(1-p_f)^{|X|} \\ &> 1-\left(1-\frac{c_4}{\log n}\right)^{(1-\delta)c_3\log n} > 1-e^{-(1-\delta)c_3c_4}, \end{aligned}$$

where δ, c_3 and c_4 are all constants. When $p_f = o(1/\log n)$, the number of nodes in each squarelet, i.e., $|X|$, is lower bounded by $(1-\delta)c_3/p_f$. We thus have

$$p_l = 1-(1-p_f)^{|X|} > 1-(1-p_f)^{(1-\delta)c_3/p_f} > 1-e^{-(1-\delta)c_3},$$

where δ and c_3 are all constants.

(iv) Firstly, from Eq. (7–6) in Scheme 3‘ we directly arrive at the conclusion that each squarelet contains at least one node on the secure backbone w.h.p. Since $r_n = \sqrt{5}s_n$, we know that in node i's transmission range, there exists at least one squarelet. Therefore, there exists at least one node in node i's transmission range which belongs to the secure backbone w.h.p. Secondly, recall that in Scheme 3‘ we can guarantee that the primary secure graph is in the percolated phase w.h.p.. We also have proved

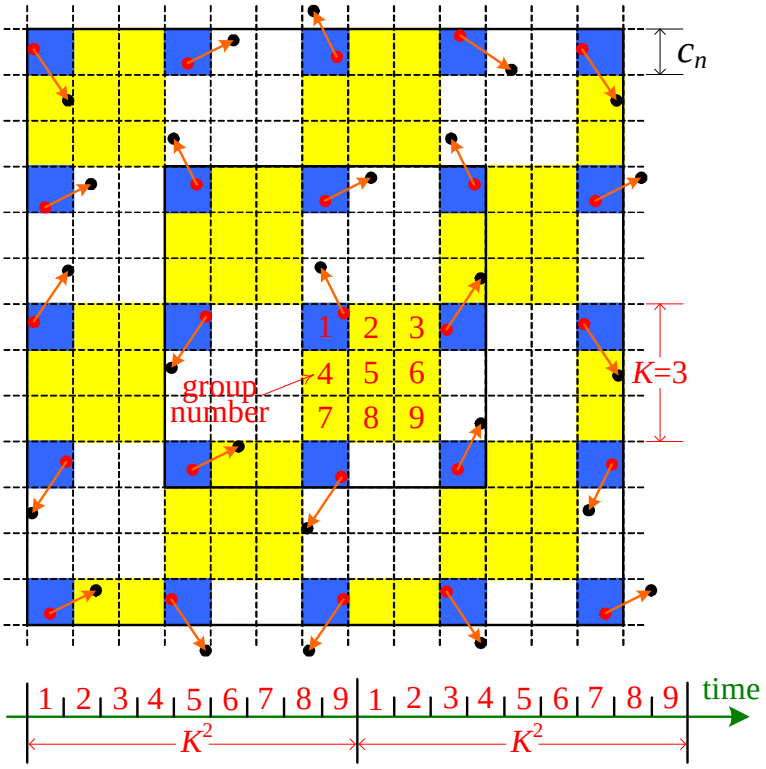


Figure C-1. Cell scheduling scheme. Here is an illustration of the cells being divided into K^2 groups for the case of $K = 3$, i.e., 9 groups. All the blue cells which are in group 1 transmit in the same timeslot. In the next timeslot all the cells in group 2 transmit and so on.

in our previous work (cf. Theorem 2 in [168]) that when the primary secure graph is in the percolated phase, each node belongs to the secure backbone with a probability S, where S is a constant. When $p_f = \Omega(\log n/n)$, there are at least $\Theta(\log n)$ friends of node i in the whole network, and each friend belongs to the secure backbone with the probability S. From the Chernoff bound, it is easy to show that at least one of these $\Theta(\log n)$ friends belongs to the secure backbone. □

Note that $\mathcal{X}_n$ can be well approximated by $\mathcal{P}_n$ as n tends to infinity. Therefore, by the de-Poissonization technique introduced in [130, Section 2.5, p.37], we can prove that Lemma 10 and Lemma 12 also hold when nodes follow a uniform point process, i.e., $\mathcal{X}_n$, for n tending to infinity. Due to space constraints, we omit this routine proof here.

APPENDIX D
SECURE NETWORK PERFORMANCE UNDER THE PHYSICAL MODEL

Here we show that the same results on secure WANETs as in Theorems 7.1 and 7.2 can be obtained under the physical model.

We first show that the constructive lower bounds provided in Section 7.4 and 7.5 will not be changed under the physical model. Note that the protocol model only relates to the cell scheduling part of schemes proposed in Section 7.4 and 7.5. Therefore, if we can show that the same property of the cell scheduling as described in Proposition 7.1 still holds for the physical model, we are done. In what follows, we prove this claim based on the assumption that $\alpha > 2$.

Proof of Proposition 7.1 under the physical model. We use the same cell scheduling scheme as in Proposition 7.1 under the protocol model (see Figure C-1 for an illustration). The received power of the desired signal is lower bounded by

$$P_n \cdot G_{ij} = P_n \cdot d_{ij}^{-\alpha} \geq P_n \cdot (\sqrt{5} c_n)^{-\alpha},$$

where c_n is the side length of each cell.

We then bound the interference, i.e., I. Consider a particular cell C. If one node from this cell is transmitting, all other simultaneous transmissions may occur in cells belonging to the same set of cells that are as a vertical and horizontal distance of exactly some multiples of a particular integer K. Actually, the interfering cells are placed along the perimeter of concentric squares, whose center is C, and each square contains $2Ki$ ($i = 1, 2, \cdots, L$) interfering cells as depicted in Figure C-1, where L is the number of such concentric squares. For example, the first concentric square contains 8 interfering cells, whereas the second concentric square contains 16 interfering cells, for the particular case where $K = 4$. Each node in the intended cell C transmits data packets to nodes in the four neighboring cells. Then, the distance between these nodes (the possible receivers in the four adjacent cells) and the interfering ones is at

least $(K-2)c_n i$ ($i = 1, 2, \cdots, L$). As we are considering a lower bound, we take the worst case. Then, the number of concentric squares (irrespective of the position of the intended cell, because the worst case is when the intended cell is at one corner of the area) is at most $L \leq \left\lceil \frac{1}{2Kc_n} \right\rceil$. We proceed to upper-bound the interference at the receiver j as

$$
\begin{aligned}
I &= \sum_{k \in \mathcal{S} \setminus \{j\}} P_n d_{kj}^{-\alpha} \qquad \text{(Recall that } \mathcal{S} \text{ is the concurrent transmitter set)} \\
&\leq \sum_{i=1}^{L} \frac{P_n \cdot 2Ki}{[(K-2)c_n i]^{\alpha}} \\
&= \frac{2P_n K}{[(K-2)c_n]^{\alpha}} \sum_{i=1}^{L} i^{1-\alpha} \\
&\leq \frac{2P_n K}{[(K-2)c_n]^{\alpha}} \left[1 + \int_1^L x^{1-\alpha} dx \right] \\
&= \frac{2P_n K}{[(K-2)c_n]^{\alpha}} \left[1 + \frac{1}{2-\alpha} (L^{2-\alpha} - 1) \right] \\
&= \frac{2P_n K}{[(K-2)c_n]^{\alpha}} \left(\frac{\alpha - 1}{\alpha - 2} \right) + \frac{2P_n K}{[(K-2)c_n]^{\alpha}} \left(\frac{L^{2-\alpha}}{2-\alpha} \right) \\
&\leq c_5 \frac{P_n K}{[(K-2)c_n]^{\alpha}} \quad \text{(Recall that } \alpha > 2\text{)},
\end{aligned}
$$

where c_5 is a positive constant. Therefore, based on the physical model (cf. Section 7.3.1.2), we have

$$
SINR_{ij} \geq \frac{P_n \cdot (\sqrt{5} c_n)^{-\alpha}}{N_0 + c_5 \frac{P_n K}{[(K-2)c_n]^{\alpha}}} = \frac{c_6 P_n}{c_7 c_n^{\alpha} N_0 + c_8 P_n}, \tag{D–1}
$$

where c_6, c_7 and c_8 are constants. Recall that $c_n \leq 1$. Therefore, $SINR_{ij}$ in Eq. (D–1) can be lower-bounded by some constant β, which guarantees the successful reception of packets at node j. So we complete the proof that Proposition 7.1 also holds under the physical model. □

Next, we show that the upper bound on the secure throughput and the lower bound on the e2e delay provided in Section 7.6 will not be changed under the physical model.

Note that the interference model only affects the interference constraint. Therefore, if we can show that the physical model yields the same interference constraint, we are done. The following lemma on the existence of a correspondence between physical and protocol models on simultaneous transmission sets guarantees that it is indeed the case.

Lemma 16. *Let* $\Delta(\beta) = \left(48\frac{2^{\alpha-2}}{\alpha-2}\beta\right)^{1/\alpha}$. *Suppose that for* $\Delta > \Delta(\beta)$ *the protocol model allows simultaneous transmissions for a transmitter-receiver (T-R) pair in a set* $\mathcal{S}$. *Then there exists a power assignment* $\{P_i, 1 \leq i \leq n\}$ *allowing the same T-S pair set* $\mathcal{S}$ *under the physical model with threshold* β.

Proof. cf. the proof of Theorem 4.1 in [163, p.174]. □

REFERENCES

[1] T.B. Achacoso and W.S. Yamamoto. *AY's Neuroanatomy of C.elegans of Computation*. CRC Press, Boca Raton, FL, 1992.

[2] R. Ahlswede, N. Cai, S. R. Li, and R. W. Yeung. Network information flow. *IEEE Transactions on Information Theory*, 46(4):1204–1216, July 2000.

[3] E. Ahmed, A. Eryilmaz, M. Medard, and A. Ozdaglar. On the scaling law of network coding gains in wireless networks. In *Proc. of MILCOM 2007*, Orlando, FL, Oct. 2007.

[4] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: A survey. *Computer Networks (Elsevier) Journal*, 38(4):169–181, 2002.

[5] I.F. Akyildiz, Xudong Wang, and Weilin Wang. Wireless mesh networks: a survey. *Computer Networks*, 47(4):445–487, March 2005.

[6] R. Albert, H. Jeong, and A.-L. Barabasi. The diameter of the world wide web. *Nature*, 401:130–131, Sep. 1999.

[7] D. Aldous and J. Fill. Reversible markov chains and random walks on graphs. Monograph in preparation.

[8] J. Aspnes, Z. Diamadi, and G. Shah. Fault-tolerant routing in peer-to-peer systems. In *Proc. of PODC 2002*, Monterey, CA, July 2002.

[9] F. Baccelli, K. Klein, M. Lebourges S., and Zuyev. Stochastic geometry and architecture of communication networks. *Telecommunication Systems*, 7(1-3):209–227, June 1997.

[10] X. Bai, S. Kumar, D. Xuan, Z. Yun, and T. Lai. Deploying wireless sensors to achieve both coverage and connectivity. In *MobiHoc'06*, Florence, Italy, May 2006.

[11] N. Bansal and Z. Liu. Capacity, mobility and delay in wireless ad hoc networks. In *Proc. of IEEE INFOCOM 2003*, San Francisco CA, March 2003.

[12] L. Barrière, P. Fraigniaud, E. Kranakis, and D. Krizanc. Efficient routing in networks with long range contacts. In *Proc. of the 15th International Symposium on Distributed Computing (DISC 01)*, Lisboa, Portugal, Oct. 2001.

[13] V. Bhandari and N.H. Vaidya. Capacity of multi-channel wireless networks with random (c, f) assignment. In *Proc. of MobiHoc'07*, Montreal, QC, Canada, Sept. 2007.

[14] V. Bhandari and N.H. Vaidya. Secure capacity of multi-hop wireless networks with random key pre-distribution. In *Proc. of 2nd IEEE Workshop on Mission-Critical Networking*, Phoenix, AZ, April 2008.

[15] B. Bloom. Spaxe/time trade-offs in hash coding with allowable errors. *Communications of the ACM*, 13(7):422C426, June 1970.

[16] B. Bollobás. *Random Graphs*. Academic Press, Orlando, FL, 1985.

[17] A. Broder and M. Mitzenmacher. Network applications of bloom filters: A survey. *Internet Mathematics*, 1(4):485–509, 2005.

[18] S. Buchegger and J.-Y. L. Boudec. Performance analysis of the CONFIDANT protocol. In *Proc. of MobiHoc 2002*, Lausanne, Switzerland, June 2002.

[19] L. Buttyan, L. Dora, M. Felegyhazi, and I. Vajda. Barter trade improves message delivery in opportunistic networks. *Ad Hoc Networks*, 8(1):1–14, January 2010.

[20] L. Buttyan and J.-P. Hubaux. *Security and Cooperation in Wireless Networks*. Cambridge University Press, Cambridge, UK, 2007.

[21] M. Cagalj, S. Capkun, and J. Hubaux. Wormhole-based anti-jamming techniques in sensor networks. Accepted for publication by IEEE Transactions on Mobile Computing (TMC), 2006.

[22] S. Capkun, L. Buttyan, and J. Hubaux. Small worlds in security systems: an analysis of the pgp certificate graph. In *New Security Paradigms Workshop 2002*, Norfolk, VA, Sep. 2002.

[23] S. Capkun, L. Buttyan, and J. Hubaux. Self-organized public-key management for mobile ad hoc networks. *IEEE Transactions on Mobile Computing*, 2(1):52–64, Jan. 2003.

[24] S. Capkun, J. Hubaux, and L. Buttyan. Mobility helps security in ad hoc networks. In *MobiHoc 2003*, Annapolis, MD, June 2003.

[25] B. Carré. *Graphs and Networks*. Oxford University Press, 1979.

[26] H. Chan, A. Perrig, and D. Song. Random key predistribution schemes for sensor networks. In *Proc. of IEEE Symposium on Security and Privacy (S&P)*, Berkeley, CA, May 2003.

[27] C.-K. Chau, R. Gibbens, and T. G. Griffin. Towards a unified theory of policy-based routing. In *Proc. of INFOCOM 2006*, Barcelona, Spain, April 2006.

[28] Tingting Chen and Sheng Zhong. INPAC: An enforceable incentive scheme for wireless networks using network coding. In *Proc. of INFOCOM 2010*, San Diego, CA, March 2010.

[29] R. Chitradurga and A. Helmy. Analysis of wired shortcuts in wireless sensor networks. In *Proc. of IEEE/ACS International Conference on Pervasive Services*, Beirut, Lebanon, July 2004.

[30] D. Coppersmith, D. Gamarnik, and M. Sviridenko. The diameter of a long-range percolation graph. *Random Structures and Algorithms*, 21(1):1–13, August 2002.

[31] J. Crowcroft, R. Gibbens, F. Kelly, and S. Ostring. Modelling incentives for collaboration in mobile ad hoc networks. *Performance Evaluation*, 57(4):427–439, August 2004.

[32] R. Davies. *A History of Money from Ancient Times to the Present Day (3rd edn.)*. University of Wales Press, Wales, UK, 2002.

[33] S. Deb, M. Medard, and C. Choute. Algebraic gossip: A network coding approach to optimal multiple rumor mongering. *IEEE Transactions on Information Theory*, 52(6):2486–2507, June 2006.

[34] O. Dousse, M. Franceschetti, and P. Thiran. Information theoretic bounds on the throughput scaling of wireless relay networks. In *Proc. of IEEE INFOCOM 2005*, Miami, FL, Mar. 2005.

[35] M. Draief and A. Ganesh. Efficient routing in poisson small-world networks. *Journal of Applied Probability*, To appear.

[36] W. Du, J. Deng, Y. S. Han, and P. K. Varshney. A pairwise key predistribution scheme for wireless sensor networks. In *Proc. of ACM CCS'03*, Washingtion, DC, Oct. 2003.

[37] A. Eryilmaz, A.Ozdaglar, and M. Medard. On delay performance gains from network coding. In *Proc. of the Conference on Information Sciences and Systems*, Princeton, NJ, Mar. 2006.

[38] L. Eschenauer and V.D. Gligor. A key-management scheme for distributed sensor networks. In *Proc. of ACM CCS'02*, Washingtion, DC, Nov. 2002.

[39] L. Eschenauer, V.D Gligor, and J.S. Baras. On trust establishment in mobile ad-hoc networks. In *Proc. of the Security Protocols Workshop*, Cambridge, UK, April 2002.

[40] Q. Fang, J. Gao, and L. Guibas. Locating and bypassing routing holes in sensor networks. In *INFOCOM'04*, Hong Kong, China, March 2004.

[41] S. Farrell and V. Cahill. *Delay- and Disruption-Tolerant Networking*. Artech House, Boston, MA, 2006.

[42] C. Fragouli and E. Soljanin. *Network coding: Fundamentals (Foundations and Trends in Networking)*. Now Publishers Inc., Boston, MA, 2007.

[43] C. Fragouli, J. Widmer, and J.-Y. Le Boudec. Efficient broadcasting using network coding. *IEEE Transactions on Networking*, 16(2):450–463, April 2008.

[44] P. Fraigniaud, C. Gavoille, and C. Paul. Eclecticism shrinks even small worlds. In *Proc. of PODC 2004*, Newfoundland, Canada, July 2004.

[45] M. Franceschetti, L. Booth, M. Cook, R. Meester, and J. Bruck. Percolation of multi-hop wireless networks. Technical Report UCB/ERL M03/18, EECS Department, University of California, Berkeley, 2003.

[46] M. Franceschetti, O. Dousse, D. Tse, and P. Thiran. Closing the gap in the capacity of wireless networks via percolation theory. *IEEE Transactions on Information Theory*, 53(3):1009–1018, March 2007.

[47] M. Franceschetti and R. Meester. Navigation in small-world networks: a scale-free continuum model. *Journal of Applied Probability*, 43(4):1173–1180, Dec. 2006.

[48] A. El Gamal, J. Mammen, B. Prabhakar, and D. Shah. Throughput-delay trade-off in wireless networks. In *Proc. of IEEE INFOCOM 2004*, Hong Kong, China, Mar. 2004.

[49] A. El Gamal, J. Mammen, B. Prabhakar, and D. Shah. Optimal throughput-delay scaling in wireless networks - part i: the fluid model. *IEEE Transactions on Information Theory*, 52(6):2568–2592, Jun. 2006.

[50] A. El Gamal, J. Mammen, B. Prabhakar, and D. Shah. Optimal throughput-delay scaling in wireless networks - part ii: Constant-size packets. *IEEE Transactions on Information Theory*, 52(11):5111–5116, Nov. 2006.

[51] P. Ganesan and M. Seshadri. On cooperative content distribution and the price of barter. In *Proc. of ICDCS 2005*, Columbus, OH, June 2005.

[52] M. Ghaderi, D. Towsley, and J. Kurose. Reliability benefit of network coding. Tech. Report 07-08, Computer Science Department, University of Massachusetts Amherst, February 2007.

[53] A. Ghosh. Estimating coverage holes and enhancing coverage in mixed sensor networks. In *LCN'04*, Washington, DC, Nov. 2004.

[54] J. Golbeck. *Trust on the World Wide Web: A survey*. Now Publishers, Delft, The Netherlands, 2006.

[55] M. Gondran and M. Minoux. *Graphs and Algorithms*. Addison Welsey, Chichester, 1984.

[56] M. Gondran and M. Minoux. *Graphs, Dioids, and Semirings: New Models and Algorithms*. Springer, 2008.

[57] T. G. Griffin and J. L. Sobrinho. Metarouting. In *Proc. of ACM SIGCOMM 2005*, Philadelphia, Pennsylvania, USA, August 2004.

[58] M. Grossglauser and M. D. Tse. Mobility increases the capacity of ad hoc wireless networks. In *Proc. of INFOCOM 2001*, Anchorage, Alaska, April 2001.

[59] P. Gupta and P. R. Kumar. Critical power for asymptotic connectivity in wireless networks. In M. McEneaney, G. Yin, and Q. Zhang, editors, *Stochastic Analysis, Control, Optimization and Applications: A Volume in Honor of W.H. Fleming*. Birkhauser, Boston, 1998.

[60] P. Gupta and P. R. Kumar. The capacity of wireless networks. *IEEE Transactions on Information Theory*, 46(2):388–404, March 2000.

[61] T. Hagerup and C. Rüb. A guided tour of chernoff bounds. *Information Processing Letters*, 33(6):305–308, Feb. 1990.

[62] P. Hall. *Introduction to the Theory of Coverage Processes*. John Wiley Sons Inc., New York, 1988.

[63] R. Hekmat. *Ad-hoc Networks: Fundamental Properties and Network Topologies*. Springer, New York, 2006.

[64] J. M. Hellerstein, W. Hong, S. Madden, and K. Stanek. Beyond average: Towards sophisticated sensing with queries. In *Proc. of the 2nd International Workshop on Information Processing in Sensor Networks (IPSN'03)*, Palo Alto, California, March 2003.

[65] A. Helmy. Small worlds in wireless networks. *IEEE Communications Letters*, 7(10):490–492, Oct. 2003.

[66] J. Herdtner and E. Chong. Throughput-storage tradeoff in ad hoc networks. In *Proc. of IEEE INFOCOM 2005*, Miami, FL, Mar. 2005.

[67] T. Ho, R. Koetter, M. Medard, D. R. Karger, and M. Effros. The benefits of coding over routing in a randomized setting. In *Proc. of ISIT 2003*, Yokohama, Japan, June-July 2003.

[68] C. Hsin and M. Liu. Self-monitoring of wireless sensor networks. *Journal of Computer Communications special issue on Sensor Networks*, 29(4):462–476, February 2006.

[69] C. F. Huang and Y. C. Tseng. The coverage problem in a wireless sensor network. In *Proc. of the 2nd ACM international Workshop on Wireless Sensor Networks and Applications (WSNA '03)*, San Diego, CA, September 2003.

[70] D. Huang, M. Mehta, A. van de Liefvoort, and D. Medhi. Modeling pairwise key establishment for random key predistribution in large-scale sensor networks. *IEEE/ACM Trans. on Networking*, 15(5):1204–1215, Oct. 2007.

[71] J. Hubaux, L. Buttyan, and S. Capkun. The quest for security in mobile ad hoc networks. In *MobiHoc 2001*, Long Beach, CA, Oct. 2001.

[72] S. Ioannidis, A. Chaintreau, and L. Massoulie. Optimal and scalable distribution of content updates over a mobile social network. In *Proc. of INFOCOM 2009*, Rio de Janeiro, Brazil, April 2009.

[73] S. Jaggi, P. A. Chou, and K. Jain. Low complxity optimal algebraic multicast codes. In *Proc. of IEEE International Symposium on Information Theory*, Yokohama, Japan, June-July 2003.

[74] S. Janson, T. Luczak, and A. Rucinski. *Random Graphs*. John Wiley & Sons, New York, 2000.

[75] J.J. Jaramillo and R. Srikant. DARWIN: Distributed and adaptive reputation mechanism for wireless networks. In *Proc. of MobiCom 2007*, Montreal, Quebec, Canada, Sept. 2007.

[76] P. R. Jelenkovic, P. Momcilovic, and M. S. Squillante. scalability of wireless networks. *IEEE Transactions on Networking*, 15(2):295–308, April 2007.

[77] A. Jøsang. An algebra for assessing trust in certification chains. In *Proc. of NDSS'99*, San Diego, CA, Feb. 1999.

[78] S. Kapoor and X. Li. Proximity structures for geometric graphs. In *WADS 2003*, Ottawa, Canada, July 2003.

[79] H. Karl and A. Willig. *Protocols and Architectures for Wireless Sensor Networks*. John Wiley Sons Inc., New York, 2005.

[80] M. Kaufmann, J. F. Sibeyn, and T. Suel. Derandomizing algorithms for routing and sorting on meshes. In *Proc. of 5th ACM-SIAM symposium on Discrete algorithms*, Arlington, Virginia, January 1994.

[81] A. Keshavarz-Haddadt and R. Riedi. Bounds on the benefit of network coding: Throughput and energy saving in wireless networks. In *Proc. of INFOCOM 2008*, Phoenix, AZ, April 2008.

[82] H. Kesten and V. Sidoravicius. The spread of a rumor or infection in a moving population. *Annals of Probability*, 33(6):2402–2462, 2005.

[83] J. Kleinberg. Navigation in a small world. *Nature*, 406:845, August 2000.

[84] J. Kleinberg. The small-world phenomenon: An algorithmic perspective. In *Proc. 32nd ACM Symposium on Theory of Computing*, Portland, Oregon, May 2000.

[85] J. Kleinberg. Small-world phenomena and the dynamics of information. *Advances in Neural Information Processing Systems*, 14, 2001.

[86] J. Kleinberg. Complex networks and decentralized search algorithms. In *Proc. of the International Congress of Mathematicians (ICM)*, Madrid, Spain, August 2006.

[87] L. Kleinrock and J. A. Silvester. Optimum transmission radii in packet radio networks or why six is a magic number. In *Proc. of the National Telecommunications Conference*, Dec. 1978.

[88] G. Kossinets, J. Kleinberg, and D. Watts. The structure of information pathways in a social communication network. In *Proc. of KDD 2008*, Las Vegas, NE, August 2008.

[89] S.R. Kulkarni and P. Viswanath. Throughput scaling in heterogeneous networks. In *Proc. of IEEE ISIT 2003*, Kanagawa, Japan, June 2003.

[90] S.R. Kulkarni and P. Viswanath. A deternimistic approach to throughput scaling in wireless networks. *IEEE Transactions on Information Theory*, 50(6):1041–1049, Jun. 2004.

[91] R. Kumar, M. Wolenetz, B. Agarwalla, J. Shin, P. Hutto, A. Paul, and U. Ramachandran. Fuse: A framework for distributed data fusion. In *Proc. of The First ACM Conference on Embedded Networked Sensor Systems (Sensys 03)*, Los Angeles, California, November 2003.

[92] S. Kumar, T. H. Lai, and A. Arora. Barrier coverage with wireless sensors. In *MobiCom'05*, Cologne, Germany, August 2005.

[93] S. Kumar, T. H. Lai, and J. Balogh. On k-coverage in a mostly sleeping sensor network. In *MobiCom'04*, Philadelphia, PA, Oct. 2004.

[94] M. Kunde. Block gossiping on grids and tori: Deterministic sorting and routing match the bisection bound. In *Proc. of 1st European Symposium on Algorithms*, Honnef, Germany, Sept. 1993.

[95] E. Lebhar and N. Schabanel. Almost optimal decentralized routing in long-range contact networks. In *Proc. of ICALP 2004*, Turku, Finland, July 2004.

[96] T. Lengauer and D. Theune. Unstructured path problems and the making of semirings. In *Algorithms and Data Structures: 2nd Workshop, WADS'91*, Ottawa, Canada, Aug. 1991.

[97] M. Leonov. Polyboolean library (2004). http://www.complex-a5.ru/polyboolean/.

[98] X. Li, G. Calinescu, P. Wan, and Y. Wang. Localized delaunay triangulation with applications in wireless ad hoc networks. *IEEE Transactions on Parallel and Distributed Systems*, 14(10):1035–1047, Oct. 2003.

[99] D. Liben-Nowell, J. Novak, R. Kumar, P. Raghavan, and A. Tomkins. Geographic routing in social networks. *Proceedings of the National Academy of Sciences*, 102(33):11623–11628, August 2005.

[100] X. Lin, G. Sharma, R. Mazumdar, and N.B. Shroff. Degenerate delay-capacity trade-offs in ad hoc networks with brownian mobility. *IEEE/ACM Transactions on Networking*, 52(6):2777–2784, June 2006.

[101] X. Lin and N.B. Shroff. The fundamental capacity-delay tradeoff in large mobile ad hoc networks. In *Proc. of Third Annual Mediterranean Ad Hoc Networking Workshop*, Bodrum, Turkey, June 2004.

[102] Y. Lin, B. Li, and B. Liang. Stochastic analysis of network coding in epidemic routing. *IEEE Journal on Selected Areas in Communications*, 26(5):794–808, June 2008.

[103] Y.-D. Lin and Y.-C. Hsu. Multihop cellular: a new architecture for wireless communications. In *Proc. of INFOCOM 2000*, Tel Aviv, Israel, March 2000.

[104] B. Liu, Z. Liu, and D. Towsley. On the capacity of hybrid wireless networks. In *Proc. of INFOCOM 2003*, San Francisco, CA, April 2003.

[105] J. Liu, D. Goeckel, and D. Towsley. Throughput order of ad hoc networks employing network coding and broadcasting. In *Proc. of Milcom*, Washington, DC, Oct. 2006.

[106] J. Liu, D. Goeckel, and D. Towsley. Bounds on the gain of network coding and broadcasting in wireless networks. In *Proc. of INFOCOM 2007*, Anchorage, Alaska, May 2007.

[107] Zhengye Liu, Hao Hu, Yong Liu, Keith Ross, Yao Wang, and Markus Mobius. P2P trading in social networks: The value of staying connected. In *Proc. of INFOCOM 2010*, San Diego, CA, March 2010.

[108] M. Lu and J. Wu. Opportunistic routing algebra and its applications. In *Proc. of INFOCOM 2009*, Rio de Janeiro, Brazil, April 2009.

[109] G.S. Manku, M. Naor, and U. Wieder. Know thy neighbor's neighbor: The power of lookahead in randomized p2p networks. In *Proc. of STOC 2004*, Chicago, IL, June 2004.

[110] C. Martel and V. Nguyen. Analyzing kleinberg's (and other) small-world models. In *Proc. of PODC 2004*, Newfoundland, Canada, July 2004.

[111] S. Marti, T. J. Giuli, K. Lai, and M. Baker. Mitigating routing misbehavior in mobile ad hoc networks. In *Proc. of MobiCom 2000*, Boston, MA, August 2000.

[112] T. R. Mathies. Percolation theory and computing with faulty arrays of processors. In *Proc. of 3rd ACM-SIAM Symposium on Discrete Algorithms*, Orlando, FL, January 1992.

[113] S. McCanne and S. Floyd. Network simulator version 2. http://www.isi.edu/nsnam/ns.

[114] R. Meester and R. Roy. *Continuum Percolation*. Cambridge University Press, Cambridge, 1996.

[115] F. Milan, J. J. Jaramillo, and R. Srikant. Achieving cooperation in multihop wireless networks of selfish nodes. In *Proc. of 2006 workshop on Game theory for communications and networks*, Pisa, Italy, Oct. 2006.

[116] S. Milgram. The small world problem. *Psychology Today*, 1:61–67, May 1967.

[117] F. S. Mishkin. *The Economics of Money, Banking and Financial Markets (7th edn.)*. Addison-Wesley Pub. Co., Reading, MA, 2003.

[118] M. Mohri. Semiring frameworks and algorithms for shortest-distance problems. *Journal of Automata, Languages and Combinatorics*, 7(3):321–350, 2002.

[119] C. R. Murthy and B. S. Manoj. *Ad Hoc Wireless Networks: Architectures and Protocols*. Prentice Hall, New Jersey, 2004.

[120] M. Neely and E. Modiano. Capacity and delay tradeoffs for ad-hoc mobile networks. *IEEE Transactions on Information Theory*, 51(6):1917–1937, June 2005.

[121] M. Neely, E. Modiano, and C. E Rohrs. Dynamic power allocation and routing for time varying wireless networks. *IEEE Journal on Selected Areas in Communications*, 23(1):89–103, January 2005.

[122] M. J. Neely. Dynamic power allocation and routing for satellite and wireless networks with time varying channels. PhD thesis, Massachusetts Institute of Technology, 2003.

[123] A. Okabe. *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*. Wiley, New York, 2000.

[124] F. Oliviero and S.P. Romano. A reputation-based metric for secure routing in wireless mesh networks. In *Proc. of IEEE GLOBECOM 2008*, New Orleans, LA, Dec. 2008.

[125] Christian Ortolf, Christian Schindelhauer, and Arne Vater. Paircoding: Improving file sharing using sparse network codes. In *Proc. of ICIW 2009*, Venice, Italy, May 2009.

[126] Gergely Palla, Imre Derenyi, Illes Farkas, and Tamas Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435:814–818, June 2005.

[127] S.-J. Park, R. Vedantham, R. Sivakumar, and I. F. Akyildiz. A scalable approach for reliable downstream data delivery in wireless sensor networks. In *Proc. of MobiHoc'04*, Roppongi, Japan, May 2004.

[128] D. Peleg. *Distributed Computing: A Loaclity-Sensitive Approach*. Society for Industrial and Applied Mathematics (SIAM), 2000.

[129] M. Penrose. A strong law for the longest edge of the minimal spanning tree. *The Annals of Probability*, 27(1):246–260, Jan. 1999.

[130] M.D. Penrose. *Random Geometric Graphs*. Oxford University Press, Oxford, 2003.

[131] C.E. Perkins. *Ad Hoc Networking*. Addison Wesley Professional, 2000.

[132] M. O. Pervaiz, M. Cardei, and J. Wu. Routing security in ad hoc wireless networks. In S. Huang, D. MacCallum, and D.-Z. Du, editors, *Network Security*. Springer, 2008.

[133] A. Reznik, S.R. Kulkarni, and S. Verdú. Scaling laws in random heterogeneous networks. In *Proc. of ISIT 2004*, Chicago, USA, June 2004.

[134] A. Reznik, S.R. Kulkarni, and S. Verdú. A "small world" approach to heterogeneous networks. *Communications in Information and Systems*, 3(4):325–348, September 2004.

[135] N. Ben Salem, L. Buttyan, J.P. Hubaux, and M. Jakobsson. Node cooperation in hybrid ad hoc networks. *IEEE Transactions on Mobile Computing*, 5(4):1–12, April 2006.

[136] Y. Sankarasubramaniam, O. B. Akan, and I. F. Akyildiz. Esrt: Event-to-sink reliable transport in wireless sensor networks. In *Proc. of MobiHoc'03*, Annapolis, Maryland, June 2003.

[137] S.Chessa and P.Santi. Comparison based system-level fault diagnosis in ad-hoc networks. In *IEEE 20th Symp. on Reliable Distributed Systems (SRDS)*, New Orleans, LA, October 2001.

[138] M. Sharifzadeh and C. Shahabi. Utilizing voronoi cells of location data streams for accurate computation of aggregate functions in sensor networks. *GeoInformatica*, 10(1):9–36, March 2006.

[139] G. Sharma and R. Mazumdar. Hybrid sensor networks: A small world. In *Proc. of ACM MobiHoc 2005*, Urbana-Champaign, IL, May 2005.

[140] G. Sharma, R. R. Mazumdar, and N. B. Shroff. Delay and capacity trade-offs in mobile ad hoc networks: A global perspective. In *Proc. of IEEE INFOCOM 2006*, Barcelona, Spain, April 2006.

[141] G. Sharma, N.B. Shroff, and R.R. Mazumdar. Hybrid sensor and mesh networks: Paradigms for fair and energy efficient communication. In *Proc. of WiMesh 2006*, Reston, Virginia, September 2006.

[142] U. Shevade, H. H. Song, L. Qiu, and Y. Zhang. Incentive-aware routing in DTNs. In *Proc. of IEEE ICNP 2008*, Orlando, FL, October 2008.

[143] J. L. Sobrinho. Algebra and algorithms for qos path computation and hop-by-hop routing in the internet. *IEEE/ACM Transactions on Networking*, 10(4):541–550, August 2002.

[144] J. L. Sobrinho. Network routing with path vector protocols: Theory and applications. In *Proc. of ACM SIGCOMM 2003*, Karlsruhe, Germany, August 2003.

[145] J. L. Sobrinho. An algebraic theory of dynamic network routing. *IEEE/ACM Transactions on Networking*, 13(5):1160–1173, Oct. 2005.

[146] T. Spyropoulos, K. Psounis, and C. S. Raghavendra. Efficient routing in intermittently connected mobile networks: The multi-copy case. *IEEE Transaction on Networking*, 16(1):77–90, Feb. 2008.

[147] V. Srinivasan, P. Nuggehalli, C.-F. Chiasserini, and R. R. Rao. Cooperation in wireless ad hoc networks. In *Proc. of INFOCOM 2003*, San Francisco, CA, March 2003.

[148] A. Srinivasany, J. Teitelbaumy, Huigang Liang, J. Wu, and M. Cardei. Reputation and trust-based systems for ad hoc and sensor networks. In A. Boukerche, editor, *Algorithms and Protocols for Wireless Ad Hoc Networks*. Wiley & Sons, 2008.

[149] F. Stann and J. Heidemann. Reliable data transport in sensor networks. In *Proc. of the international workshop on Sensor Net Protocols and Applications (SNPA)*, Apr. 2003.

[150] D. Stoyan, W.S. Kendall, and J. Mecke. *Stochastic Geometry and its Applications*. Wiley, New York, 2nd edition, 1995.

[151] Y.-L. Sun, W. Yu, Z. Han, and K.J.R. Liu. Information theoretic framework of trust modeling and evaluation for ad hoc networks. *IEEE JSAC*, 24(2):305–317, Feb. 2006.

[152] G. Theodorakopoulos and J. S. Baras. On trust models and trust evaluation metrics for ad hoc networks. *IEEE Journal on Selected Areas in Communications*, 24(2):318–328, Feb. 2006.

[153] D. Tian and N. D. Georganas. A coverage-preserving node scheduling scheme for large wireless sensor networks. In *Proc. of the First ACM Inyernational Workshop on Wireless Sensor Networks and Applications (WSNA 02)*, Atlanta, GA, September 2002.

[154] S. Toumpis and A. Goldsmith. Large wireless networks under fading, mobility, and delay constraints. In *Proc. of IEEE INFOCOM 2004*, Hong Kong, China, March 2004.

[155] C-Y. Wan, A. Campbell, and L. Krishnamurthy. Psfq: A reliable transport protocol for wireless sensor networks. In *Proc. ACM Int. Workshop on Sensor Networks and Architectures*, Atlanta, Sept. 2002.

[156] P.-J. Wan, C.-W. Yi, F. Yao, and X. Jia. Asymptotic critical transmission radius for greedy forward routing in wireless ad hoc networks. In *Proc. of ACM MobiHoc 2006*, Florence, Italy, May 2006.

[157] G. Wang, G. Cao, and T. La Porta. Movement-assisted sensor deployment. In *INFOCOM'04*, Hong Kong, China, March 2004.

[158] W. Wang, S. Eidenbenz, Y. Wang, and X.-Y. Li. OURS: optimal unicast routing systems in non-cooperative wireless networks. In *Proc. of MobiCom 2006*, Los Angeles, CA, Sept. 2006.

[159] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill. Integrated coverage and connectivity configuration in wireless sensor networks. In *ACM SenSys'03*, Los Angeles, CA, Nov. 2003.

[160] D. Watts and S. Strogatz. Collective dynamics of small-world networks. *Nature*, 393(4):440–442, June 1998.

[161] R.J. Williams and D.Z. Martinez. Simple rules yield complex food webs. *Nature*, 404(4):180–183, March 2000.

[162] H. Wu, R. Fujimoto, R. Guensler, and M. Hunter. MDDV: a mobility-centric data dissemination algorithm for vehicular networks. In *Proc. of ACM VANET 2004*, Philadelphia, PA, Oct. 2004.

[163] F. Xue and P. R. Kumar. *Scaling Laws for Ad Hoc Wireless Networks: An Information Theoretic Approach*. Now Publishers, Delft, The Netherlands, 2006.

[164] Y. Yang and J. Wang. Design guidelines for routing metrics in multihop wireless networks. In *Proc. of INFOCOM 2008*, Phoenix, AZ, April 2008.

[165] L. Ying, S. Yang, and R. Srikant. Optimal delay-throughput tradeoffs in mobile ad hoc networks. *IEEE Transactions on Information Theory*, 54(9):4119–4143, September 2008.

[166] M. Yu and K.K. Leung. A trustworthiness-based qos routing protocol for wireless ad hoc networks. *IEEE Transactions on Wireless Communications*, 8(4):1888–1898, April 2009.

[167] W. H. Yuen, R. D. Yates, and S.-C. Mau. Exploiting data diversity and multiuser diversity in noncooperative mobile infostation networks. In *Proc. of INFOCOM 2003*, San Francisco, CA, March 2003.

[168] C. Zhang, Y. Song, and Y. Fang. Modeling secure connectivity of self-organized wireless ad hoc networks. In *Proc. of INFOCOM 2008*, Phoenix, AZ, April 2008.

[169] C. Zhang, Y. Zhang, , and Y. Fang. Localized coverage boundary detection for wireless sensor networks. In *Proc. of the Third International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks (QShine)*, Waterloo, Canada, August 2006.

[170] Chi Zhang, Xiaoyan Zhu, and Yuguang Fang. On the improvement of scaling laws for large-scale manets with network coding. *IEEE Journal on Selected Areas in Communications*, 27(5):662–672, June 2009.

[171] Chi Zhang, Xiaoyan Zhu, and Yuguang Fang. Throughput-delay tradeoffs in large-scale manets with network coding. In *Proc. of INFOCOM 2009*, Rio de Janeiro, Brazil, April 2009.

[172] Chi Zhang, Xiaoyan Zhu, Yang Song, and Yuguang Fang. C4: A new paradigm for providing incentives in multi-hop wireless networks. In *Proc. of INFOCOM 2011*, Shanghai, China, April 2011.

[173] H. Zhang, A. Goel, and R. Govindan. Using the small-world model to improve freenet performance. In *Proc. of IEEE INFOCOM 2002*, New York, NY, June 2002.

[174] H. Zhang and J. Hou. Maintaining sensing coverage and connectivity in large sensor networks. *Wireless Ad Hoc and Sensor Network*, 1(1-2):89–123, January 2005.

[175] X. Zhang, G. Neglia, J. Kurose, and D. Towsley. On the benefits of random linear coding for unicast applications in disruption tolerant networks. In *Second Workshop on Network Coding, Theory, and Applications (NetCod)*, Boston, MA, April 2006.

[176] Xinyu Zhang and Baochun Li. DICE: a game theoretic framework for wireless multipath network coding. In *Proc. of MobiHoc 2008*, Hong Kong, China, May 2008.

[177] Y. Zhang, W. Liu, W. Lou, and Y. Fang. Securing mobile ad hoc networks with certificateless public keys. *IEEE Transactions on Dependable and Secure Computing*, 3(4):386–399, Oct.-Dec. 2006.

[178] Z. Zhang. Routing in intermittently connected mobile ad hoc networks and delay tolerant networks: Overview and challenges. *IEEE Communications Surveys and Tutorials*, 8(1):24–37, March 2006.

[179] J. Zhao, R. Govindan, and D. Estrin. Computing aggregates for monitoring wireless sensor networks. In *Proc. of the First IEEE International Workshop on Sensor Network Protocols and Applications (SNPA'03)*, Anchorage, AK, May 2003.

[180] Sheng Zhong, Jiang Chen, and Yang Richard Yang. Sprite: A simple, cheat-proof, credit-based system for mobile ad-hoc networks. In *Proc. of INFOCOM 2003*, San Francisco, CA, March 2003.

[181] Sheng Zhong and Fan Wu. On designing collusion-resistant routing schemes for non-cooperative wireless ad hoc networks. In *Proc. of MobiCom 2007*, Montreal, Quebec, Canada, Sept. 2007.

[182] P. Zimmerman. *The official PGP User's Guide*. MIT Press, Cambridge, MA, 1995.

BIOGRAPHICAL SKETCH

Chi Zhang was born in 1977 in Wuhan, Hubei, China. The older of two children, Chi grew up in Wuhan and graduated from the NO.1 Middle School affiliated to Central China Normal University in the summer of 1995. Following high school, Chi enrolled at Huazhong University of Science and Technology (HUST) in Wuhan, China in the fall of 1995. He received his B.E. and M.E. degrees in electrical and information engineering from HUST, in 1999 and 2002, respectively. Chi enrolled in the Ph.D. program in the Department of Electrical and Computer Engineering at the University of Florida in the fall of 2004, as a recipient of the University of Florida's Alumni Fellowship. He received his Ph.D. degree in electrical and computer engineering from the University of Florida in the summer of 2011. His research interests are in the areas of network protocol design, network performance analysis, and network security guarantee, particularly for wireless networks and social networks. He has published over 50 papers in prestigious journals including IEEE/ACM Transactions on Networking, IEEE Journal on Selected Areas in Communications, and IEEE Transactions on Mobile Computing, or in top networking conferences such as IEEE INFOCOM, IEEE ICNP, and IEEE ICDCS. He has served as the Technical Program Committee (TPC) members for several international conferences including IEEE GLOBECOM 2010 and 2011, IEEE ICC 2011, and IEEE PIMRC 2011. Chi is the 2011 recipient of the Gator Engineer Graduate Student Attribute Award for Creativity (one recipient per year). Chi also has been selected three times as a recipient of the travel grant from the National Science Foundation (NSF) to attend IEEE ICDCS 2008, IEEE INFOCOM 2009 and 2011. Chi is a student member of IEEE.