Robust Communication in Vehicular Ad Hoc Networks

Mark Johnson

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2010-162
http://www.eecs.berkeley.edu/Pubs/TechRpts/2010/EECS-2010-162.html

December 16, 2010

Robust Communication in Vehicular Ad Hoc Networks

by

Mark Christopher Johnson

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Engineering - Electrical Engineering and Computer Sciences

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Kannan Ramchandran, Chair
Professor Michael Gastpar
Professor Raja Sengupta

Fall 2010

Robust Communication in Vehicular Ad Hoc Networks

Abstract

Robust Communication in Vehicular Ad Hoc Networks

by

Mark Christopher Johnson

Doctor of Philosophy in Engineering-Electrical Engineering and Computer Sciences

University of California, Berkeley

Professor Kannan Ramchandran, Chair

The development of intelligent transportation systems is one of the most important strategies for making roads safer and more efficient. The key technology underpinning all such systems, from cooperative safety systems to electronic tolling, is wireless communications. Vehicles must be able to communicate with each other, and with fixed infrastructure, in real time. One promising architecture involves forming an ad hoc network among the vehicles and taking advantage of the peer-to-peer connectivity to reduce infrastructure costs.

In this dissertation, we consider the design and analysis of medium access control and network layer protocols for vehicular ad hoc networks. In particular, we examine the problem of distributing content over many hops from a source to a destination. The fundamental challenge in such systems is the inability of the vehicles to coordinate with each other. Because of the high frequency at which vehicles enter and exit the road, and the fact that vehicles may be moving at widely varying speeds, the topology is continuously changing. Therefore, centralized scheduling of transmissions to avoid collisions and global routing of packets is either technically infeasible or prohibitively expensive. We propose instead a class of uncoordinated protocols, based on randomized channel access and network coding. The key idea behind the coding is for the vehicles to transmit random combinations of all of the packets that they have received, instead of simply forwarding packets. This scheme does not require link level feedback, and thus vehicles do not need to track the identities of their current neighbors.

We first consider a minimally coordinated protocol, where the vehicles on a road are partitioned into spatial blocks. The block structure is used to funnel waves of data down the road, while the vehicles within each block randomly contend with each other for the channel. We show that when the vehicles use network coding in this scheme, the end-to-end throughput is constant, as long as the number of blocks in the network does not grow exponentially in the vehicle density. Network coding allows for the reliable transport of data over many hops at the same rate at which data could be sent over a single hop.

Then we consider a completely uncoordinated protocol, which does not even require the assignment of vehicles to spatial blocks. We begin by providing a general algorithm for computing the throughput of an arbitrary network using such a protocol. The main technical contribution is to account for the fact that intermediate nodes may temporarily not have

any new packets to transmit to their neighbors, due to the random channel access. We prove that the throughput of this wireless network is equal to the throughput of an equivalent wired network, despite the randomized medium access strategy. This result has application in a variety of other contexts, where it is often assumed for tractability that relays always have data when they use the channel.

We then apply this result to the problem of finding the throughput in an uncoordinated vehicular ad hoc network. In a congested highway, where the connectivity graph is homogeneous, a constant throughput can be provided to receivers that are located at an arbitrary distance from the source. In essence, every vehicle can be thought of as a digital repeater, despite the packet losses due to collisions.

To my parents and sister

Contents

List of Figures

List of Tables

Acknowledgments

I must first express my deep gratitude to my advisor, Prof. Kannan Ramchandran, for his support, guidance, and unfailing encouragement all these years. This work would not have been possible without his vision, assistance, and countless hours of research discussions. I deeply appreciate the fact that he never gave up on me.

I would like to thank Prof. Minghua Chen for his invaluable help during the past year. I could not have successfully completed this research without his guidance and many valuable Skype discussions. I would also like to thank my committee members, Prof. Michael Gastpar and Prof. Raja Sengupta, for their constructive and helpful feedback on my research and on my dissertation.

I must thank all of my collaborators in graduate school. In particular, I must acknowledge Prof. Francois Baccelli for the time he spent discussing the connections between queueing theory and wireless networks with me while in Berkeley, and Prof. Luca De Nardis for the time he spent working with me when I was getting started on this project.

I also thank all of my fellow members of the BASiCS lab and the Wireless Foundations Center for all of the great memories from my years in graduate school.

Finally, I thank the Hertz Foundation for their financial support, which helped make it possible for me to pursue the research presented in this dissertation.

Chapter 1

Introduction

The design and deployment of safe and efficient transportation networks is one of the most important large-scale engineering challenges of the 21st century. In the US alone, there are over 33,000 traffic fatalities per year, according to data from the National Highway Safety Administration [1]. Furthermore, the US Bureau of Transportation Statistics reports that in 2007 drivers spent a total of 4.2 billion hours delayed in traffic [2]. Not only does this time represent a loss in productivity, but it also significantly increases fuel consumption and air pollution. In the near future these problems will likely increase in magnitude, as the number of vehicles on the road worldwide is projected to double within the next decade [3].

One of the many strategies for making highways safer and less congested has been the development of intelligent transportation systems (ITS), the incorporation of information and communication technologies in vehicles and transport infrastructure. There are a large and growing number of systems, targeting a wide range of applications, that fall under the heading of ITS. For example, cooperative active safety systems enable numerous applications that can mitigate collisions and traffic fatalities, such as collision warning, emergency electronic brake lights, and slow/stopped vehicle alerts [4, 5]. Two systems have recently been developed which use smartphones in vehicles to estimate traffic conditions and travel times more accurately than by using existing roadside infrastructure [6, 7]. This traffic data can then be relayed to drivers, allowing them to plan trips more effectively, and can also used by traffic management centers to plan long term upgrades to the highway system. In a similar vein, the development of electronic tolling mechanisms will enable high occupancy toll lanes and congestion pricing, which can potentially reduce traffic congestion at peak hours [8]. The most forward looking ITS projects have even tested completely automated vehicles, which drive themselves without input from a human [9]. Such automated highway systems have long been known to increase the capacity of roads by as much as four times [10].

The common technology underlying all of these disparate systems is wireless communication. In each application, the vehicles on the road must be able to exchange information either with each other or with the fixed infrastructure units along the roadside. All real time intelligent transportation systems require a wireless link which can provide a sufficient throughput, delay, and reliability to each vehicle.

A variety of architectures have been proposed for providing wireless connectivity to vehicles. One general approach is to extend legacy networks by placing devices in vehicles that can utilize existing infrastructure. Examples include using cellular networks to collect traffic data generated by smartphones [11], and opportunistically making use of open WiFi access points to deliver data to vehicles [12]. These networks are all based on a single hop topology, as shown in Figure 1.1. The vehicles use the wireless channel to communicate directly with one or more base stations, which may be linked by a backend wired network.

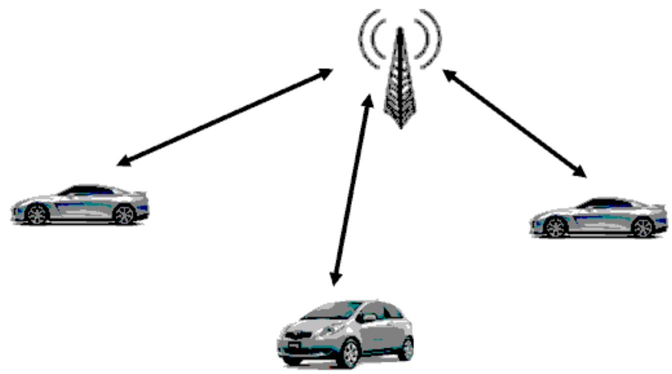

Figure 1.1: A single hop vehicular network. The vehicles communicate only with the fixed base stations along the roadside.

A second general approach, in contrast, involves forming an ad hoc network among the vehicles on the road, as shown in Figure 1.2. In this paradigm vehicles communicate directly with each other, as opposed to communicating only with fixed access points [13, 14]. Vehicular ad hoc networks (VANETs) are especially well suited for safety applications, where low latency is the most important system requirement. The development of VANETs was fostered by the decision of the FCC to allocate 75 MHz of spectrum in the 5.9 GHz band for use by Dedicated Short Range Communication (DSRC) systems, which is intended to "improve traveler safety, decrease traffic congestion, facilitate the reduction of air pollution, and help to conserve vital fossil fuels" [15]. The availability of this band has prompted research and development of a variety of ITS services that could make use of it, targeting applications beyond just vehicle safety.

In this dissertation, we consider the problem of designing communication protocols for vehicular ad hoc networks. In particular, we focus on the medium access control (MAC) and network layers of the protocol stack. The fundamental characteristic which distinguishes VANETs from other wireless networks, even other ad hoc systems such as sensor networks, is the high mobility of the individual nodes. The network topology changes at a very high rate, due to vehicles entering and exiting the road as well as the fact that vehicles often travel at very different speeds. The result is that the coordination of transmissions and routing of packets is costly and difficult. In fact, vehicles may not even know their current neighbors. While neighbor discovery and self-organization in ad hoc networks are technically feasible

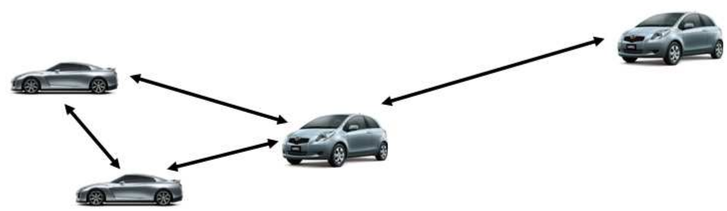

Figure 1.2: A vehicular ad hoc network. The vehicles communicate directly with each other, without requiring a base station to organize and manage the network.

in static scenarios [16, 17], they become much more challenging with high node mobility. Repeatedly executing such algorithms, each time the topology of the VANET changes, would quickly consume a large portion of the system resources.

We propose instead the use of simple uncoordinated protocols for VANETs. The key tool is to use randomization to combat the fact that no single vehicle or infrastructure element has knowledge of the global network topology. Thus, each vehicle randomly and independently accesses the channel, instead of attempting to schedule transmissions to avoid collisions and optimally reuse the available bandwidth. The only assumption is that vehicles are time synchronized, enabling the use of a slotted protocol. Similarly, instead of utilizing global routing tables, which require significant resources to initiate and maintain in a dynamic network, each node randomly mixes the packets that it transmits. This protocol does not require any link level feedback as to which packets are received and which are lost due to collisions.

We are primarily concerned with analyzing the performance provided by such uncoordinated protocols in vehicular ad hoc networks. We present a general method for computing the throughput in networks using random channel access and random linear network coding, and then apply it to a model of the topology of a VANET. Our first main contribution shows that on a congested highway, which can be modeled by a dense network with homogeneous connectivity, the throughput is independent of the number of hops between the source and destination. Thus, the network can provide a constant throughput over any distance, so long as there is the same density of vehicles. With a very simple distributed MAC protocol, the throughput is about 45% of the throughput in a fully coordinated network. Thus, the "cost of uncoordination" is a loss in performance of about a factor of 2. By using a more complex MAC protocol, such as by incorporating carrier sensing, this cost can be reduced.

The intuition behind this result is that network coding, the mixing of packets at intermediate nodes, makes the uncoordinated network robust against packet losses. Without coding, in contrast, the throughput would degrade with distance due to the inability to coordinate among nodes. In effect, each vehicle functions as a digital repeater.

In the course of studying the throughput of uncoordinated networks, we address the problem of starvation, the possibility that intermediate nodes in a multihop network may temporarily not have any new information because of the random ordering of transmissions.

We then use queueing theory to demonstrate our second main contribution, that starvation does not affect the end-to-end throughput. This result has application in a number of fields extending beyond vehicular networks. In particular, it is relevant to the study of CSMA networks, where it is often assumed for analytic tractability that all nodes are in a saturated state, meaning that they always have new information to transmit.

This dissertation is organized as follows. In Chapter 2, we give an explicit formulation of the multihop communication problem that we will study, and discuss the uncoordinated MAC and network layer protocols that we will apply to this problem. In Chapter 3 we propose a spatially synchronized protocol for multihop communication in VANETs, which requires a minimum level of coordination among the vehicles to implement. The goal in this chapter is to explore the tradeoff between coordination and performance, by studying a protocol with significant randomization but still a small degree of coordination. It also provides a benchmark, against which we can then compare a completely uncoordinated protocol. In Chapter 4 we present a general method for computing the end-to-end throughput of multihop networks using an uncoordinated MAC protocol and network coding. Then in Chapter 5 we use this algorithm to calculate the throughput of an uncoordinated VANET. We conclude the dissertation in Chapter 6 with a discussion of future work.

Chapter 2

Multihop Content Distribution in Vehicular Ad Hoc Networks

In this dissertation, we focus on multihop content distribution in vehicular ad hoc networks. There are two main features of this problem which distinguish it from other VANET communication problems. The first is that data must be transmitted over large distances, and thus a multihop communication protocol will be used. There are several scenarios where this is true. Clearly, one example is when the source and destination are separated by such a large distance that single hop communication is not possible. A second example is when limitations on the transmit power make it necessary to relay data over multiple hops. We can imagine an emergent service being deployed in unlicensed bands with strict power limitations, as opposed to purchasing licensed spectrum where it might be possible to transmit data directly from source to destination.

The second unique feature is that a relatively large amount of data must be sent, and therefore it must be divided into numerous packets. One of the key results of this dissertation will be to show that the data can be distributed over many hops without a loss in performance. There are a variety of applications where a large volume of data must be delivered to vehicles on a road. As one example, we consider applications where the VANET is being used to distribute data that is meaningful to drivers of vehicles. For example, the California Department of Transportation has cameras placed near bridges and freeway interchanges in the Bay Area [18]. We could use the vehicular network to distribute live video from such cameras to vehicles, so drivers can make decisions about which road to take or whether to delay their trip. Similarly, we could distribute a real-time map showing the average speeds on the roads in a metropolitan area, similar to the traffic map provided by Google maps. As a second general class of applications, we can consider the vehicular network to be a backbone which is used to distribute data that is independent of the fact that the network is formed on a highway. In the realm of entertainment, for example, we could use the multihop network to stream a television show which passengers in the vehicles would watch during their trip.

In order to situate this multihop content distribution problem within the larger field of vehicular networking, we briefly compare it to safety applications, which are one of the technologies driving the deployment of VANETs. In safety systems, the data is usually small enough to be contained in a single packet, for example a warning that a vehicle is braking sharply or changing lanes. In addition, safety messages are usually only relevant to vehicles that are near the source. For example, when a vehicle changes lanes, only the vehicles in the adjacent lane which are in the driver's blind spot need be warned. Thus, safety messages only need to be transmitted over a small number of hops, if not just a single hop. Due to these considerations, it suffices to flood the safety messages over the region where they are relevant, and to use repetition coding to combat channel losses. For this class of applications, there is little benefit to using the more powerful coding strategies that are examined in this dissertation.

The rest of this chapter is organized as follows. In Section 2.1 we formulate the multihop communication problem. In Sections 2.2 and 2.3, we discuss the MAC and network layer protocols, respectively, that will be used in subsequent chapters to design a scheme for multihop VANET communication. In particular, in Section 2.3 we provide background material on network coding, which is the key component of our solution. In Section 2.4 we review some related work on content distribution in VANETs.

2.1 Multihop Networks

We will begin by considering a unicast communication problem, where a fixed base station wants to use the VANET to send data to a destination vehicle that is far down the road, as illustrated in Figure 2.1. We will also refer to this base station as an infostation, to emphasize that it is the source of the data, in contrast to cellular telephone networks where one of the mobile devices is the source.

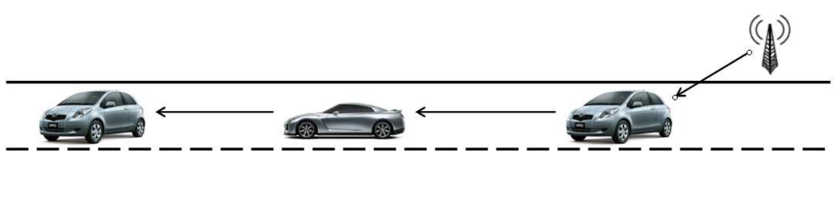

Figure 2.1: We consider a unicast communication problem in a multihop wireless vehicular ad hoc network.

We make several simplifying assumptions. Primarily, we assume throughout this dissertation that all vehicles use the same transmit power and thus have a common transmission range R meters. We also assume that the physical layer is modeled by the Gupta and Kumar protocol model for packet reception [19]. In that model, a vehicle can receive packets transmitted by other vehicles that are within a distance R of itself, but cannot receive any packets transmitted by vehicles further than a distance R from itself. If two vehicles within a distance R of the receiver transmit at the same time, the receiver hears a collision and neither packet can be decoded. In the following chapters, we will describe various models of the spatial distribution of vehicles in more detail. For now, it suffices that we will assume that there are no gaps longer than R between adjacent vehicles, and therefore there are no disconnected components of the graph.

We also assume here that the vehicles and base stations use omnidirectional antennas. A more advanced system could improve the throughput by using directional antennas. Even placing a relatively simple two antenna array at each vehicle could reduce interference by focusing the radiated power in the direction of the destination. The analysis of a VANET with directional antennas is left for future work. Finally, all packets in the network will have a fixed length.

As discussed in Chapter 1, the most important property of a VANET is that centralized coordination and control is either extremely expensive or technically infeasible. For this reason, we will design distributed MAC and network layer protocols for this problem.

2.2 MAC Layer Protocols

The vehicles cannot coordinate when accessing the channel, due to the lack of knowledge of the topology. Instead, we will utilize randomized MAC protocols, such as Aloha [20] and carrier sense multiple access [21] based schemes.

2.2.1 Slotted Aloha

The slotted Aloha protocol was originally proposed for use in fully connected computer networks [20]. The basic idea is that nodes access the common communication channel independently of one another. Whenever a terminal has data to send, it transmits a packet in the next time slot. This protocol can be readily adapted to wireless vehicular networks. Time is partitioned into slots, and vehicles randomly choose to either transmit or receive in each slot. Vehicle i will transmit a packet in every time slot with probability p_i, independent of all other vehicles and all other time slots, and with probability $(1 - p_i)$ will not transmit. Whenever a node does not transmit, it attempts to receive a packet. If exactly one neighboring vehicle within its transmission range is transmitting, then that packet is successfully demodulated and decoded. If more than one neighbor transmits, the vehicle hears a collision and is unable to decode any of the transmitted packets.

This protocol requires a very minor degree of coordination, because all vehicles must have access to a common reference clock so that the time slot boundaries are synchronized. This could be implemented, for example, by placing a GPS receiver in each vehicle, which gives them access to a universal time basis. Alternatively, a single high-power transmitter could broadcast beacons on an auxiliary channel. Since this transmitter is only demarcating the boundaries of the time slots, and not communicating information, only a very small bandwidth would be required.

2.2.2 Unslotted Aloha

The unslotted Aloha protocol for computer networks is similar to slotted Aloha, except that the terminals now transmit each packet immediately after it is generated, instead of waiting until the beginning of the next time slot. Thus, the transmission times are now asynchronous. When two terminals transmit, the packets may partially overlap, which also results in a collision and loss of information. In slotted Aloha, on the other hand, all nodes align their transmissions with the same slot boundaries, and therefore packets either completely overlap in time or are received without collision.

In the context of wireless VANETs, we will use unslotted Aloha to refer to a protocol where each vehicle independently chooses to either transmit or receive in every time slot, but the vehicles do not have access to a common reference clock. Each vehicle implements the slotted Aloha protocol described in the previous section, but the slot boundaries are shifted by a random offset. In this case, transmitted packets can partially overlap, just as in the unslotted Aloha protocol from computer networking.

2.2.3 Carrier-Sense MAC

In Aloha-style protocols, many packets are corrupted by collisions, since neighboring nodes frequently transmit at the same time. The performance can be improved, without requiring coordinated scheduling, by having each vehicle check whether the channel is in use before transmitting. When a node chooses to transmit, it will first sense the channel. If the channel is free, it proceeds with its transmission. On the other hand, if the channel is busy, then the vehicle will back-off by a random amount of time before trying to transmit again. The design of carrier sense multiple access (CSMA) networks is a very active research area, with substantial work on optimizing the channel access and back-off parameters [22, 23, 24].

While carrier sensing can mitigate the losses inherent to Aloha protocols, collisions may still occur due to the *hidden node* problem. This occurs when two transmitters do not sense each other's transmissions, but a receiver is within range of both of them. Thus, the two transmitters may decide to transmit concurrently, which results in the receiver hearing a collision. However, recent work has shown that any network can be made hidden node-free, by adjusting the carrier sensing range to be large enough in proportion to the transmission range[1] [25].

We also note that more advanced carrier-sense protocols often include collision avoidance mechanisms. In one standard scheme, a transmitter sends a *request to send* (RTS) packet before transmitting. If the receiver hears the RTS packet, it replies with a *clear to send* (CTS) packet. The transmitter only begins to transmit when it receives the CTS packet. In this dissertation, we do not consider the use of such collision avoidance protocols for several reasons. First, our goal is to show that a significant throughput can be achieved with very simple distributed protocols. Second, in Chapter 4 we will consider scenarios where vehicles

[1] This is true when there is a line of sight path between all pairs of nodes, and the hidden node problem occurs due to propagation losses. If there are obstacles between some nodes, then the hidden node problem may be unavoidable.

do not know the identity of their neighbors and cannot address each other, which makes it impossible to use RTS/CTS packets.

2.3 Network Layer Protocols

The medium access layer only defines how the vehicles will access the channel. The network layer describes how information is moved from the source to destination. In this section, we discuss several network layers that are compatible with the requirement that no single node has global knowledge of the network topology. This lack of coordination prevents the computation and distribution of centralized routing tables

We first note that by identifying themselves to their neighbors, vehicles could learn the local topology of the network. By propagating this information throughout the network, it would be possible in theory to learn the complete topology and then execute a predefined algorithm to construct common routing tables. However, such a scheme will consume a significant amount of system resources, since the topology of the VANET changes frequently due to vehicles entering and exiting the road, and moving relative to each other. If vehicles enter and exit the road on the same timescale as the algorithm that learns the topology, then the network will not be able to maintain accurate routing tables. We instead consider *distributed* network layer protocols. By this, we refer to a general class of protocols where vehicles select what data to include in a transmitted packet independently of each other. In this section we outline two such protocols, distributed routing and random network coding.

2.3.1 Distributed Routing

One possible network protocol is distributed store-and-forward routing. In this scheme, each vehicle maintains a buffer of all of the packets that it has received. When a vehicle accesses the channel, it randomly selects one of the stored packets to transmit, independently of the other vehicles.

There are a variety of heuristics that could be used to select packets to transmit. In the simplest case, a vehicle would uniformly choose one of the packets stored in its buffer. A more advanced protocol could choose the more recently received packets with a higher probability than those that have been in the buffer longer, since the newer packets are less likely to have been received by neighboring vehicles.

However, all variants of distributed routing suffer due to the lack of coordination. Because vehicles do not know which packets are present at any other vehicles, and do not receive feedback about which transmissions were received correctly and which were lost to collision, some transmissions are not useful. A transmitter may send a packet that its neighbors already possess, when there are other packets in its buffer that its neighbors do not have yet. The destination is essentially performing a coupon collecting operation, and may receive numerous copies of one source packet before receiving any copies of another source packet.

2.3.2 Random Network Coding

A second general option is for the vehicles to use network coding. The fundamental idea of network coding is to allow mixing of information at intermediate nodes, instead of simply routing it. The seminal paper by Ahlswede, et al. [26] proved that *a network code which allows a source to send h units of data per unit time through the network to a set of destinations exists if and only if the maximum flow between the source and each destination are all greater than or equal to h.*

The maximum flow is the largest rate at which information can be delivered from a source to a destination, while respecting the capacities of each link. With network coding, it is possible for all of the destinations to receive data at a rate equal to the smallest maximum flow between the source and any destination, which is clearly the largest possible multicast rate. In general, this is not possible if the intermediate nodes perform only routing. More recently, it has been shown that the maximum rate can be achieved with high probability if each node simply transmits a *random* linear combination of its inputs (using coefficients drawn from a sufficiently large finite field) [27, 28]. Randomized network coding can be practically performed without any centralized coordination [27], and is very useful in cases in which the topology of the network is unknown or dynamic [29].

In a vehicular network, the implementation of random linear network coding is particularly straightforward. Each packet in the network consists of simply a linear combination of all of the source packets. For example, consider a scenario with three source packets labeled X_1, X_2, X_3. Each X_i consists of a long vector of source symbols, which are elements in a finite field. The packets transmitted by vehicles will be of the form $Y = \alpha_1 X_1 + \alpha_2 X_2 + \alpha_3 X_3$, where the mixing coefficients $\{\alpha_i,\ i = 1, 2, 3\}$ are elements in the same finite field as the source symbols. The actual packet will consist of a header, which contains the coefficients α_i, and the data, Y.

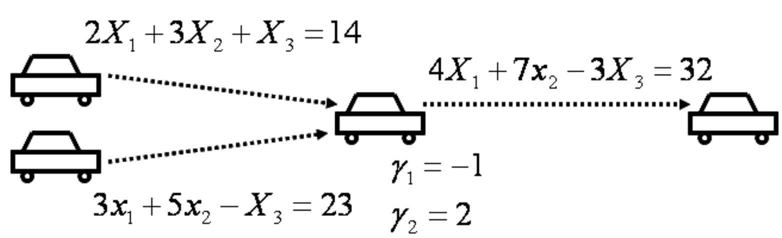

Figure 2.2: In a system based on random linear network coding, the intermediate nodes transmit linear combinations of the packets that they have received.

Every vehicle maintains a buffer containing all of the linearly independent packets that it has received. When a vehicle chooses to transmit, it generates a set of random mixing coefficients, uses those coefficients to combine the packets in its buffer, and transmits the result over the channel. Figure 2.2 provides a simple example. The relay vehicle in the middle has received two packets from its neighbors to the left, which represent the equations $Y_1 = 2X_1 + 3X_2 + X_3 = 14$ and $Y_2 = 3X_1 + 5X_2 - X_3 = 23$. While the coefficients in this example are integers, in actual implementations the coefficients will be elements in a finite field. When the relay decides to transmit a packet, it randomly generates two local mixing coefficients γ_1 and γ_2, and uses them to form the new equation

$$\gamma_1 Y_1 + \gamma_2 Y_2 = \gamma_1(2X_1 + 3X_2 + X_3) + \gamma_2(3X_1 + 5X_2 - X_3) = \gamma_1(14) + \gamma_2(23)$$

For example, if $\gamma_1 = -1$ and $\gamma_2 = 2$, then the new equation is $4X_1 + 7X_2 - 3X_3 = 32$, as shown in Figure 2.2. The header of the new packet must contain only the absolute coefficient vector $(4, 7, -3)$. The local coefficients (γ_1, γ_2) do not need to be propagated through the network.

Whenever a vehicle receives a new packet, it will check whether or not the packet is a linear combination of the packets already stored in the vehicle's buffer. If the packet is a linear combination of packets that the vehicle has already received, then the vehicle can disregard the new packet. On the other hand, if the new packet is linearly independent of the set of prior packets, then the vehicle adds the new packet to the buffer. Once the number of packets in a vehicle's buffer is equal to the number of source packets, the vehicle can recover those source packets by solving a system of equations.

Clearly, the overhead required to transmit the coefficients in each packet is proportional to the number of source packets. In a streaming application, where there is essentially an infinite amount of source data, the source packets are partitioned into blocks, referred to as epochs. Instead of mixing all source packets at the intermediate nodes, only packets that are from the same epoch are combined with one another. Dividing the source data stream into disjoint epochs keeps the overhead in each packet at a reasonable level.

Network coding provides two advantages over routing. First, nodes can simply broadcast coded packets to all nodes within their transmission range, without needing to know the network topology. Second, a system using network coding can provide substantial throughput gains even when nodes do not receive any feedback as to which of their neighbors received a transmitted packet. This is important because the communication protocol can be substantially simplified if feedback is unnecessary. While providing feedback is not impossible in a vehicular network, a protocol without feedback utilizes the scarce bandwidth resources more efficiently. In addition, there are contention issues on the feedback channel as well. In this dissertation, we analyze the end-to-end throughput when random linear coding is combined with the various distributed MAC protocols discussed in Section 2.2.

2.4 Related Work

The problem of distributing data from infostations to vehicles on a highway was first examined in [30, 31, 32]. This series of work focuses on a low power, low connectivity setting. Vehicles are by default disconnected from the rest of the network, and exchange packets over a short time period when they pass another vehicle in an adjacent lane. The number of packets that can be received is a function of the relative speeds of the vehicles. Such a scenario is significantly different from the situation considered here, where a high degree of connectivity is assumed at all times for all vehicles in the network.

In [33], the performance of network coding as a solution for content distribution in vehicular networks is investigated. The paper adopts a medium access protocol based on the

802.11(b) MAC, in combination with network coding. It analyzes, via computer simulations, the reduction in file download time as compared to a network layer based on Ad Hoc On Demand Distance Vector (AODV) routing, originally proposed in [34]. The impact of mobility and node density on the performance of network coding and routing is also briefly investigated, and simulation results indicate that network coding can take advantage of increasing mobility. It should be noted that these results were obtained in a scenario characterized by random car movement in an urban street grid, rather than on a highway.

Our work differs in several aspects. First, our goal is to understand the asymptotic performance limits for network coding in vehicular ad hoc networks, by providing a theoretical analysis and supporting it with computer simulations. In order to do so, we focus on an analytically tractable scenario, constituted by a highway where cars adopt a slotted Aloha MAC protocol to cooperatively distribute content provided by an infostation, where the vehicle density is relatively high.

Second, we carry out extensive throughput analysis in order to determine the capability of a peer-to-peer scheme using network coding to support real-time content streaming on highways. This analysis is paramount to the aim of designing a content distribution system for highways, since it provides an indication of the amount of infrastructure and the associated cost required for reliable communications. In this framework, we analyze the tradeoff between system complexity and performance, by comparing a spatially synchronized scheme using position information with a fully uncoordinated scheme.

Our work was initially motivated by the problem of communication using untuned radios in a dense sensor network, analyzed in [35]. In this work, a large number of small, inexpensive sensor nodes are used to relay information from a source to a destination. Because the radios in these sensors use an on-chip resonator, instead of a more expensive quartz crystal, their carrier frequencies are random. Because it is not known a priori which nodes can receive transmissions from which other nodes, it is not possible to coordinate the dissemination of information through the network.

Chapter 3

A Minimally Coordinated VANET Communication Protocol

In this chapter we propose and analyze a minimally coordinated transmission protocol for multihop content distribution. The basic idea underpinning this scheme is to partition the highway into spatial blocks, which are then used to facilitate the flow of information over many hops. Only vehicles in certain spatial blocks will attempt to transmit in specific time windows. However, the vehicles within a block will operate in an uncoordinated manner, randomly accessing the channel independently of each other. Therefore, this scheme requires some degree of coordination to set up the spatial structure and coordinate transmissions at the block level, as opposed to the purely uncoordinated protocols discussed in Chapter 2. We will refer to this protocol as a *synchronous* scheme, to emphasize the spatial synchronization imposed by the block structure. Our objective in proposing this scheme is to demonstrate the performance that can be achieved with a small degree of coordination, and to provide a benchmark against which we can compare the uncoordinated protocol examined in Chapter 5. The primary metric that we use to quantify the system performance is the end-to-end throughput between the source and destination.

The rest of this chapter is organized as follows. In Section 3.1 we discuss the block structure and the minimally coordinated MAC protocol. In Section 3.2, we analyze the throughput of this scheme with network coding. Then in Section 3.3 we discuss a modification of the basic synchronous protocol which can improve performance. Finally in Section 3.4 we give simulation results which compare the throughput of these two synchronous schemes when using both network coding and routing at the network layer.

3.1 Spatial Partitioning of the Road

The synchronous scheme is founded on dividing the highway into blocks, as shown in Figure 3.1. The length of each block is chosen to be $\frac{1}{2}R$, where R is the transmission range of the vehicles. By choosing $\frac{1}{2}R$ for the length of the blocks, it is guaranteed that under the physical layer model discussed in Chapter 2, a transmission from any vehicle in a given spatial block can be received by any vehicle in the adjacent block. Even when a vehicle at the left edge of block i is transmitting, the vehicle all the way at the right edge of block $i+1$ is within distance R of the transmitter and can decode the packet, so long as there is no interference.

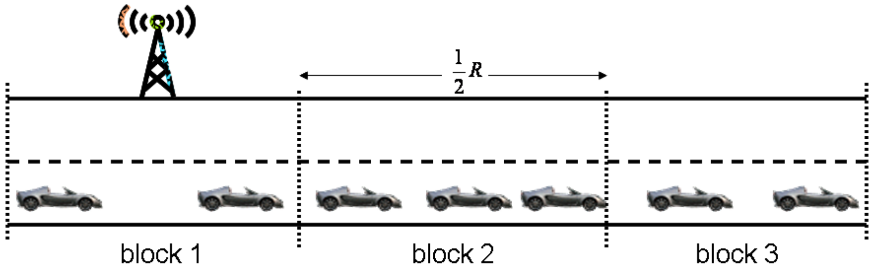

Figure 3.1: In the synchronous transmission scheme, the highway is partitioned into blocks, where the length of each block is half of the transmission range of the vehicles.

The synchronous communication protocol uses a variant of the slotted Aloha MAC protocol discussed in Section 2.2. Aloha is modified so that vehicles in the same block randomly compete with each other for the channel, but vehicles in adjacent blocks do not interfere with each other. To this end, we group the time slots into larger units, referred to as *frames*, that contain M slots each, as shown in Figure 3.2. M is a free parameter, entirely under the control of the system designer.

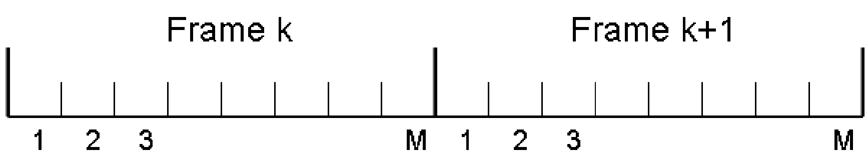

Figure 3.2: In the synchronous transmission scheme, the time axis is divided into slots, which are then grouped into frames composed of M slots.

At the macroscopic time scale of the frames, data is transmitted in a deterministic manner. In one frame of M time slots, the infostation will transmit M unique packets to the vehicles that are in block 1, i.e., the block which receives transmissions directly from the infostation. In the next frame, the vehicles in block 1 will transmit the data which they just received to the vehicles in block 2. During each subsequent frame, this data will be transmitted to the next block on the highway, similar to a “bucket brigade”. We will refer to these packets, which are forwarded from one spatial block to the next in each frame, as a *wave* of data. Figure 3.3 shows an example of the movement of waves among the blocks. The dark arrows denote that the vehicles in block 1 and block 5 are transmitting during the same frame.

The blocks which are concurrently transmitting must be separated by sufficient distance to ensure that there is no interference between these waves. It can easily be shown that the minimum separation is 4 blocks. To see that a smaller separation will not work, consider a scenario where blocks 1 and 4 are transmitting in the same frame. A vehicle in block 2 could be within distance R of both a vehicle in block 1 and a vehicle in block 4. If the vehicles in blocks 1 and 4 transmit in the same slot, this vehicle in block 2 will hear a collision.

This receiver in block 2 is then unable to decode the transmission from block 1, which contains a packet in a new wave of data, because of interference from block 4, which contains old information that block 2 has already received. This interference between waves will degrade the throughput, since vehicles in block 2 will receive fewer packets from block 1, and thus have fewer packets to forward to block 3.

On the other hand, if the waves are separated by 4 blocks, then no such interference will occur. No vehicles in block 2 are within the transmission range of any of the vehicles in block 5. Note that this does not mean that no collisions occur in the synchronous scheme. Clearly two vehicles in the same block might still transmit in the same time slot, resulting in collision. However, by separating the waves by 4 blocks, we have eliminated collisions caused by transmissions in separate waves of data.

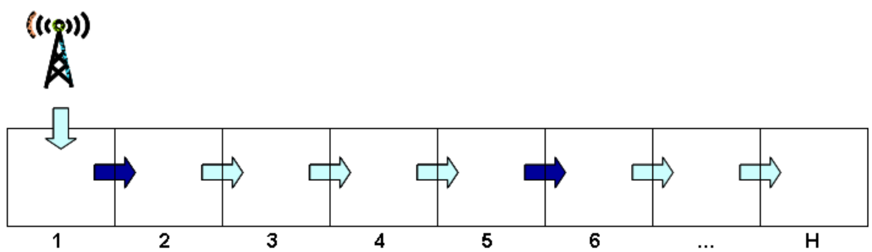

Figure 3.3: In the synchronous scheme, waves of data propagate along the highway, moving one block to the right in each frame. The dark arrows show two spatial blocks that are transmitting during the same frame. The concurrent waves are sufficiently separated so that there is no interference between them.

At the time slot scale, however, the nodes do not schedule their transmissions. As we have discussed earlier, it is not desirable, or even feasible, for centralized coordination to be implemented in this network. In the frame in which block i transmits to block $i+1$, every vehicle in block i will randomly choose exactly one slot in the frame on which to transmit, uniformly and independently of all other vehicles. During that time slot, the vehicle will either transmit one of the packets that it received from its adjacent block during the previous frame (when distributed routing is used) or a random linear combination of all of the packets it received (when network coding is used). When two or more vehicles in block i choose the same slot, a collision occurs and no vehicles in block $i+1$ can decode either packet. However, if exactly one vehicle in block i selects a given time slot, then all vehicles in block $i+1$ will successfully receive the transmitted packet.

This MAC protocol is a variation of the slotted Aloha protocol outlined in Chapter 2. The first modification is that vehicles only attempt to transmit during every fourth frame, instead of operating identically in every time slot. The second modification is that within the frame in which a given block is forwarding data, each vehicle transmits in exactly one of the M time slots in the frame, instead of making an independent random decision in every slot.

We note in passing that modifying the unslotted Aloha and carrier sense MAC protocols for use in this synchronous scheme would be much more challenging, because of the requirement that all transmissions from vehicles in the same block must be contained within a specific frame.

In order to implement the synchronous scheme, it is necessary that each vehicle knows the identity of the block in which it is located, as well as the boundaries of the time frames. Although this requires a certain level of coordination among the nodes, it is still feasible in a distributed network. If each vehicle contains a GPS receiver, the vehicles know their own locations with high precision, as well as the boundaries of the time slots. The additional information required for the synchronous scheme can be provided by having the base station periodically transmit a wave that contains control information, including the infostation location and information on the frame length and frame boundaries, instead of data. When a new vehicle enters the network, it will not transmit until it receives one such control wave. After receiving the control wave, however, the vehicle will be able to compute the location of the block boundaries, and will be able to become a full participant in the network.

3.2 Throughput Of Synchronous Protocol

We analyze the throughput provided by this synchronous protocol when the vehicles use network coding. In particular, we focus on the limit of a high vehicle density, i.e., a large number of vehicles in each spatial block. We first optimize M, the number of time slots in a frame, and then compute the end-to-end throughput.

3.2.1 Optimal Frame Length

The first question that we address is how to choose the optimal value of M, the number of time slots in a frame. If M is too small, then it will be highly probable that two or more vehicles in the same block will transmit on any single slot and most packets will be lost to collision. On the other hand, if M is large there will be few collisions, but many time slots will be empty because vehicles transmit on only one slot per frame.

We will assume that the vehicles are equidistantly spaced on the road, with exactly N vehicles in each block. Since we are primarily interested in quantifying the asymptotic throughput when the vehicle density is large, this is a reasonable assumption. However, if we instead chose to analyze the low density case, it would be more correct to assume that the vehicles were distributed randomly, e.g. according to a Poisson point process. In that case, there would be a variable number of vehicles in each block.

Given that the vehicles are spaced equally, the system designer has some control over N, because the length of each block in Figure 3.1 can be varied by adjusting the transmission range. However, due to limits on the maximum allowed transmission power in the frequency band where the system is operating, the transmission range cannot be increased arbitrarily. Thus N is dependent to some degree on the physical traffic conditions on the road.

During the frame when the vehicles in a given block are transmitting, each of the N vehicles randomly selects one of the M slots via a uniform distribution, independently of all other vehicles in the block. Our goal is to maximize the total number of packets that are successfully received by vehicles in the next block. This is equivalent to maximizing the number of time slots on which exactly one vehicle transmits. By the symmetry in the medium access protocol, this is in turn equivalent to maximizing the probability that exactly one vehicle transmits in any single time slot. We can express the probability that exactly one of the N vehicles in a block selects a fixed slot as:

$$\binom{N}{1} p(1-p)^{N-1} = N\frac{1}{M}\left(1-\frac{1}{M}\right)^{N-1}$$

where $p = 1/M$ is the probability that a given vehicle transmits during a fixed slot. We can optimize this expression by taking the derivative with respect to M and setting the result equal to zero.

$$N\left[-\frac{1}{M^2}\left(1-\frac{1}{M}\right)^{N-1} + \frac{1}{M}(N-1)\left(1-\frac{1}{M}\right)^{N-2}\frac{1}{M^2}\right] = 0$$

$$N\frac{1}{M^2}\left(1-\frac{1}{M}\right)^{N-2}\left[-\left(1-\frac{1}{M}\right)+\frac{N-1}{M}\right] = 0$$

The optimum frame length is then found by solving the equation $1 - \frac{N}{M} = 0$, which yields $M = N$. This should come as no surprise, because all of the nodes in a single block are within range of each other. It is well known that the throughput of a fully connected Aloha network is optimized when each of the N nodes transmits with probability $\frac{1}{N}$, as shown in [20].

The probability that exactly one vehicle in the block transmits on a given slot, and thus vehicles in the next block receive a packet without collision, is given by $N\frac{1}{N}\left(1-\frac{1}{N}\right)^{N-1} = \left(1-\frac{1}{N}\right)^{N-1}$. In the limit as N becomes large, this expression approaches $\frac{1}{e}$. Also, given that a vehicle transmits a packet, the probability that no other vehicles in the same block transmit on the same slot is $\left(1-\frac{1}{N}\right)^{N-1} \to \frac{1}{e}$. Each transmitted packet is successfully received with probability $\frac{1}{e}$, and lost due to collision with probability $1-\frac{1}{e}$.

3.2.2 Throughput with Network Coding

The second question that we address is the throughput of the multihop synchronous transmission scheme. The fact that data is being funneled down the highway in waves facilitates the computation of the throughput. If we can count the number of independent packets that the destination receives in a single wave, then finding the throughput is straightforward. If the destination receives an average of x independent packets per wave, then the throughput per wave is $\frac{x}{N}$, as each frame has a length of N slots. Because the waves are separated by four blocks in order to prevent interference between waves, the true throughput of the synchronous scheme will be $\frac{x}{4N}$. This throughput is measure in the abstract unit of "packets per time slot". Multiplying by the packet length and dividing by the slot duration will produce a throughput in the standard unit of bits per second.

To find the average number of independent packets that the destination receives in each wave, we formulate a random graph which models the propagation of a wave of data through the network, as shown in Figure 3.4. In this graph, the N nodes in each column represent the N vehicles in each block on the highway[1]. The H columns represent H blocks between the base station and the destination vehicle. Each node is connected to all of the nodes in the previous column with a unit capacity link, since a vehicle could potentially receive one packet from every vehicle in the previous block, in the case where all of those vehicles chose unique slots on which to transmit.

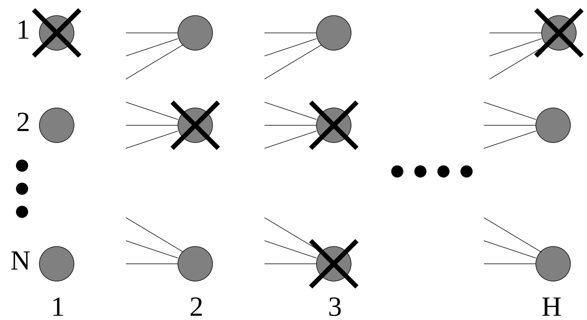

Figure 3.4: A random graph modeling the propagation of a wave of data through a vehicular ad hoc network using the synchronous transmission scheme. Each node, and all of its incoming and outgoing edges, is independently deleted with probability $1 - 1/e$.

Because a transmitted packet is corrupted by collision with probability $1 - 1/e$, each of the vertices in the random graph is independently deleted with probability $1 - 1/e$, in which case all of its incoming and outgoing links are also deleted[2]. The source of an edge that is not deleted is uniformly distributed over the non-deleted vertices in the previous column, because given that exactly one vehicle transmits in a slot, the source of that transmission is equally likely to be any of the nodes that do not experience a collision. We will label this graph $G_{1-1/e}$, to denote that the vertices are deleted with probability $1 - 1/e$.

The maximum flow is the largest number of packets that can be transported from the source to the destination of the graph under any protocol, while respecting the link capacities. In our synchronous scheme, each vehicle transmits a random linear combination of all of the packets that it received from the previous block. However, with random network coding it is known that the maximum flow is achievable with high probability, as long as the coding is done over a sufficiently large finite field [29]. The challenge in this problem is to compute the maximum flow through the random graph in Figure 3.4.

Because all vehicles in the first block deterministically receive the M packets sent by the infostation, that link is not the limiting bottleneck on the end-to-end flow. In addition, we

[1]The synchronous scheme can also accommodate situations where the number of vehicles in each block is variable. There will be a small performance degradation, due to the fact that the fixed frame length can not be optimal for every block.

[2]Here, we are making the approximation that each transmission is lost to collision independently, in order to make the computation of the throughput tractable. This is valid in the limit of large N.

recall that in the synchronous scheme, all vehicles in the same block receive the same packets. In particular, the destination receives the same set of packets as all of the vehicles in the last column. Therefore, the maximum flow between the source and destination is given by the number of vertex-disjoint paths between the first and last columns of the random graph in Figure 3.4. The following theorem quantifies the achievable flows, in the limit of high density. The proof will be given after we prove two associated lemmas.

Theorem 1: *For any constant β such that $\beta < 1/e$, the maximum flow through $G_{1-1/e}$ is greater than βN with high probability as N goes to infinity, as long as H is subexponential in N.*

As we will see, the probability that this theorem does not hold decays to zero exponentially with N. We prove Theorem 1 via two lemmas. We use a modified version of a common technique in percolation theory, which relates the probability of having many disjoint end-to-end paths through a lattice to the probability of having a single end-to-end path. We will use A to denote the event that there exists at least one end-to-end path from the first column to the last column of the random graph. We will let A_r denote the event that the random graph has the property that the deletion of *any* r vertices results in a graph belonging to A. This is equivalent to saying that any graph in A_r has at least $r+1$ vertex-disjoint paths connecting the first and last columns. As an example, Figure 3.5 shows an instantiation of the random graph, where the black circles represent nodes that are present and the white circles represent nodes that have been deleted. This graph has only one vertex-disjoint end-to-end path, and an example path is shown. Figure 3.6, in contrast, shows another instantiation of the random graph, where the nodes are deleted with a lower probability. This graph has $r = 3$ vertex-disjoint end-to-end paths. The following lemma relates the events A and A_r to the probability of deleting vertices in the two graphs.

Lemma 1: Let r be a positive integer. Then

$$P(G_{p_2} \notin A_r) \leq \left(\frac{q_2}{q_2 - q_1}\right)^r P(G_{p_1} \notin A)$$

for each $0 \leq p_2 \leq p_1 \leq 1$. Here, $q_1 = 1 - p_1$ and $q_2 = 1 - p_2$, and G_p is the random graph where each vertex is independently deleted with probability p.

Proof of Lemma 1: The proof of this lemma is based on the proof in [36] of a related result in percolation theory, and the adaptation of that proof to the problem of communicating using untuned radios in [35]. Let $X_{i,j}$ $\forall i \in \{1, 2, ...N\}$ and $j \in \{1, 2, ...H\}$ be i.i.d. random variables, uniformly distributed in the interval $[0, 1]$. We assign the variable $X_{i,j}$ to the vertex in row i and column j of a rectangular grid. We then create two random graphs G_{p_1} and G_{p_2}, which have vertices deleted with probability p_1 and p_2 respectively, as follows. We first insert edges from every vertex in columns 2 through H to all vertices in the previous column. Then we generate G_{p_1} by deleting the vertex in row i, column j (and all incoming and outgoing edges) if and only if $X_{i,j} \leq p_1$. G_{p_2} is likewise generated by deleting each vertex if and only if $X_{i,j} \leq p_2$.

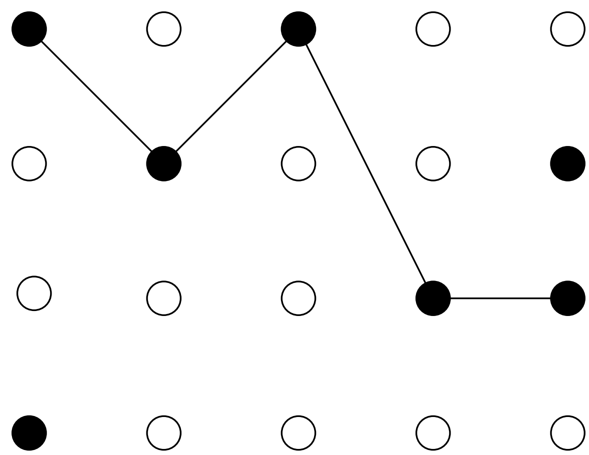

Figure 3.5: An instantiation of the random graph modeling the propagation of a single wave through the network. The white circles represent nodes that were deleted, and the black circles represent nodes that were not deleted. This graph has only a single vertex-disjoint end-to-end path. One such path is outlined.

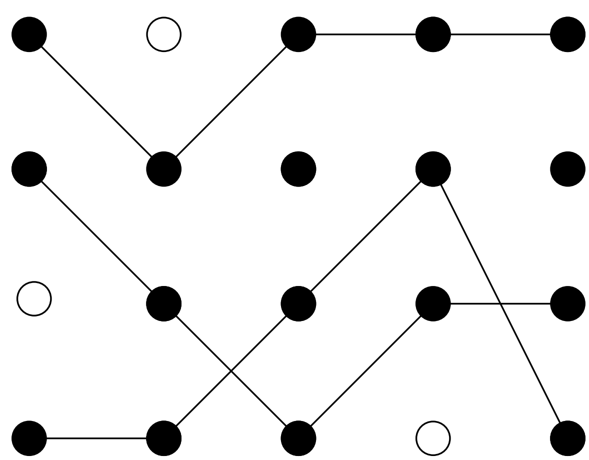

Figure 3.6: An instantiation of the random graph modeling the propagation of a single wave through the network. The white circles represent nodes that were deleted, and the black circles represent nodes that were not deleted. This graph has $r = 3$ vertex-disjoint end-to-end paths. One set of three disjoint paths is outlined.

Our goal is to find a relationship between the probability that $G_{p_2} \in A_r$ and the probability that $G_{p_1} \in A$. We observe that if $G_{p_2} \notin A_r$, then there must exist a set of vertices $\mathcal{T}$ such that:

- None of the vertices in $\mathcal{T}$ are deleted in G_{p_2}
- $|\mathcal{T}| < r$
- The graph $G^*_{p_2}$ obtained by deleting the vertices in $\mathcal{T}$ from G_{p_2} satisfies $G^*_{p_2} \notin A$.

There may exist many such sets $\mathcal{T}$. For our purposes, it is sufficient to consider any one set with this property. Suppose that $G_{p_2} \notin A_r$, and that every vertex i, j in the set $\mathcal{T}$ satisfies $p_2 < X_{i,j} < p_1$. It then follows from the third bullet point above that $G_{p_1} \notin A$. After conditioning on a set $\mathcal{T}$, the probability that $p_2 < X_{i,j} < p_1$ for all vertices in $\mathcal{T}$ is given by

$$[(p_1 - p_2)/(1 - p_2)]^{|\mathcal{T}|} = [(q_2 - q_1)/q_2]^{|\mathcal{T}|}$$

Therefore

$$P(G_{p_1} \notin A | G_{p_2} \notin A_r) \geq \left(\frac{q_2 - q_1}{q_2} \right)^r$$

Applying the definition of conditional probability shows that

$$P(G_{p_2} \notin A_r) \leq \left(\frac{q_2}{q_2 - q_1} \right)^r P(G_{p_1} \notin A \cap G_{p_2} \notin A_r)$$

Finally, the proof of Lemma 1 is completed by using the fact that

$$P(G_{p_1} \notin A \cap G_{p_2} \notin A_r) \leq P(G_{p_1} \notin A) \qquad \square$$

Lemma 1 can be used to prove Theorem 1 if we can show that the probability that $G_{p_1} \notin A$ decays exponentially to zero with N, for some p_1. In other words, if we can show that $P(G_{p_1} \notin A) \leq e^{-\alpha N}$, where α is a constant that depends on p_1, then we can rewrite Lemma 1 as

$$P(G_{p_2} \notin A_r) \leq \left(\frac{q_2}{q_2 - q_1} \right)^r e^{-\alpha N} = e^{r \ln\left(\frac{q_2}{q_2 - q_1}\right) - \alpha N}$$

and by choosing $r = \beta \cdot N$, it follows that the probability of not having $(\beta N + 1)$ end-to-end edge disjoint paths is bounded by

$$P(G_{p_2} \notin A_r) \leq e^{-\left(\alpha - \beta \ln\left(\frac{q_2}{q_2 - q_1}\right)\right) N}$$

If β satisfies the condition

$$\beta < \alpha / \ln\left(\frac{q_2}{q_2 - q_1} \right)$$

then the probability of not having βN end-to-end paths decays to zero exponentially. An exponentially tight bound on $P(G_{p_1} \notin A)$ does in fact exist, as shown by Lemma 2.

Lemma 2: The probability that there are no end-to-end paths in G_{p_1} is bounded by

$$P(G_{p_1} \notin A) \leq H e^{N \log p_1}$$

Proof of Lemma 2: Observe that if all of the nodes in a single column of the graph are deleted, then $G_{p_1} \notin A$. On the other hand, if at least one node in every column is not deleted, then an end-to-end path exists, because every node in the graph is connected to all N nodes in the previous column. The probability that all nodes in a single column are deleted is p_1^N, and, by the union bound, the probability that all N nodes in one or more columns are deleted is less than or equal to

$$Hp_1^N = He^{N \log p_1} \qquad \square$$

Proof of Theorem 1: We can rewrite the probability that $G_{p_1} \notin A$ as $e^{-N(-\log p_1 - (\log H)/N)}$. Combining Lemma 1 and Lemma 2, we see that the probability that there are not at least βN paths through G_{p_2} decays exponentially if

$$\beta < \frac{-\log p_1 - (\log H)/N}{\log\left(\frac{1-p_2}{p_1-p_2}\right)}$$

If H is constant, or growing subexponentially in N, then the $(\log H)/N$ term will go to zero in the limit of large N, and the expression becomes

$$\beta < \frac{-\log p_1}{\log\left(\frac{1-p_2}{p_1-p_2}\right)}$$

Remember that in the original vehicular network $p_2 = 1 - 1/e$, the probability that packets are lost to collision when the frame length M is chosen to be equal to N. However, p_1 is unrelated to the graph G_{p_2}, so we are free to choose any value of p_1, as long as it is larger than p_2. We evaluate the bound on β above by setting $p_2 = 1 - 1/e$ and choosing $p_1 = 1 - \epsilon$, and then taking the limit as ϵ goes to 0.

$$\begin{aligned} \frac{-\log(1-\epsilon)}{\log\left(\frac{1/e}{1-\epsilon-1+1/e}\right)} &= \frac{-\log(1-\epsilon)}{\log(1/e) - \log(1/e-\epsilon)} \\ &= \frac{\log(1-\epsilon)}{1 + \log(1/e - \epsilon)} \end{aligned}$$

Applying L'Hopital's rule, we find that as ϵ approaches 0, this bound approaches $1/e$. This proves that, with high probability, the throughput of the random graph is at least N/e. $\square$

Theorem 1 demonstrates that the maximum flow through the random graph is at least N/e. We also observe that the maximum flow, on average, cannot exceed N/e. This can be easily seen by looking at the first two blocks. With each of the N nodes in block 1 randomly choosing one of the N time slots, the vehicles in block 2 will receive, on average, N/e packets. Furthermore, nodes in subsequent blocks cannot receive more packets than the nodes in block 2. Thus, we can make the stronger statement that the maximum flow through the graph is equal to N/e.

The implication is that a fraction $(1 - 1/e)$ of the packets are lost between the first two blocks, but with high probability no more information is lost in any subsequent hops (as long as H does not grow exponentially in N). Each block in the system can be viewed as a digital repeater, which preserves all information that it received from the preceding block.

As discussed earlier, because the maximum flow for a single wave is N/e packets, and network coding can achieve the maximum flow with high probability, the throughput of the synchronous scheme is equal to $1/(4e) \approx 0.09$ packets per time slot. Thus, the destination vehicle receives a novel packet in approximately 9 percent of the time slots.

Finally, it is instructive to compare this synchronous scheme with a fully coordinated network. If complete coordination is possible, then each node in a block can be assigned a unique time slot within the frame, so that no collisions occur. In this case, the destination receives N independent packets per wave, and the corresponding throughput is 1/4 packets per time slot. Hence, the "cost of uncoordination" in the synchronous scheme is a factor of $1/e$. This protocol achieves a throughput that is about 36% of what could be achieved with a fully coordinated block synchronized protocol.

Despite this loss, this scheme still enables us to transmit the same throughput over a large distance, as long as the number of hops does not grow exponentially in the density. Like a fully coordinated protocol, the synchronous scheme has the attractive property of not losing information as the number of hops grows, even though it involves random channel access.

3.3 Synchronous Transmission with Overhearing

While the block structure described in Section 3.1 is very helpful in moving waves of data along the highway in an analytically tractable manner, it has a shortcoming in that it artificially limits the number of packets that each vehicle receives. As one example, because the length of each block is $R/2$, when a vehicle in block 1 in Figure 3.1 transmits, some vehicles in block 3 will be able to successfully receive the packet. A second example is that if two vehicles in block 1 transmit in the same time slot, all vehicles in block 2 will hear a collision, but some vehicles in block 3 will be able to receive one of the packets, by virtue of being within range of one of the transmitters but not the other. Likewise, if vehicles listen to transmissions from other vehicles within their own block, they might also receive new information[3].

Taken together, these examples raise the possibility that a vehicle could improve the number of independent packets that it receives in each wave of data by decoding the transmissions from all nodes within its range, instead of just transmissions from vehicles in the

[3]In the basic synchronous scheme, all vehicles in one block receive the same set of packets from the previous block, and thus decoding transmissions from neighbors in the same block provides no additional information. However, when vehicles are allowed to receive packets transmitted from non-adjacent blocks, it is no longer true that all vehicles in a block must have the same information. In that case, listening to transmission from a vehicle's own block may be beneficial.

previous block. While many of these additional received packets will clearly provide redundant information, some of them could be independent of the packets received from the previous block. We use the term "overhearing" to describe this extension of the basic synchronous scheme. In general, there is no reason not to receive and decode any transmission not corrupted by collision, since battery life is not a constraint in the vehicular networking application. The inclusion of overhearing complicates the analysis, and thus a rigorous bound similar to Theorem 1 for the basic synchronous scheme is beyond the scope of this work. However, in the next section we provide simulations that demonstrate the gains that are achievable when overhearing is included.

3.4 Simulation Results

The synchronous transmission scheme was simulated in the OMNeT framework [37], under the following scenario. We considered a highway strip of length D, populated by N_{cars} evenly spaced cars. The highway was served by an infostation, located along the strip in position $x_{IS} = R/4$, where R is the transmission range, assumed to be identical for the infostation and the vehicles.

The transmission scheme used in the simulations can be summarized as follows.

1. The cars on the highway strip are partitioned into blocks of length $R/2$.
2. Time is divided into slots of duration T_s, which are in turn organized into frames of N slots, leading to a frame duration $T_f = N \cdot T_s$. The duration of a slot is selected as the time required to send a packet of length L_{packet} at a bit rate W.
3. Both the infostation and each block of cars transmit in one out of every four frames.
4. During frames in which transmission for a block is allowed, every car in the block transmits in each slot with probability $p = \frac{1}{N}$.

We simulated N_{waves} waves of information and averaged the measured throughput, corresponding to a total simulation time of $T_{sim} = 4 \cdot T_f \cdot N_{waves}$.

The values used for the various parameters of the network and the synchronous protocol are given in Table 3.1. We can make several observations about the scenario that is simulated here. First, the duration of the simulation time T_{sim}, although sufficient to analyze the behavior of the system over several waves, is short enough to allow us to neglect the displacement of vehicles with respect to the infostation caused by their movement at speed v_{cars}. This justifies the assumption of a static scenario adopted in our analysis. Second, the vehicle density is fairly high, corresponding to a traffic jam scenario.

The above scheme was used to compare the performance of four different strategies, which are given by combining the two network layer protocols and the two variations on when vehicles listen to the channel:

- Routing with the basic synchronous protocol
- Routing with overhearing

Table 3.1: The system parameters used to simulate the performance of the synchronous transmission scheme

Parameter	**Value**
D	8000 m
N_{cars}	2400
R	800 m
T_s	1 ms
L_{packet}	976 bit
W	1 Mb/s
N	120
T_f	0.12 s
p	0.0083
N_{waves}	20
T_{sim}	9.6 s

- Network coding with the basic synchronous protocol
- Network coding with overhearing

For each strategy we measured the system rank received per wave, averaged over N_{waves} waves. System rank was defined as the number of unique packets in the case of routing, and as the number of linearly independent packets in the case of network coding.

Figure 3.7 confirms the analysis in Section 3.2. This plot shows the system rank as a function of the distance of the car from the infostation. For clarity, only the rank for the car at the center of each block is shown. These simulation results demonstrate that the use of network coding in place of routing significantly increases performance. More importantly, the simulation confirms that network coding leads to a different scaling law than routing. Even in the case of overhearing, in fact, the rank clearly decreases with distance under routing, eventually decaying to zero. On the other hand, network coding is able to support a high rank over large distances, as was proven analytically, and thus provides an appealing solution for streaming data delivery between roadside infostations. Also note that the introduction of overhearing is seen to improve throughput by about a factor of 1.75.

As an aside, we also verify via simulation that the increase in performance with network coding is independent of the number of delivered packets. Figure 3.8 shows the number of packets which are successfully received per wave, regardless of whether or not those packets are independent. Because we selected the duration of the frame to maximize the number of correct receptions, the number of received packets per wave is very close to the expected value of N/e in the case of the basic synchronous scheme, without overhearing. The important point is that the same number of raw packets is received with both routing and network coding. With overhearing, the total number of received packets per wave increases to $4N/e$. However, many more of these packets will be dependent, and thus the throughput does not increase by a factor of four with overhearing.

Finally, we also empirically studied the delay introduced by synchronous strategies. Figure 3.9 shows the time when the infostation begins transmitting the first packet of each wave, as well as the time when the destination node receives the last packet of that wave. The results are shown only for network coding with overhearing. In this synchronous scheme, the delay with routing would be the same, although the destination would receive fewer independent packets in each wave.

The figure verifies that the delay experienced between one wave and the next is constant, and equal to the delay introduced by the infostation, which transmits a new wave in every fourth frame. The implication is that the synchronous scheme is suitable for delivering delay-sensitive traffic, such as streaming audio and video. It is also evident from Figure 3.9 that synchronous strategies introduce a significant latency before the first wave reaches the destination, due to the pipeline-like nature of the scheme. It can be readily shown that the delay for each wave is about $(N_{blocks} + 2) \cdot T_f$ seconds. For the settings used in these simulations, this corresponds exactly to the $22 \cdot 0.12 = 2.64$ seconds shown in Figure 3.9. The conclusion is that the synchronous scheme may not be suitable for use in interactive applications, where latency must be minimized.

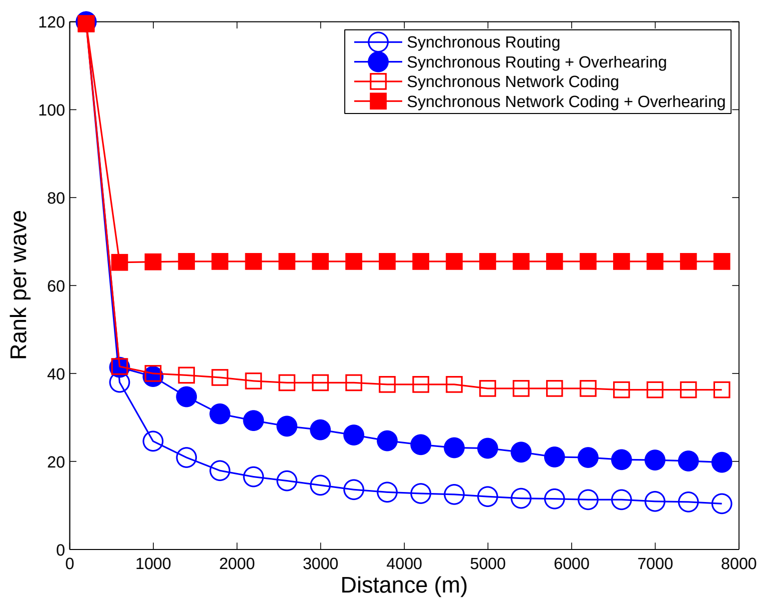

Figure 3.7: The number of independent packets received per wave as a function of the location of the destination vehicle in the synchronous transmission scheme. Results are shown for four different strategies: routing with the basic synchronous protocol, routing with overhearing, network coding with the basic synchronous protocol, and network coding with overhearing.

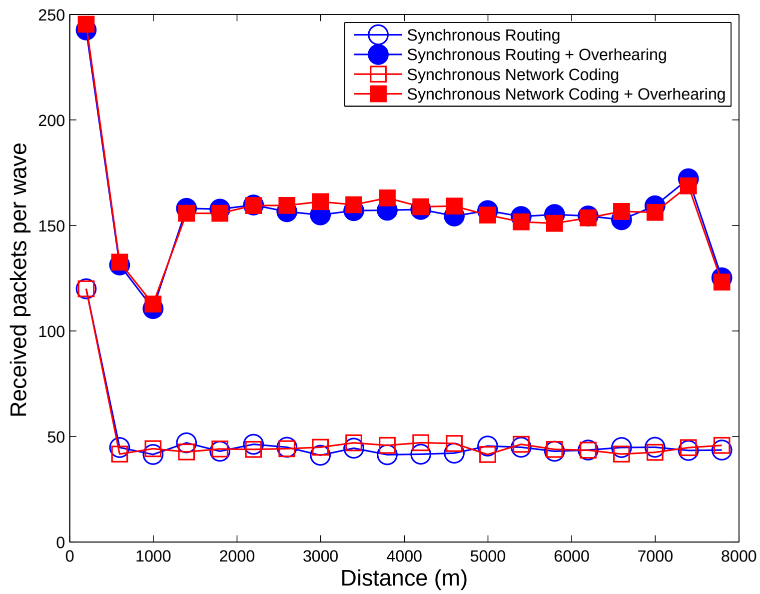

Figure 3.8: The total number of received packets per wave as a function of the location of the destination vehicle in the synchronous transmission scheme. This plot counts all packets received without collision, regardless of whether they are independent. Results are shown for four different strategies: routing with the basic synchronous protocol, routing with overhearing, network coding with the basic synchronous protocol, and network coding with overhearing.

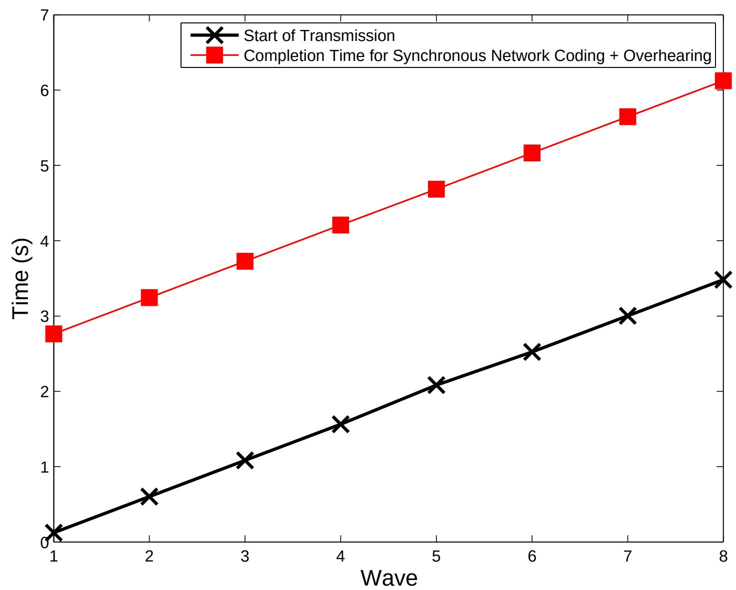

Figure 3.9: The time when the infostation begins transmitting each wave, and the time when the destination vehicle receives the final packet of the wave, for the scenario outlined in Table 3.1. The vertical distance between the two curves is the time delay of the synchronous scheme.

Chapter 4

Uncoordinated Wireless Networks

We now examine the throughput of fully uncoordinated networks, which do not require the minimal level of coordination necessary for the synchronous scheme considered in Chapter 3. In this chapter, we will give a general method for computing the throughput of a wireless network with arbitrary topology that utilizes a randomized MAC protocol and random linear network coding. Then in Chapter 5 we will apply this algorithm to the topology of a vehicular ad hoc network and compute the throughput of an uncoordinated VANET.

In this chapter, we consider an even stronger restriction than our definition of uncoordinated networks in Chapter 2. Previously, we had simply disallowed any centralized scheduling of transmissions or routing of packets. By this definition, an uncoordinated network could still incorporate link level feedback and could potentially build routing tables via local interaction, although for simplicity in Chapter 3 we designed a scheme that did not require feedback.

In this chapter, in contrast, we consider the throughput of *neighbor-oblivious* wireless networks. This refers to networks where nodes do not know the identities of any of their neighbors. Thus, they cannot address neighboring nodes, and sending feedback about successfully received packets is not possible. This is a stricter limitation, which forces the network to use distributed MAC and network layer protocols. Clearly, a protocol which satisfies the requirements of a neighbor-oblivious network can be used in an uncoordinated VANET.

The rest of this chapter is organized as follows. In Section 4.1 we discuss the relationship between this neighbor-oblivious problem and prior work on scheduling in wireless networks. In Section 4.2 we review the protocols that we use, and discuss the primary challenge in calculating the multihop throughput. Then in Section 4.3 we present a method for computing the capacity of an uncoordinated network with arbitrary topology.

4.1 Neighbor-Oblivious Networks

One of the most challenging problems in wireless networking has proven to be scheduling: the question of how the nodes should share the limited wireless resource in order to maximize performance. A throughput-optimal centralized scheme, known as maximal-weight scheduling, was first presented in [38]. Jiang and Walrand made a major contribution to the field by developing a distributed algorithm that uses carrier sensing to attain the optimal throughput [22]. A similar distributed algorithm has been proposed for wavelength assignment in optical networks [39]. Subsequent work has extended [22] to the discrete-time setting [24, 23], characterized the delay [23], and implemented the algorithm in practice [40].

However, these schemes all make the implicit assumption that once the resource allocation problem is solved, packets can be routed from the source to the destination, either via a global routing table or by implementing a distributed routing algorithm. Nodes must know the identities of their neighbors in order to forward packets to the next hop in the desired path. In this chapter, we consider multihop communication in *neighbor-oblivious* wireless networks. When nodes do not know who their neighbors are, they cannot address packets to specific neighbors and thus it is not possible to route packets. One obvious, though limited, approach is to use a simple packet flooding scheme. We present a more complex scheme that combines the random medium access control protocols outlined in Chapter 2 with network coding to achieve a higher throughput. Our focus will be on providing a method to compute the end-to-end capacity.

As in Chapter 2, we consider a unicast communication problem. We study the end-to-end throughput in a multihop network with arbitrary topology, as shown in Figure 4.1. Furthermore, we require that the nodes must be neighbor-oblivious, i.e., that they do not know the identities of neighboring nodes and thus cannot route packets. Instead, when nodes transmit they must broadcast packets to all of their neighbors.

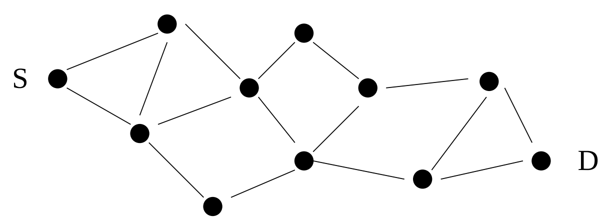

Figure 4.1: We consider a unicast communication problem in a multihop neighbor-oblivious network. Our objective is to find the maximum throughput under the limitation that none of the nodes know the identities of their neighbors.

4.2 A Distributed Communication Protocol for Neighbor-Oblivious Networks

Due to the constraint that nodes do not know the identity of their neighbors, all nodes must use one of the distributed MAC protocols discussed in Chapter 2 and a distributed network layer protocol. In this chapter, we analyze the throughput under the combination of random network coding and a randomized MAC.

The primary challenge when calculating the multihop throughput is to account for the possibility that when an intermediate node accesses the channel, it might not have any new information to send to its neighbors. Even if the transmitted packet is received without collision, it does not convey any useful information to the receivers and the time slot is effectively wasted. This problem can be best illustrated through a concrete example. Consider the network in Figure 4.2, where a source S sends packets to a destination D through a single relay node R. Further, imagine that due to the random nature of the medium access, the source and relay nodes happen to access the channel in four consecutive time slots in the order S, R, R, S.

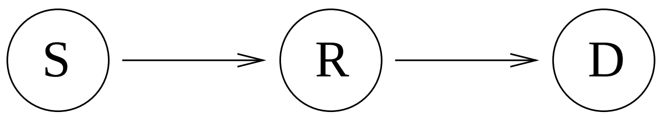

Figure 4.2: This simple network illustrates the starvation problem that can occur when nodes access the channel in a random order. Consider a case when the source and relay transmit in the order S, R, R. When the relay transmits the second time, it has not yet received a second packet from the source and hence can only send a redundant packet to D.

Even though S and R both transmitted two packets, D has only received one independent packet. This is because when R transmits the second time, it has not yet received another packet from S, and hence must retransmit the same packet that it sent previously. Although both links may support the same average rate, the relay temporarily did not have any new packets to send to the destination. We refer to this as the *starvation* problem to emphasize that the relay has "consumed" all of its packets (forwarded them to the destination) without being "fed" (received new packets from the source).

We emphasize that this is not a problem that can be solved by coding. Even if R is using network coding, when it transmits the second time it can only send a multiple of the first packet, which does not provide an independent packet to D. This starvation problem is caused by the random ordering of when links are used, and it can occur regardless of whether the network uses routing or coding at the network layer. The source of the problem is the fact that the randomized MAC induces a random ordering on transmissions, as opposed to a deterministic ordering which could be implemented with centralized scheduling.

In wired networks, starvation is not an issue because all of the links in the network can be used concurrently. In a wired network with the same topology as Figure 4.2, both S and R can transmit packets in every time slot. Hence, R always has a new packet, which it received in the previous slot, to send to D. Starvation is also not an issue in the spatially synchronized protocol studied in Chapter 3. Because of the manner in which waves propagated from block to block down the road in the synchronous protocol, each vehicle only transmitted a single packet in each wave. When this transmission occurs, the vehicle always has received some set of packets from the previous block, and hence transmits one linear combination of those packets. However, in more general networks where a node can transmit multiple packets containing combinations of the same data, we must account for starvation.

Much of the existing work on designing and analyzing random distributed scheduling algorithms makes the assumption that nodes are in a saturated state, i.e., that they always have a useful packet to send when they access the channel [22, 24]. This is equivalent to assuming that the starvation effect can be ignored. In the course of analyzing uncoordinated networks, we show that while starvation does lead to some transmissions being wasted, the average end-to-end throughput is not decreased. In essence, the loss due to starvation is a sub-linear effect. Our results in this chapter validate the standard saturation assumption, and thus are useful in a wide class of random scheduling problems beyond the neighbor-oblivious networks considered here.

4.3 Capacity of Neighbor-Oblivious Networks

We now present an algorithm for computing the end-to-end throughput of a wireless network with arbitrary topology that uses a randomized MAC protocol and random network coding. The method can be summarized as follows. We will discuss the various steps in more detail in the following sections.

1. Enumerate the *independent sets*, the sets of links in the network that can be used concurrently.
2. Compute the capacities of the individual links from a Markov chain on the independent sets.
3. Construct an equivalent wired network, with the same topology and link capacities as the wireless network.
4. Find the minimum cut of the equivalent wired network.

The minimum cut of the equivalent wired network is clearly an upper bound on the throughput of the wireless network, since the nodes in a wired network are always saturated due to the fact that all links can be used concurrently. We prove that in fact this minimum cut can also be achieved in the wireless network. The key tool is the construction of a queueing network which models the flow of equations in the wireless network. We use queueing theory to show that whenever the source in the wireless network transmits data at a rate smaller than the min-cut of the equivalent wired network, the destination receives independent packets at the same rate. Therefore, the maximum end-to-end throughput of the wireless network is the same as in the wired network, despite the fact that the starvation problem may affect some transmissions.

4.3.1 Computation of Link Capacities

Our algorithm incorporates a recently proposed method for finding the capacities of individual links in CSMA networks. The throughput of links was shown to be computable from a time-reversible Markov chain in [41]. An efficiently computable approximation was given in [42] and the approach has been further developed in [43].

As a concrete example, we will consider the linear network shown in Figure 4.3, with links labeled 1 through 4. The first step is to list the sets of links that can be used concurrently without interfering with each other, which are known as independent sets. Trivially, any set containing only one of the links is an independent set, because it is always possible to use only one link at a time. In addition, we note that links 1 and 4 can be used at the same time without interfering, and thus $\{1, 4\}$ is an independent set. However, no other set of multiple links can be used concurrently. For example, imagine that node S attempts to transmit a packet on link 1 and node B attempts to transmit a packet on link 3 at the same time. Because A is within transmission range of both S and B, a collision occurs and the packet on link 1 cannot be received. Thus, there are five independent sets in this example: $\{1\}$, $\{2\}$, $\{3\}$, $\{4\}$, and $\{1, 4\}$.

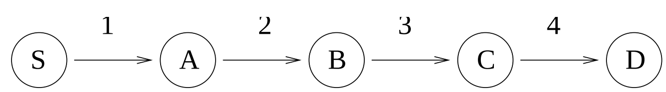

Figure 4.3: A linear wireless network with four links. The independent sets, which are sets of links that can be used concurrently without interfering with each other, are $\{1\}$, $\{2\}$, $\{3\}$, $\{4\}$, and $\{1, 4\}$.

The second step is to form a Markov chain whose current state is the set of links that are carrying information at the current time. We refer to this set as the *active* links. Clearly, the state space of the Markov chain is given by the independent sets plus the null set. This Markov analysis is valid for networks using Aloha and CSMA medium access protocols without hidden nodes[1].

The valid transitions of this Markov chain, and the transition probabilities, depend on the specific MAC protocol being used. For slotted Aloha, the Markov chain is a fully connected graph, since the set of active links in two different time slots is independent. The network can transition from any independent set to any other independent set in a single time step. With CSMA, on the other hand, time dependencies are introduced because of the backoff counter that is started when a node senses that the channel is in use. The transition diagram for the network in Figure 4.3 will have a form like that shown in Figure 4.4.

Next, we find the steady state distribution of this Markov chain. Whenever there is a finite number of links in the network, there will be a finite number of independent sets. The Markov chain then has a steady state distribution, which can be computed from the transition probabilities. Finally, the link capacities are given by the fraction of time when a link is active, which is the sum of the steady state probabilities of independent sets containing that link. In this example, C_1, the capacity of link 1, is equal to $\pi_1 + \pi_{14}$.

We can then create an equivalent wired network, which has the same topology and link capacities as the wireless network. Figure 4.5 shows this equivalent wired network for the example in Figure 4.3. We will also refer to this graph as the *link capacity graph*.

[1]As discussed in Chapter 2, any CSMA network can be made free of hidden nodes by appropriately adjusting the carrier sensing range.

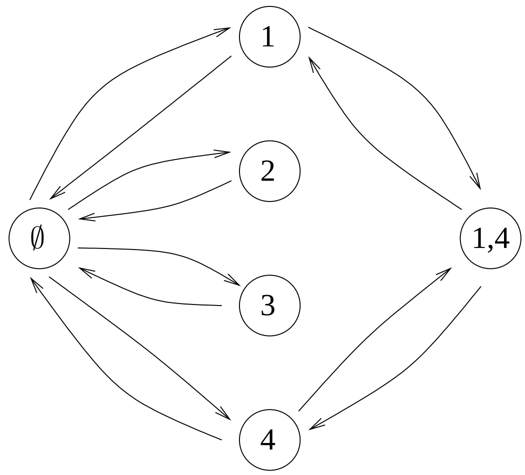

Figure 4.4: The set of links that are currently active can be modeled with a Markov chain defined on the independent sets. This figure shows the Markov chain corresponding to the network in Figure 4.3 when using a carrier sensing MAC protocol.

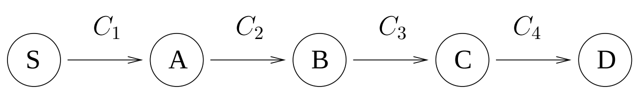

Figure 4.5: The wireless network in Figure 4.3 can be replaced with an equivalent wired network, know as a link capacity graph, whose individual links have the same capacities as the wireless links.

A simple application of the max-flow min-cut theorem shows that the capacity of the equivalent wired network is equal to the minimum cut of this link capacity graph. Furthermore, as stated in Chapter 3, random network coding can achieve this min-cut if the coding is done over a large enough field. However, it is not obvious that the capacity of the wireless network is the same as the equivalent wired network, due to the possibility of starvation at the intermediate nodes.

4.3.2 Virtual Relay Nodes

Before addressing the relationship between the wireless and wired networks, we take a brief detour to introduce a more general example network. In the simple linear network in Figure 4.3, each transmitted packet is received by only a single receiver. This masks the broadcast nature of the wireless channel, where a single transmission could be received by multiple nodes. We now consider the example shown in Figure 4.6. When S transmits, depending on which of the extraneous interferers are simultaneously transmitting, the packet could be received by A only, by B only, or by both A and B. However, in the last case, A and B receive the *same* packet. It is not possible for S to send independent packets to A and B in a single time slot, even though there are links to both A and B.

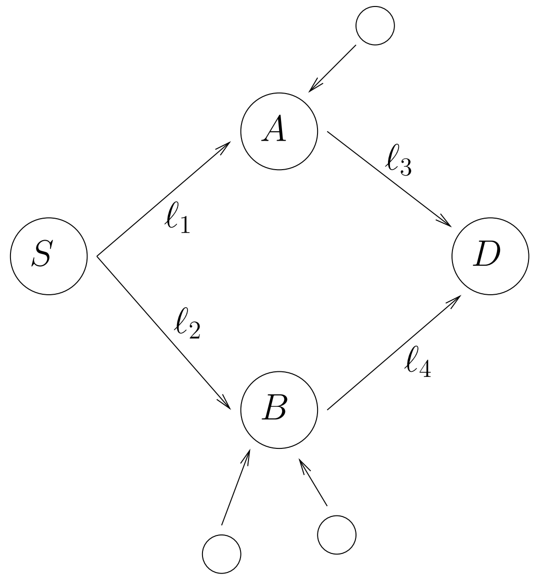

Figure 4.6: A diamond network with source S, destination D, and two relays A and B. Transmissions from S may be received by either or both of the relay nodes.

To account for this effect, we introduce a virtual node AB, which represents packets that are received by both of the relay nodes. The link capacity graph for this example is shown in Figure 4.7. Note that there are infinite capacity links from the virtual node to both of the relays, since packets sent to the virtual node are physically received by both A and B.

In the remainder of this section, we will use this example, since it brings out the general structure that will be needed when we study vehicular networks in Chapter 5

4.3.3 Achievability of Min-Cut of Equivalent Wired Network

Our main result is that the end-to-end capacity of the wireless network is equal to the minimum cut of the equivalent wired network, even when intermediate nodes experience starvation. A natural tool to track the propagation of information through a multihop network is queueing theory. However, there is an immediate complication in neighbor-oblivious wireless networks, where nodes are required to broadcast their transmissions. If a transmitted packet is received by two or more nodes, all of the receivers will store a copy of it in their buffer. In queueing theory, in contrast, it is generally assumed that when a customer is processed by one server, it is then forwarded to the queue of only one other server.

In order to apply queueing theoretic results, we first introduce a genie into the system which allows only one receiver to store each transmitted packet, even if multiple nodes were able to receive the packet. We prove that this genie can only decrease the end-to-end throughput of the network. Therefore, analyzing the system with the genie provides a lower bound on the performance of the true system. Finally, we show that there is one genie such that the end-to-end throughput is given by the minimum cut of the equivalent wired network.

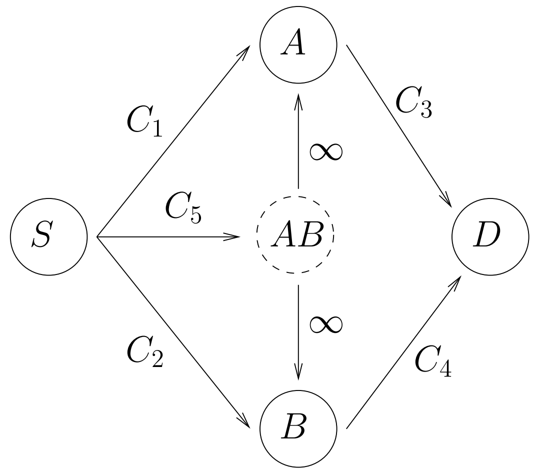

Figure 4.7: The link capacity graph of the network in Figure 4.6. The dashed node represents a virtual node, which accounts for packets that are received by both of the relays.

Genie-Degraded Wireless Network

We first find the maximum flow through the equivalent wired network, and one rate allocation that achieves this flow. We denote the rate assigned to each link by R_i^*, and the value of the flow by λ^*. By the max-flow min-cut theorem, λ^* is equal to the min-cut of the graph. Clearly, if the underlying physical network was a wired network with the same capacities, then λ^* would be the end-to-end capacity.

As an example, Figure 4.8 shows this *rate allocation graph* corresponding to the link capacity graph in Figure 4.7. Because this is a valid flow, $R_i^* < C_i$ for all i, and the total rate entering each intermediate node equals the total rate exiting. We further note that for any $\lambda < \lambda^*$, we can write $\lambda = \gamma\lambda^*$, where $0 < \gamma < 1$. Then, $R_i = \gamma R_i^*$ is a valid flow that achieves rate λ.

To find the end-to-end capacity of the wireless network, we would like to construct a network of queues with the same topology as the wireless network. However, as noted above, in a neighbor-oblivious wireless network, nodes broadcast all transmissions. We thus introduce a genie, which alters the physical network in two ways. First, each node randomly drops incoming packets with probability $1-\frac{R_i}{C_i}$. Thus, the rate of each link is now R_i, instead of the link capacity C_i. Second, packets entering a virtual node are assigned to exactly one physical node, in a ratio proportional to the outgoing rates from the virtual node. Thus, there is now only a single copy of each transmitted packet, regardless of how many nodes receive it, and standard queueing methods can be applied.

We note that in both of these modifications, the action of the genie is to force some nodes to drop packets that they actually received. We prove in Lemma 3 below that this genie can only decrease the end-to-end throughput. Thus, the *genie-degraded* network provides a lower bound on the actual system.

Lemma 3: *The unicast capacity of a network using random network coding cannot increase*

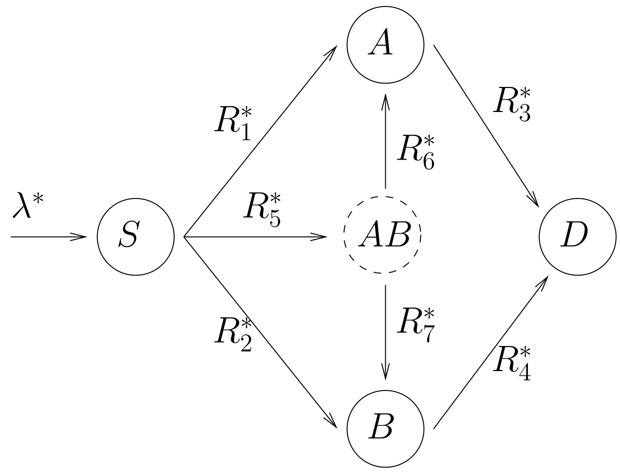

Figure 4.8: A flow assignment that is compatible with the link capacity graph in Figure 4.7 and achieves the maximum flow from S to D.

when nodes discard a subset of packets that were successfully received.

Proof of Lemma 3: We begin by examining a time-space representation of the system, shown in Figure 4.9. All of the nodes in a row represent the same physical node, with each column corresponding to a separate time step. Thus, we can think of the graph as "unwrapping" the time axis. We are interested in the max-flow from the source at the first time step (the top left node) to the destination at the final time step (the bottom right node). Observe that the horizontal edges represent each node's internal buffer, i.e., each physical node stores all packets that it receives and can code over them in future time slots. The diagonal edges represent successful transmissions between nodes.

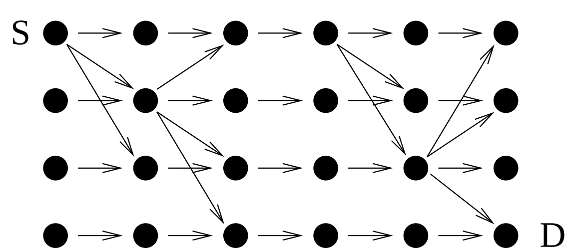

Figure 4.9: A time-space representation of the network. Each row represents one physical node and each column represents a different time step. The amount of information transmitted from the source to the destination is the min-cut between the top left node and the bottom right node.

A node ignoring a packet that it received is equivalent to removing one of the diagonal edges in the time-space graph, which cannot increase the min-cut. Thus, when coding over an infinite field, where the max-flow always equals the min-cut, the effect of removing some edges is that the max-flow either is unchanged or decreases.

When coding over a finite field, the probability that the destination can decode the source data is at least $(1 - 1/q)^{\eta}$ where q is the field size and η is the number of links carrying coded packets [29]. Removing edges will reduce η, which increases this lower bound on the probability of decoding. Thus, removing links has not increased the min-cut, but we must increase the field size q if we want to guarantee that the decoding probability in the original system meets a desired target. □

With the genie inserted, there is now only one copy of each packet in the network, even if a transmission was physically received by multiple nodes. Therefore, we can represent the flow of information in the genie-degraded system by a queueing network. The queue length at each node represents how many more packets a node has received than it has successfully transmitted to one of its neighbors[2].

We also note that because of interference constraints in the underlying wireless network, it is not possible for adjacent servers in the queueing network to process packets at the same time. Therefore, the server processes are correlated across the nodes, and the queueing network is a *generalized* Jackson network, as opposed to a standard Jackson network with independent service processes.

Stability Conditions of Generalized Jackson Networks

A generalized Jackson network refers to a network of queues where the only requirements on the arrivals, service times, and routing decisions are that they are stationary and ergodic, i.e., the service times at different servers do not have to be independent. An example network is shown in Figure 4.10. A fundamental problem is to find the conditions on the arrival rate λ such that the queueing network is *stable*, which means that none of the queue lengths grow arbitrarily large.

It is assumed that there are K stations in the network, numbered $1, 2, \ldots, K$. In addition, station 0 will represent the source of the external arrivals and station $K + 1$ will denote customers that leave the network. Furthermore $\sigma_j^{(k)}$ denotes the service time of the j^{th} customer served at station k, and $\nu_j^{(k)}$ denotes the station to which this customer is sent after service is complete, i.e., the "next hop" in the network. We emphasize that in this model the service times and routing decisions are associated with the servers and not the customers, which is also true in the uncoordinated wireless network that we are studying.

Because of the stationarity and ergodicity assumptions, the long run averages of these variables are well defined. For the service times, we use the notation

$$\frac{1}{n}\sum_{j=1}^{n}\sigma_j^{(k)} \to \frac{1}{\mu^{(k)}} \qquad 1 \le k \le K$$

[2]Recall that each node stores all packets that it receives, and therefore the physical buffer at each node only increases in length. Because there is no feedback in the network layer protocol, nodes do not know which of their transmissions were lost due to collisions. The queue length is thus a virtual quantity, of which nodes do not have any knowledge. One consequence is that it is not possible to implement schemes where the transmission probability is modulated by queue length, such as [22, 23].

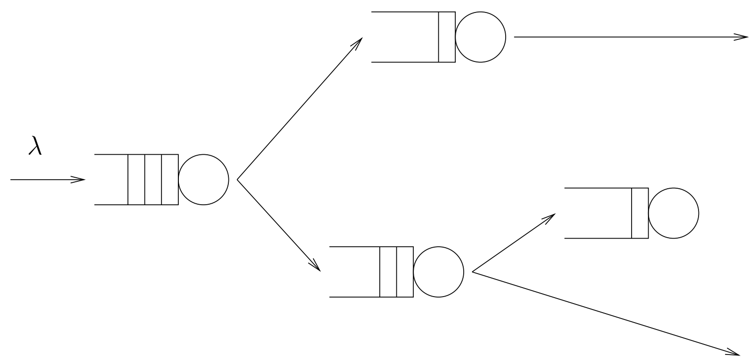

Figure 4.10: A generalized Jackson network. The service times and routing decisions need only be stationary and ergodic.

For the routing decisions, let $P_{i,j}(n) = \sum_{\ell \leq n} \mathbf{1}_{\{\nu_\ell^{(i)}=j\}}$ where $0 \leq i \leq K$ and $1 \leq j \leq K+1$. In words, $P_{i,j}(n)$ counts how many of the first n customers served at station i are routed to station j. Then, the averages of these variables converge to the following matrix

$$\frac{1}{n}P_{i,j}(n) \to p_{i,j} \qquad 0 \leq i \leq K, \qquad 1 \leq j \leq K+1$$

To find the stability conditions, we first must solve the following system of equations

$$\pi_i = p_{0,i} + \sum_{j=1}^{K} p_{j,i}\pi_j \qquad\qquad i = 1, \ldots, K$$

The variable π_i can be interpreted as the average number of times that a customer is processed by station i while it is in the network.

The stability of the network is defined via the time needed to empty all of the queues in the network. The variable Z denotes the amount of time to empty the system after the last external arrival, and A denotes the event that $Z = \infty$. The main result is stated in terms of the probability of event A, as a function of the external arrival rate λ. This is made mathematically precise by considering a generalized Jackson network with only n external arrivals. The time to empty this network after the last arrival is denoted by $Z_{[-n,0]}$, and A is defined by the following limit

$$A = \left\{Z = lim_{n\to\infty} Z_{[-n,0]} = \infty\right\}$$

The stability conditions for the generalized Jackson network are given by the following theorem due to Lelarge [44], which is equivalent to earlier results due to Baccelli and Foss [45, 46]. We state the result without proof here.

Theorem 2. *Let*

$$\rho = \lambda \max_{1\leq i\leq K} \frac{\pi_i}{\mu^{(i)}}$$

If $\rho < 1$ *then* $P(A) = 0$*, while if* $\rho > 1$ *then* $P(A) = 1$.

We note that this result gives a sharp threshold condition for stability. Depending on the choice of the arrival rate λ, the network is either stable with probability one or unstable with probability one.

Also, $\pi_i/\mu^{(i)}$ can be interpreted as the total service time of a customer at station i, i.e., the average number of times it visits the station multiplied by the average service time for each visit. Thus, the theorem states that the arrival rate must be small enough that none of the servers are overwhelmed. By maximizing over the servers, we are checking that even the worst station is not overwhelmed by arrival rate λ.

Stability Conditions of the Neighbor-Oblivious Wireless Network

To demonstrate the desired result, that the maximum throughput of the wireless network is the min-cut of the equivalent wired network, we apply this theory of generalized Jackson networks to the queuing network that we derived from the neighbor-oblivious wireless network.

We first describe the connection between stability of the queuing network and the rate of the wireless network. If the queueing network is stable, the queue length at each node is finite and there are only a finite number of packets at all of the intermediate nodes at any given time. As time gets large, the rate of packets arriving at the destination must approach λ, the rate of packets injected into the network from the source. On the other hand, if the queueing network is unstable, one or more queue lengths becomes infinite and the rate of packets arriving at the destination is less than λ. Therefore, the maximum end-to-end throughput is equal to the maximum rate λ for which the queueing network is stable.

Theorem 3: *The generalized Jackson network derived from a rate allocation graph is stable if λ is less than the min-cut of the graph.*

Proof of Theorem 3:

The mean service rate of a node i, denoted by $\mu^{(i)}$ in the above notation, is given by the total capacity out of that node:

$$\mu^{(i)} = \sum_{e:out(i)} R_i^*$$

Further, the transition probabilities p_{ij} are given by the fraction of outgoing rate from node i that is assigned to the link from i to j:

$$p_{ij} = \frac{R_{ij}}{\sum_{e:Out(i)} R_e}$$

Thus, for any $\lambda < \lambda^*$, we can write the flow balance equation at node i as

$$\sum_{e:Out(i)} R_e = \sum_{e:In(i)} R_e$$

Defining $V_i = \sum_{e:Out(i)} R_e$, we have

$$\sum_{e:In(i)} R_e = \sum_{e=(j,i)} \frac{R_e}{V_j} V_j = \sum_{e=(j,i)} p_{ji} V_j = V_i$$

Thus, by setting $V_i = \lambda \pi_i$, we recover the same system of equations as in Theorem 2. Applying the theorem shows that

$$\frac{\lambda \pi_i}{\mu^{(i)}} = \frac{V_i}{\mu^{(i)}} = \frac{\sum_{e:Out(i)} R_e}{\sum_{e:Out(i)} R_e^*} = \gamma < 1$$

Therefore, Theorem 2 is satisfied and the network is stable. □

Combining Theorem 3 and Lemma 3 leads to the desired conclusion. Theorem 3 proves that if we introduce a genie which drops packets so that the rate on the links is a valid flow with value λ less than the min-cut of the equivalent wired network, then the destination receives a rate λ. Lemma 3 then shows that in the actual wireless network, the end-to-end throughput must be at least as much as the throughput in this genie-degraded network. Therefore, any rate less than the min-cut of the equivalent wired network can be supported, even when the possibility of starvation is allowed. It is not necessary to assume that all nodes are in a saturated state.

Chapter 5

An Uncoordinated VANET Communication Protocol

A completely uncoordinated communication scheme for a VANET, as opposed to the synchronous scheme presented in Chapter 3 which required some coordination to spatially partition the vehicles, can be viewed as an instance of a neighbor-oblivious wireless network. We will refer to a network that uses the distributed MAC protocols outlined in Chapter 2 and random network coding, without any spatial synchronization, as an *asynchronous* scheme, in contrast to the synchronous scheme in Chapter 3.

We originally only required that the VANET protocol be distributed, while still allowing for the possibility that vehicles could know, or learn, their neighbors. However, requiring the VANET to operate in a neighbor-oblivious mode does not relax any of the original conditions, and also prevents the use of feedback. This will simplify the implementation of such a scheme in a practical deployment.

Theorem 3 in Chapter 4 shows that the capacity of the vehicular network is equal to the min-cut of the link capacity graph. The two main challenges when studying the VANET are (a) modeling the connectivity of the network, and (b) computing the min-cut of the link capacity graph. Because the vehicles are no longer grouped into spatial blocks, a more precise model of the vehicle locations and connectivity is necessary.

The rest of this chapter is organized as follows. In Section 5.1 we present a model for the VANET topology. In Section 5.2 we calculate the end-to-end throughput of a dense vehicular network under the slotted Aloha and unslotted Aloha MAC protocols, and discuss the optimal centralized protocol. We show that the homogeneous connectivity of a dense network enables data to be transmitted over large distances without a loss in performance. Finally in Section 5.3 we give simulation results that compare the throughput of the asynchronous scheme under routing and network coding, and the throughput of the asynchronous scheme to the synchronous scheme proposed in Chapter 3

5.1 Modeling the Vehicular Network Connectivity

The transmission power used in vehicular ad hoc networks is typically quite large, which in turn leads to a large transmission range. For example, transmitters operating in the Dedicated Short Range Communication (DSRC) band can transmit up to a range of about 300 meters [47]. Thus, the transmission range is significantly larger than the width of the road, which is no more than about 50 meters for an 8 lane highway, and clearly much smaller for a road with fewer lanes. Figure 5.1 shows the relative sizes of the transmission range and road width when the vehicles are making full use of the available transmit power.

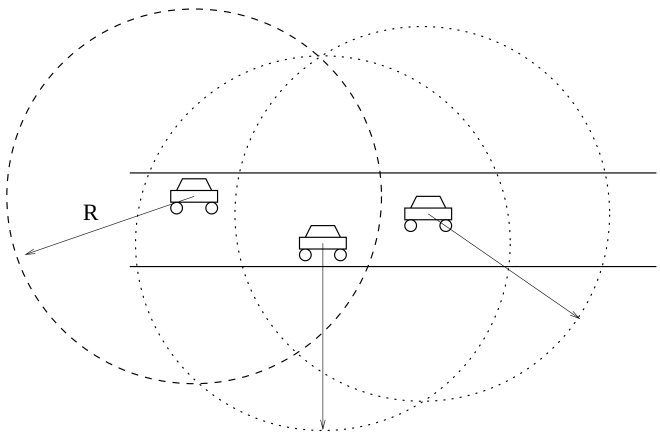

Figure 5.1: The transmission range R of a vehicular network is typically much larger than the width of the road.

Because the transmission range is much larger than the width of the road, a reasonable model for the VANET connectivity is a one-dimensional line network, as shown in Figure 5.2. Although the vehicles may be located in different lanes, we will make the approximation that each vehicle can communicate with any other vehicles that are within a longitudinal distance R from itself along the road[1].

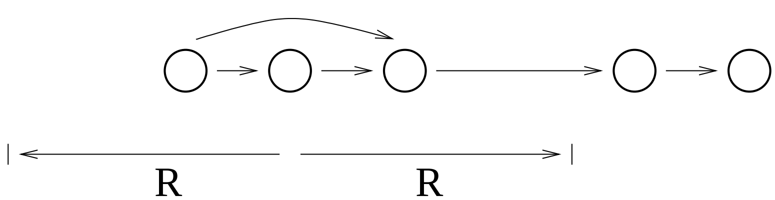

Figure 5.2: The connectivity of the VANET will be modeled by a line network, where each node connects to all neighbors within a distance R of itself.

[1]When vehicles use a low transmit power, this approximation is no longer valid. In the case of extremely low transmit power and a heavily congested road, where all lanes are at the maximum vehicle density, the connectivity can be modeled as a two dimensional grid lattice. We leave the analysis of the throughput of the low power vehicular network to future work.

The number of neighbors of each node depends on the transmission range and the vehicle density. An example where each node has 2 neighbors in both directions is shown in Figure 5.3. A fixed number of neighbors for each node is an appropriate model for a heavily congested highway, where the inter-vehicle spacing is constant. In this work, we focus on such a setting, and in particular quantify how the throughput scales as density increases. For a road in the free-flow state, the distance between vehicles is not constant, e.g., the vehicle locations may be modeled with a Poisson distribution. We will show in the following section that the throughput is governed by the sparsest region of the network.

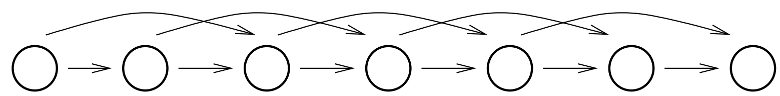

Figure 5.3: The connectivity of a VANET with evenly spaced nodes. Each vehicle has two neighbors in both the forward and reverse directions.

5.2 Throughput of A Dense Vehicular Network

We begin by examining a VANET with constant vehicular spacing, such that there are K edges emanating from each node, i.e., each vehicle can communicate with the next K vehicles along the road. In this section, we give exact expressions for the end-to-end throughput with the slotted Aloha and unslotted Aloha MAC protocols, and an upper bound on the throughput with a carrier sense MAC protocol.

We also make the additional assumption that the source node, which could be either a vehicle or a fixed base station, has the ability to communicate with its neighbors without contention. This could be implemented either by allowing local coordination at the source node, or by allocating an auxiliary channel for the source to "seed" its own neighbors. Likewise, we also assume that the destination node can collect all of the packets received by its neighbors.

The motivation for these assumptions is to allow the unicast flow to take full advantage of the capacity provided by the multihop network. When the source and destination randomly access the channel, similar to the other nodes, the min-cut of the link capacity graph will be the cut that separates the source from the rest of the graph and the cut that separates the destination from the rest of the graph. Conceptually, the source cannot get information into the network fast enough to take advantage of the relays. By allowing the source and destination to communicate without random access, the min-cut is not determined by the cuts in the boundary regions of the network. For simplicity, we will model the source and destination as having infinite capacity links to their K neighbors[2], as shown in Figure 5.4.

In this case, the minimum cut is given by any cut which does not pass through these infinite capacity links at the source and destination. Two such cuts are shown in Figure 5.5.

[2] In practice, these links will have finite capacity. However, since our objective is only to find the min-cut of the network, setting the capacity of links that do not pass through the minimum cut to be infinite will not affect the result.

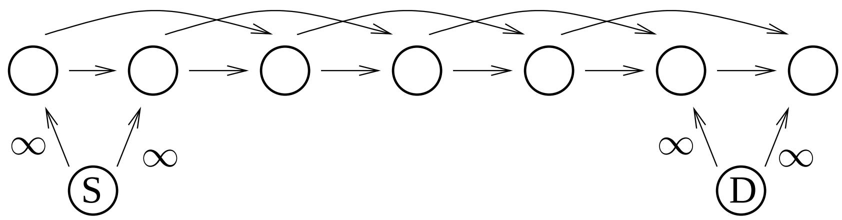

Figure 5.4: A model of a vehicular network in which the source and destination can communicate with their neighbors without randomly contending for the channel, as the relay nodes do.

By the symmetry of the graph, all such cuts must have the same value. This proves our first key result for the asynchronous transmission scheme: *the end-to-end throughput of the VANET is independent of the number of relay nodes*. Therefore, by using a random MAC protocol and network coding, data can be transmitted arbitrary distances along the road. Thus each vehicle can be viewed as a digital repeater

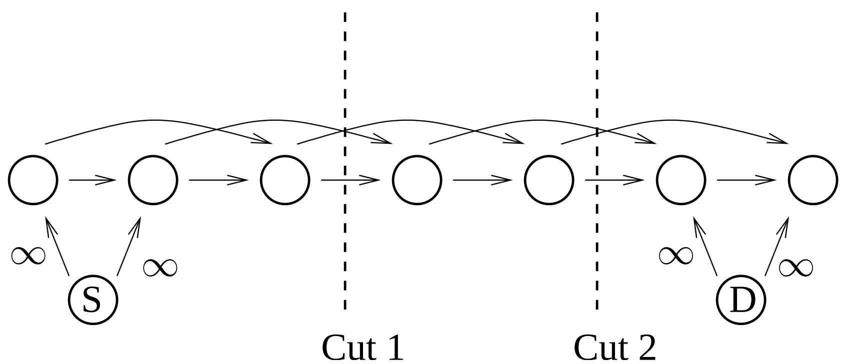

Figure 5.5: In a homogeneous vehicular network, all cuts that do not pass through the infinite capacity edges near the source and destination will have the same (minimum) value.

We emphasize that this is not an intuitively obvious result. In the synchronous scheme presented in Chapter 3 the capacity is constant if the number of blocks grows subexponentially in the number of vehicles per block (the vehicle density), but decays to zero otherwise. In contrast to that result, the asynchronous scheme allows us to transmit data over arbitrarily long distances while maintaining a constant throughput.

Our second result is to compute the numerical value of the capacity under the MAC protocols discussed in Section 2.2. We summarize the results in the limit of large density (large number of outgoing links K from every node) in Table 5.1, and discuss the computation in the following subsections.

MAC protocol	Capacity
Slotted Aloha	0.2248
Unslotted Aloha	0.1012
TDMA	0.5

Table 5.1: The end-to-end throughput of the asynchronous transmission scheme, in the limit of high density, for three common MAC protocols.

5.2.1 Min-Cut of Vehicular Network Graph

We first note that under any MAC protocol, we must introduce virtual nodes to account for the broadcast nature of the wireless channel, as discussed in Section 4.3. Because the min-cut is finite, it must be true that none of the infinite capacity edges from the virtual nodes to the physical nodes pass through it[3]. Therefore, we must consider cuts of the form shown in Figure 5.6, where all of the virtual nodes that can be reached from nodes to the left of the cut are on the right side of that cut.

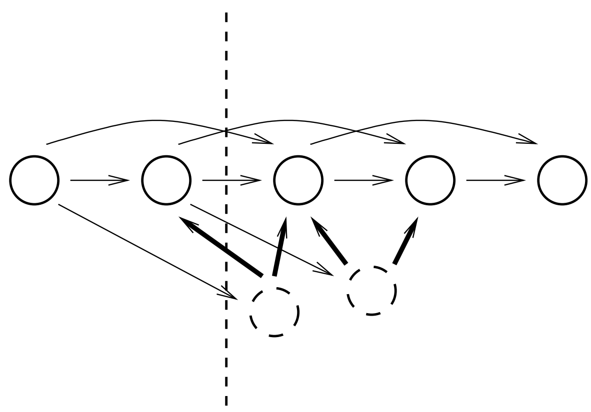

Figure 5.6: The topology of the vehicular ad hoc network. The virtual nodes are represented by the dashed circles, and the infinite capacity edges from the virtual nodes to the physical nodes by the bold lines. The minimum cut is a cut with no infinite capacity edges passing through it from the source side to the destination side.

5.2.2 Throughput under Slotted Aloha MAC Protocol

With slotted Aloha, the set of active links in any time slot is independent of all other time slots. The Markov chain representing the network is then fully connected, with transition probabilities that depend only on the destination state. Thus, each link capacity can be directly computed by finding the probability that the link is active in a single time slot.

[3]Specifically, no infinite capacity edges pass through the min-cut from the source side of the graph to the destination side.

We assume that each vehicle transmits with probability p in every time slot, independent of all other times and vehicles, where p is a parameter that will be optimized. We compute the sum of the values of all links crossing the cut shown in Figure 5.6. To simplify the calculation, we will group together links by the "right most" node in the destination set. As a concrete example, consider Figure 5.7 where each node can transmit to $K = 2$ neighbors. When node A transmits, the packet may be received by B only, C only, or both B and C. We simplify the computation of the cut by finding the sum of the link from A to C and the link from A to the virtual node in one step. This sum is the probability that a transmission from node A is received by node C, regardless of whether or not B receives it.

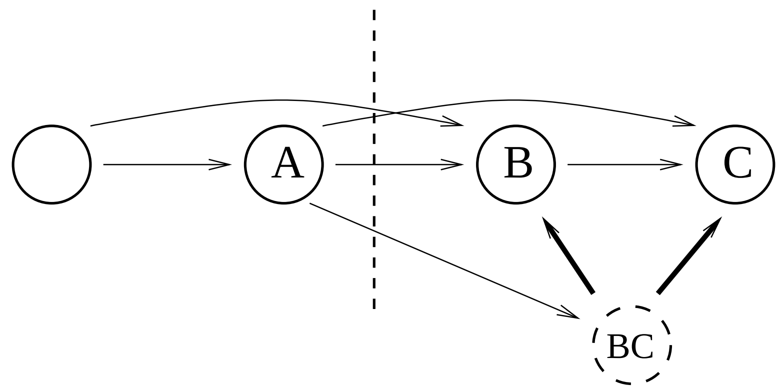

Figure 5.7: To simplify computation of the min-cut of the vehicular network, the weight of the edge from A to C and the edge from A to the virtual node BC will be calculated in one step.

After grouping the links in this manner, there are now two general classes of links crossing the cut: groups that contain the "furthest neighbor" from the transmitter, and groups that do not reach the furthest neighbor. The total probability of each group in the first class is $p(1-p)^{2K}$, which is the probability that the transmitting node chooses to access the channel (with probability p) and the furthest destination node and its remaining $2K - 1$ neighbors are silent. There are K such links crossing the cut.

The probability of each group in the second class is $p^2(1-p)^{2K}$, because a second (interfering) node must also transmit to prevent the packet from being received by the furthest neighbor of the transmitter. The number of links in this class is given by

$$\sum_{m=1}^{K-1} m = \frac{K(K-1)}{2}$$

Therefore, the sum of the link capacities of all links crossing the cut is given by

$$Kp(1-p)^{2K} + \frac{K(K-1)}{2}p^2(1-p)^{2K}$$
$$\approx Kp(1-p)^{2K} + \frac{K^2}{2}p^2(1-p)^{2K}$$

By setting $p = \frac{1}{2K}$, the capacity in the limit of large K can be found using a standard approximation as $\frac{1}{2}\frac{1}{e} + \frac{1}{8}\frac{1}{e} = \frac{5}{8e} \approx 0.23$. The capacity can be further increased by optimizing over p. However, numerical optimization shows that even for large K the improvement by using the optimum p is relatively small, as shown in Figure 5.8.

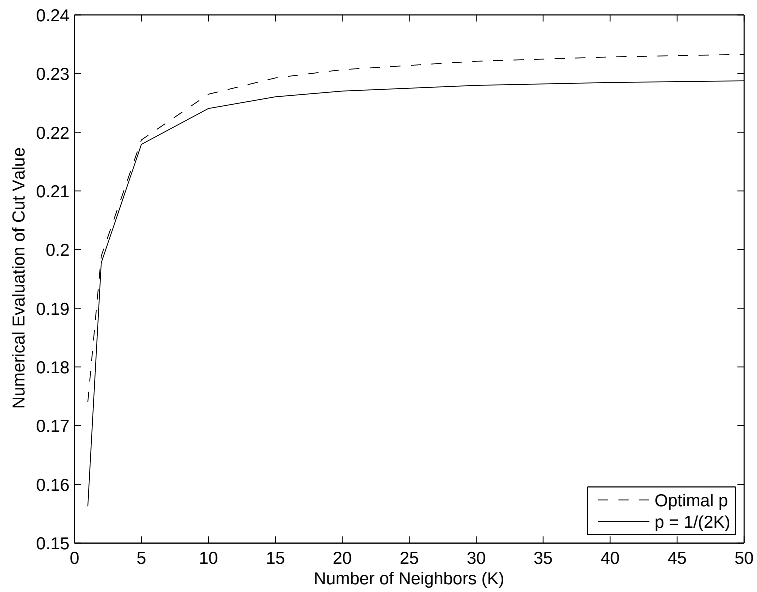

Figure 5.8: A numerical evaluation of the expression for the min-cut under slotted Aloha, for both the optimal transmission probability p and the nominal value $p = \frac{1}{2K}$, reveals that using the optimal value provides only a small improvement over the nominal value, even for large K.

To verify these results, we simulated the throughput in a network with the homogeneous topology shown in Figure 5.4 using slotted Aloha and random network coding. First, we checked whether the end-to-end throughput does approach the predicted limit of $\frac{5}{8e}$ as the number of neighbors of each node K increases. Figure 5.9 shows a simulation of the throughput as a function of K. The total number of nodes between source and destination was varied, so that for all values of K, the expected number of hops was approximately five[4]. Here, we simulated an idealized scenario, where network coding was done over the real field. Thus, packets were only discarded by the receiver in the case when the transmitter was starved of new information, and never because the transmitter had novel information but chose a bad set of mixing coefficients. Although in practical settings the coding must be done over a finite field, by using a real field we were able to isolate the effect of starvation from the effect of random coding over a finite field. The plot shows that when K is on the order of 20, the min-cut is already close to the limiting value of the high density regime.

[4]Observe that increasing K for fixed network size presents an unfair comparison, since each packet will travel fewer hops between source and destination and thus the problem of starvation will be reduced.

Next, we simulated the dependence of the end-to-end throughput on the distance between source and destination. Figure 5.10 shows a plot of the simulated throughput as a function of the number of nodes between the source and destination, where the neighborhood size is fixed at $K = 5$ and the network coding is again done over the real field. This plot demonstrates that the throughput is in fact nearly constant when the distance to the destination increases, as our analysis suggests.

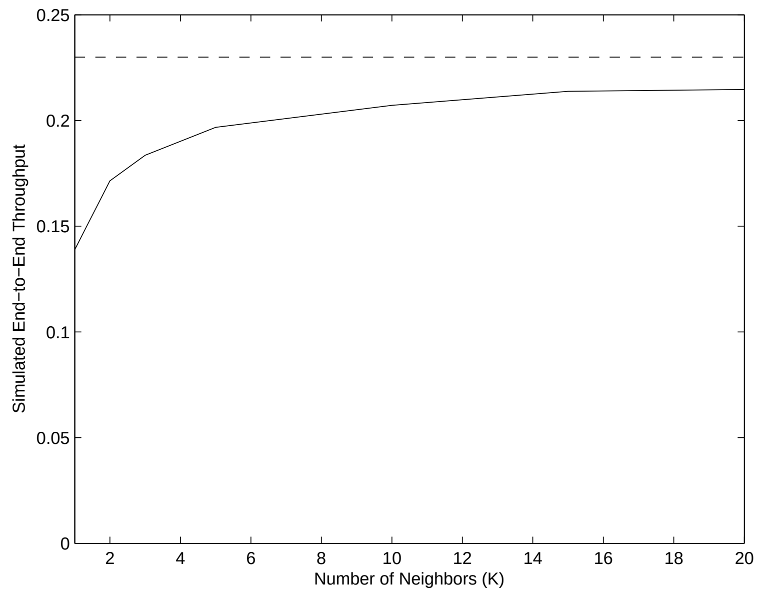

Figure 5.9: A simulation of the end-to-end throughput as a function of the neighborhood size K for the homogeneous topology shown in Fig 5.4. The network coding is done over the real field.

5.2.3 Throughput under Unslotted Aloha MAC Protocol

We model the unslotted Aloha protocol with a system where each vehicle implements slotted Aloha, but the slot boundaries are offset by a random phase. In this case, the same analysis as with the slotted Aloha MAC applies. However, when a transmitter sends a packet, the receiver and the interfering nodes must be silent for *two* consecutive time slots because of the offsets[5]. (See Figure 5.11 for a graphical example of the offset time slot boundaries.) Therefore, the capacity under unslotted Aloha is approximately equal to

$$Kp(1-p)^{4K} + \frac{K^2}{2}p^2(1-p)^{4K}$$

[5]We assume that whenever two transmissions collide for any portion of the packet duration, the entire packet is lost.

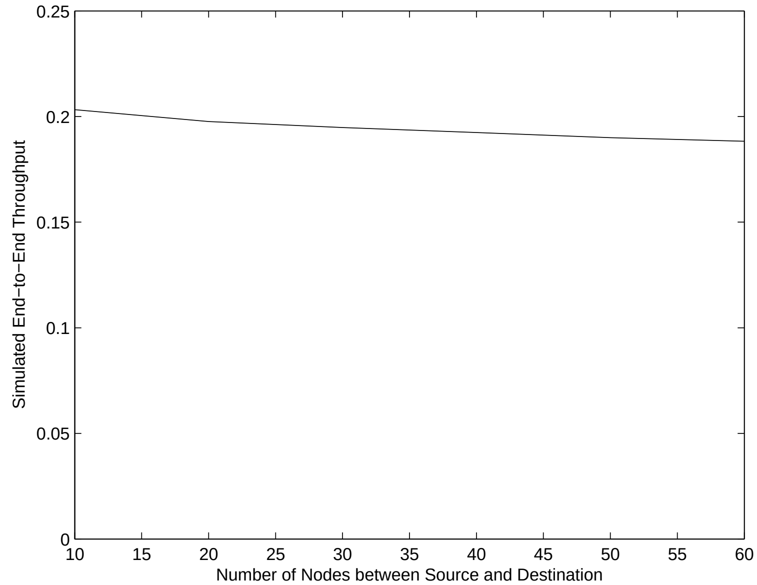

Figure 5.10: A simulation of the end-to-end throughput as a function of the number of nodes between the source and the destination for the homogeneous topology shown in Fig 5.4. The neighborhood size is $K = 5$ and the coding is done over the real field.

By setting $p = \frac{1}{4K}$, the capacity can be analytically found to be equal to $\frac{9}{32e}$. As in the case of slotted Aloha, numerical optimization shows that the capacity with the optimal probability p is only slightly better.

5.2.4 Throughput under TDMA

While it is possible to write a general closed-form expression for the throughput of a linear network (a network where each node has $K = 1$ neighbors) using CSMA, evaluating it is difficult. In addition, even finding such an expression for cases with $K > 1$ is complex. Thus, computing the end-to-end throughput of an uncoordinated VANET using CSMA is beyond the scope of this dissertation.

However, in networks with no hidden nodes, it is known that a dual-channel CSMA scheduler can achieve the same average throughput scaling law as the optimal TDMA scheduler [25]. Furthermore, a single-channel CSMA scheduler is known to achieve the same achievable rate region as the optimal TDMA scheduler, except for the boundary region [43, 22]. For our purposes, this means that it is instructive to study the throughput of the VANET under an optimal TDMA protocol, both because this provides an upper bound on what an uncoordinated MAC can achieve and because CSMA can approach this bound in some settings.

Consider the simple network where each node can reach $K = 2$ neighbors shown in Figure 5.12. In order to optimally reuse the channel, in one time slot nodes $1, 6, 11, \ldots$ will transmit

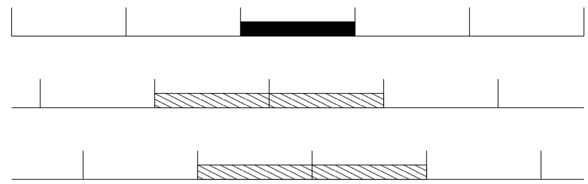

Figure 5.11: In unslotted Aloha, when a node transmits a packet (the shaded rectangle) neighboring nodes must be silent for two consecutive time slots (the crosshatched rectangles) to prevent a collision.

packets. In the second time slot, nodes $3, 8, 13, \ldots$ will forward the packets that they received in the first slot. In each subsequent slot, packets are forwarded by the "furthest neighbor" that received the transmission in the previous time slot, with transmissions separated by just enough distance to prevent any collisions. In a block of $2K+1$ time slots, each node will receive K novel packets (from nodes on its left), receive K redundant packets (from nodes on its right), and transmit once. Thus, for high densities the capacity of the optimal TDMA scheme approaches $\frac{1}{2}$.

We note that with slotted Aloha, the throughput is about 45% of the throughput in a fully coordinated network. Thus, the "cost of uncoordination" for slotted Aloha is a loss in performance of about a factor of 2.

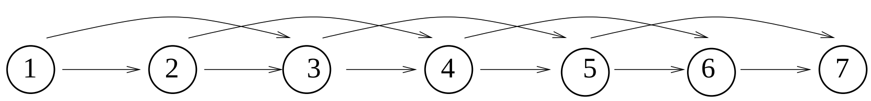

Figure 5.12: The optimal TDMA scheme is found by packing concurrent transmission as tightly as possible without inducing collisions, and having each packet forwarded by the furthest node that received it.

5.2.5 Throughput of Non-Homogeneous Networks

In practical settings, the vehicular network will not have a homogeneous connectivity structure, even in relatively congested regimes. In lighter traffic conditions, the structure will be even more variable. A first approximation to modeling such scenarios is to represent them by regions of constant density. Figure 5.13 shows an example with a more congested region on the left and a less congested region to the right.

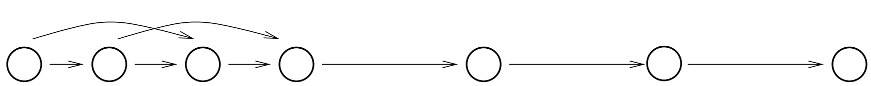

Figure 5.13: To a first order, a non-homogeneous vehicular network can be approximated by a network with several different regions of constant connectivity.

In this case, the min-cut within each region is constant. The analysis in Section 5.2.2 shows that the min-cut is an increasing function of K, the number of neighbors in a homogeneous region. Thus, the smallest cut will be governed by the subsection of the VANET with the least connectivity. (For systems with fixed transmit power, this will be the least dense region, where the vehicles are spaced furthest apart.)

5.3 Simulation Results

We simulated the performance of the asynchronous scheme using the same parameters as the synchronous simulations in Section 3.4. In these simulations, the asynchronous scheme always used the slotted Aloha MAC protocol. Our objective was to compare network coding and routing in the asynchronous scheme, as well as to compare the asynchronous and synchronous protocols. The main differences in the simulation parameters are the following:

- The nodes are not organized into spatial blocks, and are allowed to transmit in any time slot.
- The time axis, although divided into slots as in the synchronous case, is not organized into frames.
- Each node chooses to transmit in every slot with probability $p = 1/(L)$, where L is the number of neighbors within range of a node.

In the asynchronous scheme, there are no longer distinct waves of data flowing down the highway. Thus, many of the practical considerations related to the length of packet headers and the field size, which are minimized in the synchronous case where nodes only combine packets in the same wave, are much more severe. In particular, while the theoretical analysis assumes that vehicles can code over all source packets, this is not possible in even a moderately sized simulation, due to the fact that the packet headers would become unreasonably long. There are thus nontrivial questions about when the infostation should transmit, and how vehicles should separate packets that the infostation originally injected into the network at different times. In these simulations, the source data is divided into finite length epochs instead of coding over a single infinite block. In our simulations, we have assumed that the infostation operates in exactly the same manner as in the synchronous case, transmitting for N slots out of every $4N$ slots. The infostation protocol was kept unchanged in order to facilitate a direct comparison between the synchronous and asynchronous schemes. However, the problem of identifying the optimal choice for the infostation transmission schedule in the asynchronous system is left for future work.

In addition, it was assumed that during each "duty cycle" the infostation would transmit a new epoch of source data, independent of all previously transmitted epochs. When a vehicle hears a packet from a new epoch, it immediately decodes the received packets from the last epoch and flushes them from its buffer. Although this policy of immediately flushing the buffer, which was inspired by [27], is clearly far from optimal, it has the advantage of being simple and robust[6].

We first compared the performance of routing and network coding in the asynchronous protocol. The first parameter we analyzed was the rank of the system as a function of the distance from the infostation. Again, the rank is defined as the number of unique packets per epoch for routing, and the number of independent packets per epoch for network coding. Figure 5.14 shows a plot of average rank versus the distance between the infostation and the destination vehicle. The plot shows that in the asynchronous protocol the rank is constant as the distance increases when network coding is used, while with routing the rank decreases rapidly to values close to 0.

In the case of network coding the asymptotic value of the curve is approximately equal to N/e independent packets per wave. Because the infostation initiates a new wave every $4N$ slots in these simulations, the average throughput at the destination is thus about $\frac{1}{4e}$. We observe that this is less than $\frac{5}{8e}$, which is the throughput predicted by our analysis in Section 5.2.2. However, the practical considerations discussed above clearly cause the performance of the asynchronous scheme to degrade. In particular, the policy of immediately flushing the packets in one epoch as soon as a vehicle receives a single packet in the next epoch causes the throughput to decrease. The design of network protocols which can be implemented in practice and bridge this gap is an important open question.

As in the case of the synchronous protocol, we again measured the total number of received packets, regardless of whether they were independent. Figure 5.15 confirms that there is no difference in the number of raw received packets with the two strategies.

Finally, we also empirically studied the delay introduced by the asynchronous scheme. Figure 5.16 shows the time when the infostation begins transmitting the first packet of each epoch, as well as the time when the destination node receives the last packet of that epoch. The results are shown only for network coding. In the asynchronous scheme, the latency before the first wave is completed is very low, about 0.7 s.

We also compared the performance of the synchronous and asynchronous schemes. Figure 5.17 compares the number of independent packets received at the destination vehicle. In this figure the results obtained in the case of synchronous strategies without overhearing were not considered, since it was empirically shown in Section 3.4 that incorporating overhearing provides better performance without any additional cost. The plot shows that the synchronous scheme can sustain higher rank than the asynchronous scheme, with an increase on the order of 50%. While the asynchronous network coding scheme does not provide the same throughput as the synchronous network coding scheme, we still would like to highlight the fact that the asynchronous scheme can sustain a constant throughput over relatively large distances, without the synchronous block structure. Similar considerations hold in the case of routing, however neither the synchronous nor asynchronous schemes can sustain an acceptable throughput over long distances. The fact that the synchronous scheme outperforms the asynchronous scheme is to be expected, since the spatial partitioning imposes additional structure on the problem. Because additional resources must be invested to set

[6]It is not possible for vehicles to combine packets across source epochs, since the overhead required to transmit the coefficients would eventually consume the entire packet

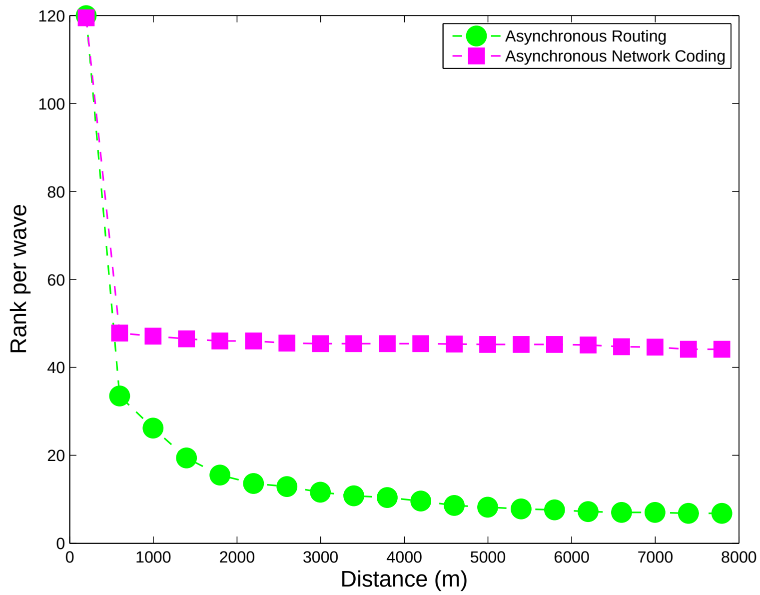

Figure 5.14: The number of independent packets received per epoch as a function of the location of the destination vehicle in the asynchronous transmission scheme. Results are shown for two strategies: routing with the asynchronous scheme and network coding with the asynchronous scheme.

up the spatial blocks, it is unsurprising that better performance is obtained.

On the other hand, the results for the delay, presented in Figure 5.18, indicate that the asynchronous strategy can guarantee a lower initial latency, due to the pipeline effect that is artificially introduced in the synchronous scheme, which forces nodes to wait for a longer time before the first wave is received. The figure shows indeed that there is a trade-off between throughput and latency in the selection of a synchronous or asynchronous method. If the main goal is to guarantee a high throughput and the system has the resources set up the spatial block structure, then the synchronous solution should be used, at the price of a higher complexity and a larger initial latency. On the other hand, if the throughput provided by asynchronous network coding is sufficient, then this strategy is preferable because of the lower complexity and the lower latency.

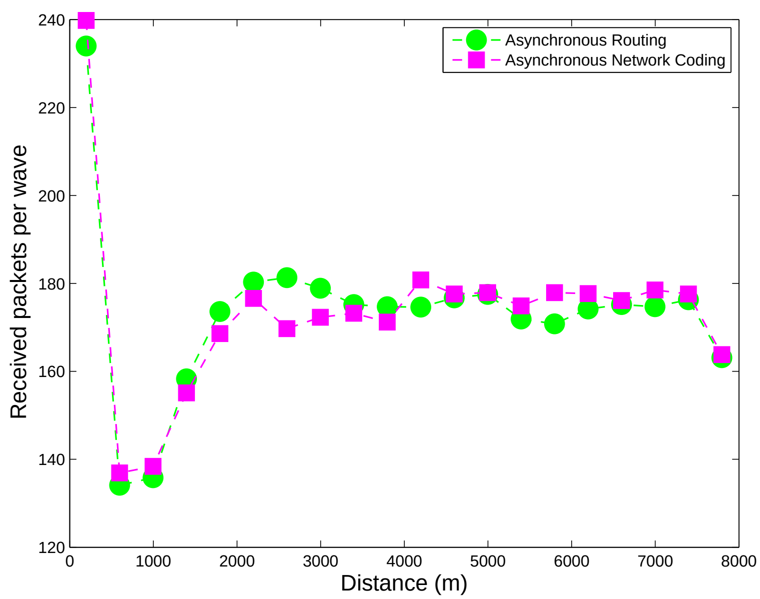

Figure 5.15: The total number of received packets per epoch as a function of the location of the destination vehicle in the asynchronous transmission scheme. This plot counts all packets received without collision, regardless of whether they are independent. Results are shown for two strategies: routing with the asynchronous scheme and network coding with the asynchronous scheme.

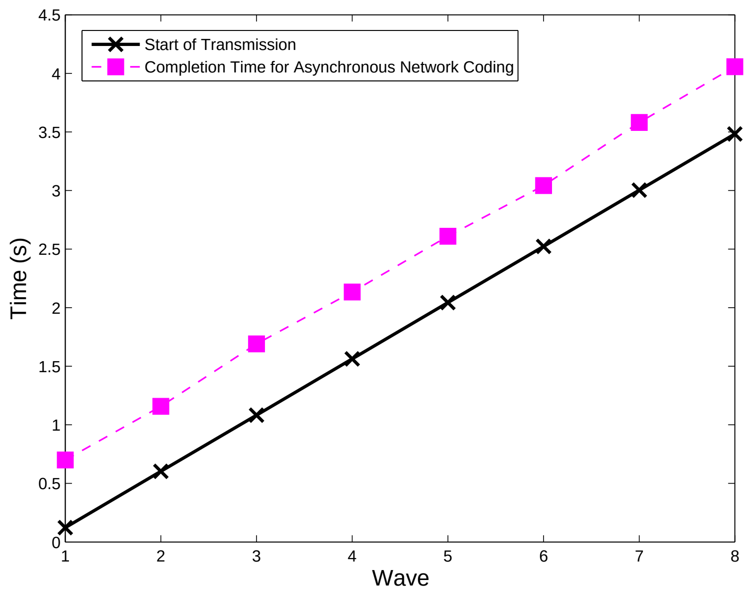

Figure 5.16: The time when the infostation begins transmitting each epoch, and the time when the destination vehicle receives the final packet of the epoch, for the asynchronous scheme. The vertical distance between the two curves is the time delay of the asynchronous scheme with network coding.

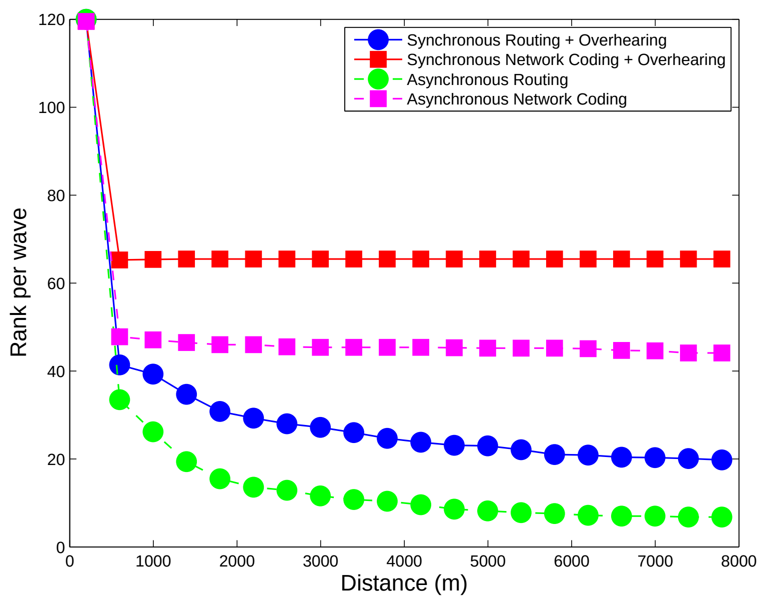

Figure 5.17: The number of independent packets received per wave as a function of the location of the destination vehicle. Results are shown for four strategies: routing with the synchronous overhearing scheme, network coding with the synchronous overhearing scheme, routing with the asynchronous scheme, and network coding with the asynchronous scheme.

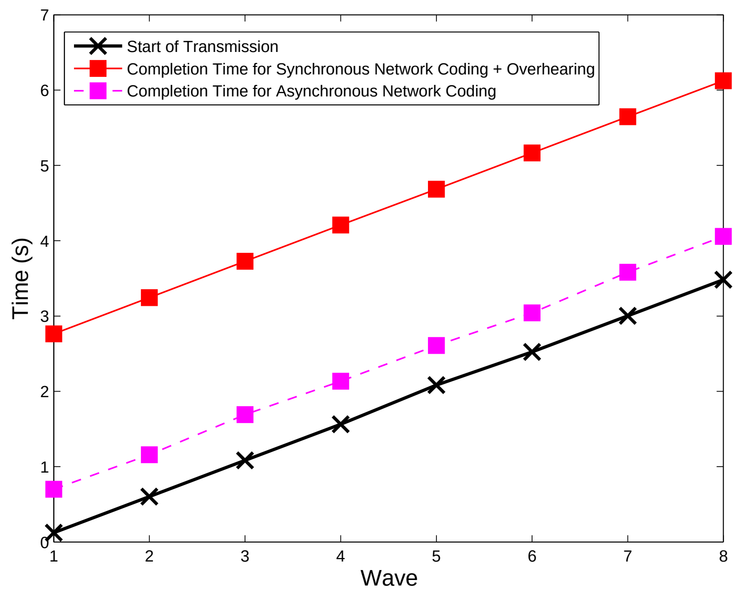

Figure 5.18: The time when the infostation begins transmitting each wave, and the time when the destination vehicle receives the final packet of the wave for network coding with the synchronous overhearing scheme and network coding with the asynchronous scheme. The vertical distance between the curves is the time delay.

Chapter 6

Conclusions and Future Directions

In this dissertation, we have considered the problem of multihop communication in vehicular ad hoc networks. Because vehicles frequently enter and exit roads, and different vehicles may be traveling at very different speeds, the topology of VANETs changes rapidly. Therefore, communication protocols that rely on coordinated scheduling of transmissions to avoid collisions and global routing tables to move data from source to destination are either technically infeasible, or require that a very large portion of system resources by devoted to tracking and disseminating the network topology. We have proposed to instead use randomized, uncoordinated protocols in such networks.

We have proposed to combine randomized MAC protocols, such as Aloha and a simple carrier sensing protocol, with random linear network coding for use in VANETs. Network coding, which is based on the mixing of packets at intermediate nodes, makes the system robust to packet losses without requiring any link level feedback. Our work focuses on analyzing the throughput that is provided by vehicular networks that implement this combination of MAC and network layer protocols.

In particular, we have focused on the case of a single multihop unicast flow, where the vehicles on the road are serving as relays to forward packets from the source to the destination. We began by considering a spatially synchronized protocol. This scheme required a minor degree of coordination, in that vehicles are grouped into spatial blocks and transmissions are scheduled at the block level to move waves of data down the road, although vehicles within each block randomly contend with each other when transmitting. We proved that the throughput with network coding in this synchronous scheme is constant, as long as the number of blocks grows slower than exponentially in the vehicle density.

We then turned to a completely uncoordinated protocol. We showed that when vehicles combine random channel access with network coding, the throughput is given by the min-cut of the link capacity graph between source and destination. The implication is that on a congested highway, where the connectivity is homogeneous and every vehicle has the same number of neighbors, the throughput does not degrade with distance. In contrast to the spatially synchronized protocol, a constant throughput can be supported over an arbitrarily long distance.

The key challenge in demonstrating this result is the problem of starvation. Due to the random ordering of transmissions induced by the randomized MAC protocol, the intermediate nodes may temporarily not have any new information to forward to their neighbors. We used tools from queueing theory to show that starvation does not decrease the throughput below the min-cut of the network. This result has many applications beyond the vehicular networks considered here. It closes an open question in the analysis of CSMA networks, where it is often assumed that all nodes are in a saturated state, meaning that they always have new information to transmit when they access the channel.

There are numerous open questions and directions for future research, both theoretical and applied, that must be addressed in order to make VANETs a practical reality in the near future:

- In practice, in order for the destination to be able to decode its received packets, each packet must contain a header that specifies the linear combination contained in its body. Therefore, vehicles cannot code over an infinite number of source packets. This is especially true in wireless networks, where the packet length must be smaller than the coherence time of the channel, and thus the header must be kept to a reasonably small length. Due to these practical issues, the source data is partitioned into epochs, and intermediate nodes only combine packets in the same epoch. Before deploying a system that incorporates network coding, the performance loss that results from dividing the source data into epochs must be analyzed. In particular, the effect of partitioning the source on the result that a constant throughput can be sent over an arbitrary number of hops must be carefully studied.

- While the problem of sending a single unicast flow through a vehicular network is an important starting point, a practical network must be able to support a variety of other modes of communication, such as multicast. One of the most interesting problems is when there are multiple overlapping streams. Future work must address how receivers that are interested in different flows should share the channel and cooperate with each other in relaying packets.

- In this work, we have focused on data streams which are short enough that the network can be considered effectively static during the duration of transmission. For very long data streams, such as movies, the vehicles will move relative to each other while the data is being relayed, which can improve performance by physically carrying data through regions of low connectivity but presents additional challenges to quantifying network performance.

- We have up to this point considered a simplified physical layer model, where receivers within range of a transmitter can perfectly demodulate a packet, and receivers out of range experience no interference. A more accurate analysis of the system performance should incorporate a more realistic physical layer model based on the SINR at the receiver.

- A more accurate approximation of the individual link capacities under the CSMA protocol, as a function of the number of neighbors of each vehicle, is worth exploring.

We hope that the results contained in this dissertation will prompt further research in the field of protocol design and analysis for vehicular ad hoc networks, and eventually lead to the deployment of practical networks which can serve to enable intelligent transportation systems that improve highway safety and reduce the expenses associated with traffic congestion.

Bibliography

[1] National Highway Traffic Administration. [Online]. Available: http://www.nhtsa.gov/PR/DOT-165-10

[2] Bureau of Transportation Statistics. [Online]. Available: http://www.bts.gov

[3] D. Sperling and D. Gordon, *Two Billion Cars: Driving Toward Sustainability*. Oxford University Press, 2009.

[4] R. Sengupta, S. Rezaei, S. Shladover, D. Cody, S. Dickey, and H. Krishnan, "Cooperative collision warning systems: Concept definition and experimental implementation," *Journal of Intelligent Transportation Systems*, vol. 11, no. 3, July 2007.

[5] C.-L. Huang, Y. Fallah, R. Sengupta, and H. Krishnan, "Adaptive intervehicle communication control for cooperative safety systems," *IEEE Network Magazine*, vol. 24, no. 1, pp. 6–13, January 2010.

[6] A. Thiagarajan, L. Ravindranath, K. LaCurts, S. Toledo, J. Eriksson, H. Balakrishnan, and S. Madden, "VTrack: Accurate, energy-aware road traffic delay estimation using mobile phones," in *ACM SenSys*, Berkeley, CA, November 2009.

[7] J.-C. Herrera, D. Work, X. Ban, R. Herring, Q. Jacobson, and A. Bayen, "Evaluation of traffic data obtained via GPS-enabled mobile phones: the Mobile Century field experiment," *Transportation Research C*, vol. 18, pp. 568–583, 2010.

[8] P. Varaiya, "Congestion, ramp metering, and tolls," *Philosophical Transactions of the Royal Society A*, vol. 366, pp. 1921–1930, 2008.

[9] The New York Times, "Google cars drive themselves, in traffic," October 9, 2010. [Online]. Available: http://www.nytimes.com/2010/10/10/science/10google.html

[10] P. Varaiya, "Smart carts on smart roads: Problems of control," *IEEE Transactions on Automatic Control*, vol. 38, no. 2, February 1993.

[11] D. Work, O.-P. Tossavainen, Q. Jacobson, and A. Bayen, "Lagrangian sensing: Traffic estimation with mobile devices," in *American Control Conference*, June 2009.

[12] J. Eriksson, H. Balakrishnan, and S. Madden, "Cabernet: Vehicular content delivery using wifi," in *ACM MOBICOM*, San Francisco, CA, September 2008.

[13] H. Hartenstein and K. Laberteaux, “A tutorial survey on vehicular ad hoc networks,” *IEEE Communications Magazine*, vol. 46, no. 6, pp. 164–171, June 2008.

[14] M. Torrent-Moreno, “Inter-vehicle communications: Achieving safety in a distributed wireless environment,” Ph.D. dissertation, University of Karlsruhe, Karlsruhe, Germany, 2007.

[15] Federal Communications Commission, “Report and order,” October 1999. [Online]. Available: http://hraunfoss.fcc.gov/edocs_public/attachmatch/FCC-99-305A1.pdf

[16] K. Sohrabi, J. Gao, V. Ailawadhi, and G. Pottie, “Protocols for self-organization of a wireless sensor network,” *IEEE Personal Communications*, vol. 7, no. 5, pp. 16–27, October 2000.

[17] S. Borbash, A. Ephremides, and M. McGlynn, “An asynchronous neighbor discovery algorithm for wireless sensor networks,” *Ad Hoc Networks*, vol. 5, no. 7, pp. 998–1016, September 2007.

[18] California Department of Transportation. [Online]. Available: http://video.dot.ca.gov/

[19] P. Gupta and P. R. Kumar, “The capacity of wireless networks,” *IEEE Trans. on Information Theory*, vol. 46, no. 2, pp. 388–404, March 2002.

[20] N. Abramson, “The ALOHA system - another alternative for computer communications,” in *1970 Fall Joint Comput. Conf., AFIPS Conf. Proc.*, Montvale, NJ, 1970, pp. 281–285.

[21] R. Boorstyn, A. Kershenbaum, B. Maglaris, and V. Sahin, “Throughput analysis in multihop CSMA packet radio networks,” *IEEE Transactions on Communications*, vol. 35, pp. 267–274, March 1987.

[22] L. Jiang and J. Walrand, “A distributed CSMA algorithm for throughput and utility maximization in wireless networks,” in *Allerton Conference on Communication, Control, and Computing*, Monticello, IL, Oct. 2008.

[23] J. Ni, B. Tan, and R. Srikant, “Q-CSMA: Queue-length based CSMA/CA algorithms for achieving maximum throughput and low delay in wireless networks,” in *IEEE INFOCOM Mini-Conference*, San Diego, CA, March 2010.

[24] J. Liu, Y. Yi, A. Proutiere, M. Chiang, and H. Poor, “Towards utility-optimal random access without message passing,” *Special issue in Wiley Journal of Wireless Communications and Mobile Computing*, December 2009.

[25] C.-K. Chau, M. Chen, and S. Liew, “Capacity of large scale CSMA wireless networks,” in *ACM MOBICOM*, Beijing, China, September 2009.

[26] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, “Network information flow,” *IEEE Trans. on Information Theory*, vol. 46, no. 4, pp. 1204–1216, July 2000.

[27] P. A. Chou, Y. Wu, and K. Jain, "Practical network coding," in *Allerton Conference on Communication, Control, and Computing*, Monticello, IL, Oct. 2003.

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

[29] T. Ho, M. Medard, J. Shi, M. Effros, and D. Karger, "On randomized network coding," in *Allerton Conference on Communication, Control, and Computing*, Monticello, IL, Oct. 2003.

[30] W. H. Yuen, R. D. Yates, and S. C. Mau, "Noncooperative content distribution in mobile infostation networks," in *IEEE Wireless Communications and Networking Conference (WCNC)*, New Orleans, LA, March 2003.

[31] W. H. Yuen, R. D. Yates, and C. W. Sung, "Performance evaluation of highway mobile infostation networks," in *IEEE Global Telecommunications Conference (GLOBECOM)*, Dec. 2003.

[32] W. H. Yuen, R. D. Yates, and S. C. Mau, "Exploiting data diversity and multiuser diversity in noncooperative mobile infostation networks," in *IEEE INFOCOM*, San Francisco, CA, April 2003.

[33] U. Lee, J.-S. Park, J. Yeh, G. Pau, and M. Gerla, "CodeTorrent: Content distribution using network coding in VANETs," in *First International Workshop on Decentralized Resource Sharing in Mobile Computing and Networking (MobiShare)*, Los Angeles, CA, Sept. 2006.

[34] A. Nandan, S. Das, G. Pau, M. Gerla, and M. Sanadidi, "Co-operative Downloading in Vehicular Ad-hoc Wireless Networks," in *International Conference on Wireless On Demand Network Systems and Services (WONS)*, St. Moritz, Switzerland, Jan. 2005.

[35] D. Petrovic, K. Ramchandran, and J. Rabaey, "Overcoming untuned radios in wireless networks with network coding," *IEEE Trans. on Information Theory*, vol. 52, no. 6, pp. 2649–2657, June 2006.

[36] M. Aizenman, J. T. Chayes, L. Chayes, J. Frohlich, and L. Russo, "On a sharp transition from area law to perimeter law in a system of random surfaces," *Communications in Mathematical Physics*, vol. 92, pp. 16 – 69, 1983.

[37] OMNeT++ Network Simulation Framework. [Online]. Available: https://www.omnetpp.org

[38] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE Transactions on Automatic Control*, vol. 37, no. 12, pp. 1936–1948, December 1992.

[39] S. Rajagopalan and D. Shah, "Distributed algorithm and reversible network," in *Conference on Information Sciences and Systems*, March 2008.

[40] J. Lee, J. Lee, Y. Yi, S. Chong, A. Proutiere, and M. Chiang, "Implementing utility-optimal CSMA," in *Allerton Conference on Communication, Control, and Computing*, Monticello, IL, Oct. 2009.

[41] X. Wang and K. Kar, "Throughput modelling and fairness issues in CSMA/CA based ad-hoc networks," in *IEEE INFOCOM*, Miami, FL, March 2005.

[42] S. C. Liew, C. Kai, J. Leung, and B. Wong, "Back-of-the-envelope computation of throughput distributions in CSMA wireless networks," in *IEEE International Conference on Communications (ICC)*, Dresden, Germany, June 2009.

[43] M. Chen, S. Liew, Z. Shao, and C. Kai, "Markov approximation for combinatorial network optimization," in *IEEE INFOCOM*, San Diego, CA, March 2010.

[44] M. Lelarge, "Fluid limit of generalized Jackson queueing networks with stationary and ergodic arrivals and service times," *Journal of Applied Probability*, vol. 42, pp. 491–512, 2005.

[45] F. Baccelli and S. Foss, "Ergodicity of Jackson-type queueing networks," *Queueing Systems (QUESTA)*, no. 17, pp. 5 – 72, 1994.

[46] ——, "On the saturation rule for the stability of queues," *Journal of Applied Probability*, vol. 32, pp. 494–507, 1995.

[47] Q. Xu, T. Mak, J. Ko, and R. Sengupta, "Medium access control protocol design for vehicle-vehicle safety messages," *IEEE Transactions on Vehicular Technology*, vol. 56, no. 2, pp. 499–518, March 2007.