Title
Multi-hop routing for wireless mesh networks

Author
Bhorkar, Abhijeet

Publication Date
2012

UNIVERSITY OF CALIFORNIA, SAN DIEGO

Multi-hop Routing for Wireless Mesh Networks

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Electrical Engineering (Communication Theory & Systems)

by

Abhijeet Bhorkar

Committee in charge:

Professor Tara Javidi, Chair
Professor Massimo Franceschetti
Professor Bill Lin
Professor Bhaskar Rao
Professor Alex Snoeren

2012

The dissertation of Abhijeet Bhorkar is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

Chair

University of California, San Diego

2012

DEDICATION

To my beloved family

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

First and foremost, I owe my deepest gratitude to my advisor, Prof. Tara Javidi for her guidance, understanding, patience. She guided me in my research,inspired me with her keen intuitions and logical thinking, and offered me help when there was turmoil in my life. She encouraged me to grow as an experimentalist with theorist perception. I would also like to thank Prof. Alex Snoeren for providing direction regularly at UCSD. I give my thanks to Prof. Bhaskar Rao for his help during my PhD. I really enjoyed working with Mohammad Naghshwar and Anders Plymoth whom I had an opportunity to work closely on various research projects. I thank my dissertation committee members, Professor Massimo, Prof. Bill Lin for their insightful comments and advice on the dissertation. I would like also thank Mr. Nathaneil Feldman for carefully going over my dissertation.

I thank my friends I made during my five years of stay at UCSD, Rohit Ramanujam, Jayadev Acharya, Hirakendu Das, Yogananda, Vipul Bhargava, Saurabh Joshi, Anand, Shue-Shue Tan, Bakhtiyar Neymano, Nikhil Karamchandani, Rathinakumar Appuswamy, for providing me with an enjoyable research and learning environment.

Finally, and most importantly, I would like to thank my wife Akshaya. Her support, encouragement, quiet patience and unwavering love were undeniably the base upon which the past few years of my life have been built. I thank my parents for their faith in me and allowing me to be as ambitious as I wanted.

Chapter 2 is adapted from A. Bhorkar, M. Naghshvar, T. Javidi and B. Rao, An Adaptive Opportunistic Routing Scheme for Wireless Ad-hoc Networks,, IEEE Transaction on Networking, 2012.

Chapter 3 is adapted from A. Bhorkar, T. Javidi, No regret routing in wireless Ad-hoc networks, Asilomar 10.

Chapter 4 is adapted from A. Bhorkar, M. Naghshvar, T.Javidi, Opportunistic routing for congestion diversity in wireless Ad-hoc networks, submitted to Transactions on Networking.

Chapter 5 is adapted from A. Bhorkar, T. Javidi, A. Snoeren, Achieving congestion diversity in wireless Ad-hoc networks, Infocom 11

VITA

2006	B. Tech in Electrical Engineering, Indian Institute of Technology, Bombay, India
2006	M.tech in Telecommunications and Signal Processing, Indian Institute of Technology, Bombay, India
2012	Ph. D. in Electrical Engineering (Communication Theory & Systems), University of California, San Diego

PUBLICATIONS

A. Bhorkar, M. Naghshvar, T.Javidi, Opportunistic routing for congestion diversity in wireless Ad-hoc networks, submitted to Transactions on Networking.

A. Bhorkar, T.Javidi, A. Snoeren, Achieving congestion diversity in wireless Ad-hoc networks,, in preparation.

A. Bhorkar, T. Javidi, A. Snoeren, Empirical Measurement of Draining Time, Expected Transmission (ETX), and Packet Erasure Probability in WiFi-based Mesh Networks, WinMEE 11

A. Bhorkar, T. Javidi, A. Snoeren, Achieving congestion diversity in wireless Ad-hoc networks, Infocom 11

A. Bhorkar, T. Javidi, No regret routing in wireless Ad-hoc networks, Asilomar 10

A. Bhorkar, T. Javidi, A. Snoeren, On practical implementation for congestion diversity in wireless Ad-hoc networks, Allerton 10

A. Bhorkar, A. Nilson, P. Johansoon, Common Opportunistic Routing and Forwarding, VTC10

A. Bhorkar, M. Naghshvar, T. Javidi and B. Rao, An Adaptive Opportunistic Routing Scheme for Wireless Ad-hoc Networks,, IEEE Transaction on Networking, 2012.

A. Bhorkar, M. Naghshvar, T. Javidi and B. Rao, AdaptOR An Adaptive Opportunistic Routing Scheme for Wireless Ad-hoc Networks, ISIT 09

A. Bhorkar, M. Naghshvar, T. Javidi and B. Rao Exploration vs Exploitation in wireless Ad-hoc networks, CDC 09

E. Coviello, A. Bhorkar, F. Rossetto, B. D. Rao, M. Zorzi, A Robust Approach to Carrier Sense for MIMO Ad Hoc Networks, ICC 2009

A. Bhorkar, B. S. Manoj, B. D. Rao, R. R. Rao, Antenna Selection Diversity Based MAC Protocol for MIMO Ad Hoc Wireless Networks, GLOBECOM 2008: 671-676

A. Bhorkar, A. Karandikar, V.S. Borkar, Opportunistic Power Optimal Scheduling IEEE GLOBECOM 06

H. Rath, A. Bhorkar, V. Sharma, An Opportunistic DRR (O-DRR) Uplink Scheduling Scheme for IEEE 802.16-based Broadband Wireless Networks, IETE, International Conference on Next Generation Networks (ICNGN), February 9, 2006, Mumbai.

H.Rath, A. Bhorkar, V. Sharma, An Opportunistic Uplink Scheduling Scheme to Achieve Bandwidth Fairness and Delay for Multiclass Traffic in WiMax (IEEE 802.16) Broadband Wireless Networks, IEEE GLOBECOM, 2006

N. Salodkar, A. Bhorkar, A. Karandikar, V. S. Borkar, An on-line learning algorithm for energy efficient delay constrained scheduling over a fading channel, IEEE Journal on Selected Areas in Communications 26(4): 732-742 (2008)

ABSTRACT OF THE DISSERTATION

Multi-hop Routing for Wireless Mesh Networks

by

Abhijeet Bhorkar

Doctor of Philosophy in Electrical Engineering (Communication Theory & Systems)

University of California, San Diego, 2012

Professor Tara Javidi, Chair

Wireless Mesh networks have the potential to provide inexpensive and quick access to the internet for military communications, surveillance, education, healthcare and disaster management. This work caters to the growing high-bandwidth demands by providing low delay and high throughput by designing efficient, robust routing algorithms for wireless mesh networks. Chapters 2 and 3 of this dissertation describe adaptive routing algorithms that opportunistically route the packets in the absence of reliable knowledge about channel statistics and the network model. We design two adaptive routing algorithms, Distributed Opportunistic Routing (d-AdaptOR) and No Regret Routing (NRR), which minimize the expected number of transmissions and thus improving the throughput. The remainder of the dissertation concerns with the design routing algorithms to avoid congestion in the network. In Chapter 4, we describe a

Distributed Opportunistic Routing algorithm with Congestion Diversity (ORCD) which employs receiver diversity and minimizes end-end delay. In Chapter 5, we present the Congestion Diversity Protocol (CDP), a distributed routing protocol for 802.11-based multi-hop wireless networks that combines important aspects of shortest-path and back-pressure routing to achieve improved end-end delay performance. This work reports on a practical (hardware and software) implementation of CDP in an indoor Wi-Fi testbed.

Chapter 1

Introduction

1.1 Routing in Wireless Mesh Networks

A wireless mesh network (WMN) is a network of nodes connected by wireless communication links. Wireless Mesh Networks is envisioned as a promising wireless technology for numerous applications, e.g. broadband home networking, community and enterprise networking, public Internet access, and so on [2]. In WMN, the wireless clients are connected to a mesh backbone for network access. The wireless mesh backbone typically consists of mesh routers to transport the traffic of wireless client to the internet using gateways chosen among mesh routers. Wireless mesh backbone can be rapidly deployed with minimal cost and provides a robust, efficient, reliable, and flexible system that supports the network access for mesh clients.

Due to the limited transmission range of the wireless nodes, many pairs of nodes in WMN may not be able to communicate directly; hence they may need other relay nodes to forward packets for them. The purpose of routing is generally to find paths from the source to the destination, maintain or update paths when the topology or link quality changes, and forward packets along the paths. The development of routing algorithms for WMN has received a tremendous amount of attention from researchers in recent years. The routing protocols in MWNs can be classified into four different categories using different notions of relay selection and the forwarding mechanism at the Media access Control layer as shown in Fig. 1.1.

Most conventional routing strategies in wireless mesh networks are typically

(I) Conventional unicast MAC Congestion unaware	(II) Opportunistic MAC Congestion unaware
(III) Conventional unicast MAC Congestion aware	(IV) Opportunistic MAC Congestion aware

Figure 1.1: Classification of routing protocols

time invariant and do not change the routing path [3–5] belong to type I in Fig. 1.1. Furthermore, these routing strategies employ a conventional scheme at the MAC layer where the each packets next hop is determined apriori before the actual transmission occurs. The shortest path routing algorithms [3–5] attempt to decrease the number of transmissions required to relay a packet to its destination. For example, SRCR [6] extends shortest path routing by selecting paths based upon the expected number of transmission attempts (ETX [3]). However, such fixed path schemes fail to take advantage of the broadcast nature and opportunities provided by the wireless medium and result in unnecessary packet retransmissions. Recently, Opportunistic routing algorithms have been developed in order to overcome the deficiencies of conventional routing [7–12]. The opportunistic routing decisions, in contrast, are made in an online manner by choosing the next relay based on the actual transmission outcome as well as on a rank ordering of neighboring nodes. Opportunistic routing mitigates the impact of poor wireless links by exploiting the broadcast nature of wireless transmissions and the path diversity. The authors in [7], [12] provided a Markov decision theoretic formulation for opportunistic routing. In particular, it is shown that the optimal routing decision at any epoch is to select the next relay node based on a distance-vector summarizing the expected-cost-to-forward from the neighbors to the destination. This “distance” is shown to be computable in a distributed manner and with low complexity using the probabilistic description of wireless links. The studies in [7], [12] provided a unifying framework for almost all versions of opportunistic routing such as Selection Diversity Forwarding SDF [8], Geographic Random Forwarding (GeRaF) [9], and ExOR [10], where the variations in [8–10] are due to the authors’ choices of which cost measures to optimize. For instance an optimal route in the context of ExOR [10] is computed so as to minimize the expected number of transmissions (ETX), while GeRaF [9] uses the shortest geographical distance from the destination as a criterion for selecting the next-hop. These routing algorithms belong to class II in Fig. 1.1 which are congestion unaware and use opportunistic MAC.

When multiple streams of packets are to traverse the network, however, it might be necessary to route some packets along longer paths, if these paths eventually lead to links that are less congested. More precisely, and as noted in [1, 13], the opportunistic routing schemes in [3–5, 8–12] can potentially cause severe congestion and unbounded delays (see examples given in [1]). In contrast, it is known that a simple routing policy, known as backpressure [14], ensures bounded expected total backlog for all stabilizable arrival rates. In the opportunistic context, diversity backpressure routing (DIVBAR) [13] provides an opportunistic generalization of backpressure which incorporates the wireless diversity. To ensure throughput optimality, backpressure-based algorithms [13, 14] do something very different from [8–12]; rather than any metric of closeness (or cost) to the destination, they choose the receiver with the largest positive differential backlog (routing responsibility is retained by the transmitter if no such receiver exists). This very property of ignoring the cost to the destination, however, becomes the bane of this approach, leading to poor delay performance (see [1, 13]). These localized and topology independent routing decisions are also used in various throughput optimal routing policies [14–20] distributing the traffic locally and resulting in large queue length. In [13], elements of shortest path computations are used to arrive at an enhanced version of DIVBAR (E-DIVBAR) in an attempt to mitigate the shortcomings of the two approaches. When choosing the next relay among the set of potential forwarders, E-DIVBAR considers the sum of the differential backlog and the expected hop-count to the destination (also known as ETX). However, and as shown in [1], E-DIVBAR does not guarantee a better delay performance than DIVBAR.

1.2 Dissertation Overview

The opportunistic algorithms proposed in [7–12] depend on a precise probabilistic model of wireless connections and the local topology of the network. We refer to these solutions as genie aided solutions. In a practical setting, however, these probabilistic models have to be "learned" and "maintained." In other words, a comprehensive study and evaluation of any opportunistic routing scheme requires an integrated approach to the issue of probability estimation. The authors in [21] provide a sensitivity analysis for the opportunistic routing algorithm given in [12]. By and large, however, the question of learning or estimating channel statistics in conjunction with opportunistic routing remains unexplored. In chapters 2 and 3, we investigate the problem of opportunistically routing packets in a wireless multi-hop network when zero or erroneous knowledge of transmission success probabilities and the network topology is available. In Chapter 2, we propose an distributed adaptive opportunistic routing algorithm (d-AdaptOR). This scheme is shown to be optimal with respect to an expected average per packet reward criterion. In particular, d-AdaptOR achieves an optimal per packet expected cost even when zero knowledge of the transmission success probabilities and network topology is available. The optimality of d-AdaptOR in terms of its per packet expected cost criterion ensures that the accumulated loss relative to the genie aided routing algorithm (known as *regret*) grows sub-linearly based on the number of packets in the network. In Chapter 3, we consider a stronger notion of the optimality criterion and account for the rate of learning with respect to regret (accumulated loss relative to the genie aided solution). In particular, we devise an opportunistic routing scheme NRR and show that it achieves a minimum rate of growth. These class of algorithms belong to type II in Fig. 1.1.

In Chapter 4, we present a distributed opportunistic routing policy with congestion diversity (D-ORCD) under which the congestion information is integrated with the distributed shortest path computations of [12]. The main contribution of this chapter is to design a distributed routing policy that improves the delay performance over existing routing policies while ensuring throughput optimality for opportunistic routing. We tackle some of the system level issues observed in realistic settings via detailed Qualnet simulations. We then show that D-ORCD exhibits better delay performance than state of the art routing policies, namely, EXOR, DIVBAR and E-DIVBAR. D-ORCD employes an opportunistic MAC and is time variant based on the congestion in the network; thus belong to type IV in Fig. 1.1.

In Chapter 5, we present and report on a practical implementation and testbed evaluation of a set of congestion aware routing protocols in 802.11 based multihop wireless networks: and Backpressure (BP), Enhanced Backpressure protocol (E-BP), Congestion Diversity Protocol (CDP). This work is based on extensions of D-ORCD, E-DIVBAR and DIVBAR to 802.11 wireless networks. These class of algorithms belong to type III in Fig. 1.1. This chapter deals with practical issues on the utilizing congestion diversity and compares the performance of various algorithms against benchmark shortest path routing algorithm SRCR. Under shortest path model, under-utilization of path diversity can result in increased delay, congestion, or even queue instability and buffer overflow. In contrast, the other class of algorithms, exemplified by the well known Backpressure Protocol (BP) [14] uses differential backlogs at nodes to make routing decisions. CDP and E-BP build upon these classes of existing multi-hop routing protocols which combine the backlog information and the expected hop-count to the destination.

A brief summary and some concluding remarks are made in Chapter 6.

Chapter 2

Adaptive Opportunistic Routing protocol

Abstract

In this work, a distributed adaptive opportunistic routing scheme for multi-hop wireless ad-hoc networks is proposed. The proposed scheme utilizes a reinforcement learning framework to opportunistically route the packets even in the absence of reliable knowledge about channel statistics and the network model. This scheme is shown to be optimal with respect to an expected average per packet reward criterion.

The proposed routing scheme jointly addresses the issues of learning and routing in an opportunistic context, where the network structure is characterized by the transmission success probabilities. In particular, this learning framework leads to a stochastic routing scheme which optimally "explores" and "exploits" the opportunities in the network.

2.1 Introduction

In this chapter, we first investigate the problem of opportunistically routing packets in a wireless multi-hop network when zero or erroneous knowledge of transmission success probabilities and network topology is available. Using a reinforcement learning framework, we propose a distributed adaptive opportunistic routing algorithm (d-

AdaptOR) that minimizes the expected average per packet cost for routing a packet from a source node to a destination. This is achieved by sufficiently exploring the network using data packets and exploiting the best routing opportunities.

Our proposed reinforcement learning framework allows for a low complexity, low overhead, distributed asynchronous implementation. The significant characteristics of d-AdaptOR are that it is oblivious to the initial knowledge about the network, it is distributed, and it is asynchronous.

The main contribution of this chapter is to provide an opportunistic routing algorithm which 1) assumes no knowledge about the channel statistics and network, but 2) uses a reinforcement learning framework in order to enable the nodes to adapt their routing strategies, and 3) optimally exploits the statistical opportunities and receiver diversity. In doing so, we build on the Markov decision formulation in [12] and an important theorem in Q-learning proved in [22]. There are many learning-based routing solutions (both heuristic or analytically driven) for conventional routing in wireless and wired networks [23–28]. None of these solutions exploit the receiver diversity gain in the context of opportunistic routing. For the sake of completeness, however, we provide a brief overview of the existing approaches. The authors in [23–27] focus on heuristic routing algorithms which adaptively identify the least congested path in a wired network. If the network congestion, hence delay, were to be replaced by time-invariant quantities,[1] the heuristics in [23–26] would become a special case of d-AdaptOR in a network with deterministic channels and no receiver diversity. In this light, Theorem 1 in Section 2.4 provides analytic guarantees for the heuristics obtained in [23–26]. In [28] analytic results for ant routing are obtained in wired networks without opportunism. Ant routing uses ant-like probes to find paths of optimal cost such as expected hop count, expected delay, packet loss probability.[2] This dependence on ant-like probing represents a stark difference with our approach where d-AdaptOR relies solely on the data packet for exploration.

The rest of the chapter is organized as follows: In Section 2.2, we discuss the

[1]Delay and congestion are highly time-varying quantities.

[2]Here, we note that unlike congestion or instantaneous delay, the expected delay under a stable and stationary routing algorithm is indeed time-invariant and allows for a similar mathematically-sound treatment.

system model and formulate the problem. Section 2.3 formally introduces our proposed adaptive routing algorithm d-AdaptOR. We then state and prove the optimality theorem for d-AdaptOR in Section 2.4. In Section 2.5, we present the implementation details and practical issues of d-AdaptOR. We perform a simulation study of d-AdaptOR in Section 2.6. Finally, we conclude the chapter and discuss future work in Section 2.7.

2.2 System Model

We consider the problem of routing packets from a source node 0 to a destination node d in a wireless ad-hoc network of $d+1$ nodes denoted by the set $\Theta = \{0, 1, 2, \ldots, d\}$. The time is slotted and indexed by $n \geq 0$ (this assumption is not technically critical and is only assumed for ease of exposition). A packet indexed by $m \geq 1$ is generated at the source node 0 at time τ_s^m according to an arbitrary distribution with rate $\lambda > 0$.

We assume a fixed transmission cost $c_i > 0$ is incurred upon a transmission from node i. Transmission cost c_i can be considered to model the amount of energy used for transmission, the expected time to transmit a given packet, or the hop count when the cost is set to unity. We consider an opportunistic routing setting with no duplicate copies of the packets. In other words, at a given time only one node is responsible for routing any given packet. Given a successful packet transmission from node i to the set of neighbor nodes S, the next (possibly randomized) routing decision includes 1) retransmission by node i, 2) relaying the packet by a node $j \in S$, or 3) dropping the packet altogether. If node j is selected as a relay, then it transmits the packet at the next slot, while all other nodes $k \neq j, k \in S$, expunge that packet.

We define the termination event for packet m to be the event that packet m is either received at the destination or is dropped by a relay before reaching the destination. We denote this termination action by T. We define termination time τ_T^m to be the stopping time when packet m is terminated. We discriminate amongst the termination events as follows: We assume that upon the termination of a packet at the destination (successful delivery of a packet to the destination), a fixed and given positive delivery reward R is obtained, while no reward is obtained if the packet is terminated before it reaches the destination. Let r_m denote this random reward obtained at the termination time τ_T^m, i.e. either zero if the packet is dropped prior to reaching the destination node or R if the packet is received at the destination.

Let $i_{n,m}$ denote the index of the node which at time n transmits packet m, and accordingly let $c_{i_{n,m}}$ denote the cost of transmission (equal to zero if at time n packet m is not transmitted). The routing scheme can be viewed as selecting a (random) sequence of nodes $\{i_{n,m}\}$ for relaying packets $m = 1, 2, \ldots$.[3] As such, the expected average per packet reward associated with routing packets along a sequence of $\{i_{n,m}\}$ up to time N is:

$$J_N = \mathbf{E}\left[\frac{1}{M_N}\sum_{m=1}^{M_N}\left\{r_m - \sum_{n=\tau_s^m}^{\tau_T^m - 1} c_{i_{n,m}}\right\}\right], \tag{2.1}$$

where M_N denotes the number of packets terminated up to time N and the expectation is taken over the events of transmission decisions, successful packet receptions, and packet generation times.

Problem (P) Choose a sequence of relay nodes $\{i_{n,m}\}$ in the absence of knowledge about the network topology such that J_N is maximized as $N \to \infty$.

In the next section we propose d-AdaptOR algorithm which solves Problem **(P)**. The nature of the algorithm allows nodes to make routing decisions in a distributed, asynchronous, and adaptive manner.

Remark 1. The problem of opportunistic routing for multiple source-destination pairs, without loss of generality, can be decomposed to the single source-destination problem described above (Problem **(P)** is solved for each distinct flow).

2.3 Distributed Algorithm

Before we proceed with the description of d-AdaptOR, we provide the following notations. Let $\mathcal{N}(i)$ denote the set of neighbors of node i including node i itself. Let $\mathfrak{S}^i$ denote the set of potential reception outcomes due to a transmission from node $i \in \Theta$, i.e. $\mathfrak{S}^i = \{S : S \subseteq \mathcal{N}(i), i \in S\}$. We refer to $\mathfrak{S}^i$ as the state space for node i's

[3]Packets are indexed according to the termination order.

transmission. Furthermore, let $\mathfrak{S} = \cup_{i\in\Theta}\mathfrak{S}^i$. Let $A(S) = S \cup \{T\}$ denote the space of all allowable actions available to node i upon successful reception at nodes in S. Finally, for each node i we define a reward function on states $S \in \mathfrak{S}^i$ and potential decisions $a \in A(S)$ as

$$g(S,a) = \begin{cases} -c_a & \text{if } a \in S \\ R & \text{if } a = T \text{ and } d \in S \\ 0 & \text{if } a = T \text{ but } d \notin S \end{cases}.$$

2.3.1 Overview of d-AdaptOR

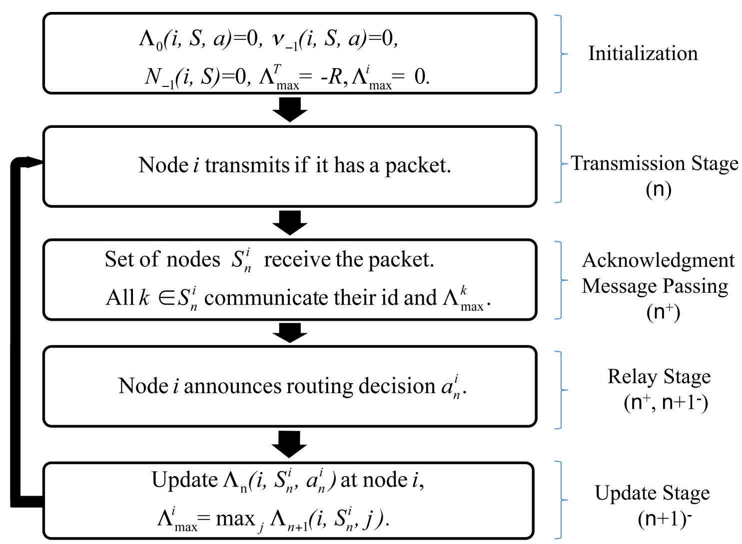


Figure 2.1: Flow of the algorithm.

As discussed before, the routing decision at any given time is made based on the reception outcome and involves retransmission, choosing the next relay, or termination. Our proposed scheme makes such decisions in a distributed manner via the following three-way handshake between node i and its neighbors $\mathcal{N}(i)$.

1. At time n node i transmits a packet.

2. The set of nodes S_n^i who have successfully received the packet from node i, transmit acknowledgement (ACK) packets to node i. In addition to the node's identity,

the acknowledgement packet of node $k \in S_n^i$ includes a control *message* known as *estimated best score* (EBS) and denoted by Λ_{max}^k.

3. Node i announces node $j \in S_n^i$ as the next transmitter or announces the termination decision T in a forwarding (FO) packet.

The routing decision of node i at time n is based on an adaptive (stored) score vector $\Lambda_n(i, \cdot, \cdot)$. The score vector $\Lambda_n(i, \cdot, \cdot)$ lies in space $\mathbb{R}^{v_i}$, where $v_i = \sum_{S \in \mathfrak{S}^i} |A(S)|$, and is updated by node i using the EBS messages Λ_{max}^k obtained from neighbors $k \in S_n^i$. Furthermore, node i uses a set of counting variables $\nu_n(i, S, a)$ and $N_n(i, S)$ and a sequence of positive scalars $\{\alpha_n\}_{n=1}^{\infty}$ to update its score vector at time n. The counting variable $\nu_n(i, S, a)$ is equal to the number of times neighbors S have received (and acknowledged) the packets transmitted from node i and routing decision $a \in A(S)$ has been made up to time n. Similarly, $N_n(i, S)$ is equal to the number of times that a set of nodes S have received (and acknowledged) packets transmitted from node i up to time n. Lastly, $\{\alpha_n\}_{n=1}^{\infty}$ is a fixed sequence of numbers available at all nodes.

Table 2.1 provides the notations used in the description of the algorithm, while Fig. 2.1 gives an overview of the components of the algorithm. Next we present further details.

2.3.2 Detailed description of d-AdaptOR

The operation of d-AdaptOR can be described in terms of initialization and four stages of transmission, reception and acknowledgement, relay, and adaptive computation as shown in Fig. 2.1. For simplicity of presentation we assume a sequential timing for each of the stages. We use n^+ to denote some (small) time after the start of n^{th} slot and $(n+1)^-$ to denote some (small) time before the end of n^{th} slot such that $n < n^+ < (n+1)^- < n+1$.

0. **Initialization**:

   For all $i \in \Theta, S \in \mathfrak{S}^i, a \in A(S)$, initialize

$$\Lambda_0(i, S, a) = \nu_{-1}(i, S, a) = N_{-1}(i, S) = \Lambda_{max}^i = 0, \text{ while } \Lambda_{max}^T = -R.$$

Table 2.1: Notations used in the description of the algorithm

Symbol	Definition
S_n^i	Nodes receiving the transmission from node i at time n
a_n^i	Decision taken by node i at time n
$A(S)$	Set of available actions when nodes in S receive a packet
$\mathcal{N}(i)$	Neighbors of node i including node i
$g(S,a)$	Reward obtained by taking decision a when the set S of nodes receive a packet
$\nu_n(i,S,a)$	Number of times up to time n, nodes S have received a packet from node i and decision a is taken
$N_n(i,S)$	Number of times up to time n, nodes S have received a packet from node i
$\Lambda_n(i,S,a)$	Score for node i at time n, when nodes S have received the packet and decision a is taken
Λ_{max}^i	Estimated best score for node i

1. **Transmission Stage**:

   Transmission stage occurs at time n in which node i transmits if it has a packet.

2. **Reception and acknowledgement Stage**:

   Let S_n^i denote the (random) set of nodes that have received the packet transmitted by node i. In the reception and acknowledgement stage, successful reception of the packet transmitted by node i is acknowledged by all the nodes in S_n^i. We assume that the delay for the acknowledgement stage is small enough (not more than the duration of the time slot) such that node i infers S_n^i by time n^+.

   For all nodes $k \in S_n^i$, the ACK packet of node k to node i includes the EBS message Λ_{max}^k.

   Upon reception and acknowledgement, the counting random variable N_n is incremented as follows:

$$N_n(i,S) = \begin{cases} N_{n-1}(i,S)+1 & \text{if } S = S_n^i \\ N_{n-1}(i,S) & \text{if } S \neq S_n^i \end{cases} .$$

3. **Relay Stage**:

   Node i selects a routing action $a_n^i \in A(S_n^i)$ according to the following (randomized) rule parameterized by $\epsilon_n(i,S) = \frac{1}{N_n(i,S)+1}$:

   - with probability $(1 - \epsilon_n(i, S_n^i))$,

$$a_n^i \in \arg\max_{j \in A(S_n^i)} \Lambda_n(i, S_n^i, j)$$

     is selected,[4]

   - with probability $\epsilon_n(i, S_n^i)$,

$$a_n^i \in A(S_n^i)$$

     is selected uniformly with probability $\frac{\epsilon_n(i,S_n^i)}{|A(S_n^i)|}$.

[4] In case of ambiguity, the node with the smallest index is chosen.

Node i transmits FO, a control packet which contains information about routing decision a_n^i at some time strictly between n^+ and $(n+1)^-$. If $a_n^i \neq T$, then node a_n^i prepares for forwarding in next time slot, while nodes $j \in S_n^i, j \neq a_n^i$ expunge the packet. If termination action is chosen, i.e. $a_n^i = T$, all nodes in S_n^i expunge the packet.

Upon selection of a routing action, the counting variable ν_n is updated.

$$\nu_n(i,S,a) = \begin{cases} \nu_{n-1}(i,S,a)+1 & \text{if } (S,a) = (S_n^i, a_n^i) \\ \nu_{n-1}(i,S,a) & \text{if } (S,a) \neq (S_n^i, a_n^i) \end{cases} .$$

4. **Adaptive Computation Stage**:

   At time $(n+1)^-$, after being done with transmission and relaying, node i updates score vector $\Lambda_n(i,\cdot,\cdot)$ as follows:

   - for $S = S_n^i$, $a = a_n^i$,

   $$\begin{aligned} \Lambda_{n+1}(i,S,a) &= \Lambda_n(i,S,a) + \alpha_{\nu_n(i,S,a)} \\ &\times \bigg(-\Lambda_n(i,S,a) + g(S,a) + \Lambda_{max}^a \bigg), \end{aligned} \tag{2.2}$$

   - otherwise,

   $$\Lambda_{n+1}(i,S,a) = \Lambda_n(i,S,a). \tag{2.3}$$

   Furthermore, node i updates its EBS message Λ_{max}^i for future acknowledgements as:

   $$\Lambda_{max}^i = \max_{j \in A(S_n^i)} \Lambda_{n+1}(i, S_n^i, j).$$

2.3.3 Computational issues

The computational complexity and control overhead of d-AdaptOR is low.

Complexity

To execute stochastic recursion (2.2), the number of computations required per packet is of the order of $O(\max_{i\in\Theta} |\mathcal{N}(i)|)$ at each time slot. The space complexity of d-AdaptOR is exponential in the number of neighbors i.e. $O(\max_{i\in\Theta} 2^{|\mathcal{N}(i)|})$ for each node. The reduction in storage requirement using approximation techniques in [29] is left as future work.

Control overhead

The number of acknowledgements per packet is order of $O(\max_{i\in\Theta} |\mathcal{N}(i)|)$, independent of network size.

Exploration overhead

The adaptation to the optimal performance in the network is guaranteed via a controlled randomized routing strategy which can be viewed as a cost of exploration. The cost of exploration is proportional to the total number of packets whose routes deviate from the optimal path. In proof of Theorem 1, we show that this cost increases sublinearly with the number of delivered packets, hence the per packet exploration cost diminishes as the number of delivered packets grow. Additionally, communication of Λ_{max} adds a very modest overhead to the genie-aided or greedy-based schemes such as ExOR or SR.

2.4 Analytic Optimality of d-AdaptOR

We will now state the main result establishing the optimality of the proposed d-AdaptOR algorithm under the assumptions of a time-invariant model of packet reception and reliable control packets. More precisely, we have the following assumptions.

Assumption 1. *The probability of successful reception of a packet transmitted by node i at set $S \subseteq \mathcal{N}(i)$ of nodes is $P(S|i)$, independent of time and all other routing decisions.*

The probabilities $P(\cdot|\cdot)$ in Assumption 1 characterize a packet reception model which we refer to as a *local broadcast model*. Note that for all $S \neq S'$, successful reception at S and S' are mutually exclusive and $\sum_{S\subseteq\Theta} P(S|i) = 1$. Furthermore, logically node i is always a recipient of its own transmission, i.e. $P(S|i) = 0$ iff $i \notin S$.

Assumption 2. *The successful reception at set S due to transmission from node i is acknowledged perfectly to node i.*

Remark 2. Assumption 1 is in line with the experimentally tested state of the art routing protocols MORE [30] and ExOR [10]. These studies seem to indicate that reasonably simple probabilistic models provide good abstractions of media access control (MAC) and physical (PHY) layers at the routing layer.

Remark 3. In practice, Assumption 2 is hard to satisfy. But as we will see in Section 2.6, when the rates and power of the control packets are set to maximize the reliability, the impact of violating this assumption can be kept extremely low.

Remark 4. In Sections 2.6, we address the severity as well as the implications of Assumptions 1 and 2. In particular, via a set of QualNet simulations, we will show that d-AdaptOR exhibits many of its desirable properties in a realistic setup despite the relaxation of the analytical assumptions.

Given Assumptions 1 and 2, we are almost ready to present Theorem 1 regarding the optimality of d-AdaptOR among the class of policies that are oblivious to the network topology and/or channel statistics. More precisely, let a *distributed routing policy* be a collection $\phi = \{\phi^i\}_{i\in\Theta}$ of routing decisions taken at nodes $i \in \Theta$, where ϕ^i denotes a sequence of random actions $\phi^i = \{a_0^i, a_1^i, \ldots\}$ for node i. The policy ϕ is said to be (P)-*admissible* if for all nodes $i \in \Theta$, $S \in \mathfrak{S}^i$, $a \in A(S)$, the event $\{a_n^i = a\}$ belongs to the σ-field $\mathcal{H}_n^i$ generated by the observations at node i, i.e. $\bigcup_{j\in\mathcal{N}(i)}\{S_0^j, a_0^j, \ldots, S_{n-1}^j, a_{n-1}^j, S_n^j\}$. Let Φ denote the set of such (P)-admissible policies. Theorem 1 below states that d-AdaptOR, denoted by $\phi^* \in \Phi$ is an optimal (P)-admissible policy.

Theorem 1. *Suppose $\sum_{n=0}^{\infty}\alpha_n = \infty$, $\sum_{n=0}^{\infty}\alpha_n^2 < \infty$, and Assumptions 1 and 2 hold. Then for all $\phi \in \Phi$,*

$$\begin{aligned}
&\lim_{N\to\infty}\mathbf{E}^{\phi^*}\left[\frac{1}{M_N}\sum_{m=1}^{M_N}\left\{r_m - \sum_{n=\tau_s^m}^{\tau_T^m-1} c_{i_{n,m}}\right\}\right] \\
&\geq \limsup_{N\to\infty}\mathbf{E}^{\phi}\left[\frac{1}{M_N}\sum_{m=1}^{M_N}\left\{r_m - \sum_{n=\tau_s^m}^{\tau_T^m-1} c_{i_{n,m}}\right\}\right]
\end{aligned}$$

where $\mathbf{E}^{\phi^*}$ *and* $\mathbf{E}^{\phi}$ *are the expectations taken with respect to policies* ϕ^* *and* ϕ *respectively.*[5]

Next we prove the optimality of d-AdaptOR in two steps. In the first step, we show that Λ_n converges in an almost sure sense. In the second step we use this convergence result to show that d-AdaptOR is optimal for Problem (**P**).

2.4.1 Convergence of Λ_n

Let $U : \prod_i \mathbb{R}^{v_i} \to \prod_i \mathbb{R}^{v_i}$ be an operator on vector Λ such that,

$$(U\Lambda)(i, S, a) = g(S, a) + \sum_{S' \in \mathfrak{S}^a} P(S'|a) \max_{j \in A(S')} \Lambda(a, S', j). \tag{2.4}$$

Let $\Lambda^* \in \prod_i \mathbb{R}^{v_i}$ denote the fixed point of operator U,[6] i.e.

$$\Lambda^*(i, S, a) = g(S, a) + \sum_{S' \in \mathfrak{S}^a} P(S'|a) \max_{j \in A(S')} \Lambda^*(a, S', j). \tag{2.5}$$

The following lemma establishes the convergence of recursion (2.2) to the fixed point of U, Λ^*.

Lemma 1. *Let*

(J1) $\Lambda_0(\cdot, \cdot, \cdot) = 0$, $\Lambda^T_{max} = -R$, $\Lambda^i_{max} = 0$ *for all* $i \in \Theta$,

(J2) $\sum_{n=0}^{\infty} \alpha_n = \infty$, $\sum_{n=0}^{\infty} \alpha_n^2 < \infty$.

Then the sequence Λ_n *obtained by the stochastic recursion (2.2) converges to* Λ^* *almost surely.*

The proof uses known results on the convergence of a certain recursive stochastic process as presented by Fact 4 in Appendix A.1.1.

---

[5]This is a strong notion of optimality and implies that the proposed algorithm's expected average reward is greater than the best case performance ($\limsup$) of all policies [31, Page 344].

[6]Existence and uniqueness of Λ^* is provided in Appendix A.1.1.

2.4.2 Proof of optimality

Using the convergence of Λ_n we show that the expected average per packet reward under d-AdaptOR is equal to the optimal expected average per packet reward obtained for a genie-aided system where the local broadcast model is known perfectly. In other words, we take a cue from known results associated with a closely related Auxiliary Problem **(AP)**. In this Auxiliary Problem **(AP)**, there exists a centralized controller with full knowledge of the local broadcast model $P(\cdot|\cdot)$ as well as the transmission outcomes across the network [7, 12]. The objective in the Auxiliary Problem **(AP)** is a single packet variation of that in Problem **(P)**: the reward

$$\mathbf{E}\left[r_m - \sum_{n=0}^{\tau_T^m - 1} c_{i_{n,m}}\right]$$

for routing a single packet m from the source to the destination is maximized over a set Π of (AP)-*admissible* policies, where this set Π of (AP)-admissible policies is a superset of (P)-admissible policies Φ which also includes all topology-aware and centralized policies. This Auxiliary Problem **(AP)** has been extensively studied in [7,12,32], where a Markov decision formulation provides the following important result:

Fact 1. [12, Theorem 2.1] Consider the unique solution $V^* : \Theta \cup \{T\} \to \mathbb{R}^+$ to the following fixed point equation:

$$V^*(d) = R \tag{2.6}$$

$$V^*(i) = \max(\{-c_i + \sum_{S'} P(S'|i)(\max_{j \in S'} V^*(j))\}, 0) \tag{2.7}$$

There exists an optimal topology-aware and centralized admissible policy $\pi^* \in \Pi$ such that

$$\mathbf{E}^{\pi^*}\left[r_m - \sum_{n=0}^{\tau_T^m - 1} c_{i_{n,m}}\right] = V^*(0). \tag{2.8}$$

Lemma 2 below states the relationship between the solution of Problem **(P)** and that of the Auxiliary Problem **(AP)**. More specifically, Lemma 2 shows that $V^*(0)$ is an upper bound for the solution to Problem **(P)**.

Lemma 2. *For any (P)-admissible policy $\phi \in \Phi$ for Problem* **(P)** *and for all $N = 1, 2, \ldots$,*

$$\mathbf{E}^{\phi}\left[\frac{1}{M_N}\sum_{m=1}^{M_N}\left\{r_m - \sum_{n=\tau_s^m}^{\tau_T^m-1} c_{i_{n,m}}\right\}\right] \leq V^*(0).$$

The proof is given in Appendix A.1.2. Intuitively the result holds because the set of (P)-admissible policies is a subset of (AP)-admissible policies, i.e. $\Phi \subset \Pi$.

Lemma 3 below gives the achievability proof by showing that the expected average per packet reward of d-AdaptOR is lower bounded by $V^*(0)$.

Lemma 3. *For any $\delta > 0$,*

$$\liminf_{N\to\infty} \mathbf{E}^{\phi^*}\left[\frac{1}{M_N}\sum_{m=1}^{M_N}\left\{r_m - \sum_{n=\tau_s^m}^{\tau_T^m-1} c_{i_{n,m}}\right\}\right] \geq V^*(0) - \delta.$$

The proof is given in Appendix A.1.3. Lemmas 2 and 3 imply that ϕ^* (which is (P)-admissible by construction) is an optimal policy under which

$$\lim_{N\to\infty} \mathbf{E}^{\phi^*}\left[\frac{1}{M_N}\sum_{m=1}^{M_N}\left\{r_m - \sum_{n=\tau_s^m}^{\tau_T^m-1} c_{i_{n,m}}\right\}\right]$$

exists and is equal to $V^*(0)$, establishing the proof of Theorem 1.

Corollary 1. *When $c_i = 1$, $i \in \Theta$, the network is connected, and R is greater than the worst case routing cost,*[7] *d-AdaptOR minimizes*

$$D_N = \mathbf{E}^{\pi}\left[\frac{1}{M_N}\sum_{m=1}^{M_N}\{\tau_T^m - \tau_s^m)\}\right], \tag{2.9}$$

the expected per packet delivery time as $N \to \infty$.

This is because when $c_i = 1$, R is sufficiently large, and the network is connected

$$\begin{aligned} V^*(0) &= R - \inf_{\pi\in\Pi} \mathbf{E}^{\pi}\left[\frac{1}{M_N}\sum_{m=1}^{M_N}\left\{\sum_{n=\tau_s^m}^{\tau_T^m-1} c_{i_{n,m}}\right\}\right] \\ &= R - \inf_{\pi\in\Pi} D_N, \text{as } N \to \infty. \end{aligned}$$

[7] The worst case routing cost can be determined by taking the supremum over ETX metrics for all source-destination pairs.

2.5 Protocol Design and implementation issues

In this section, we describe an 802.11 compatible implementation for d-AdaptOR.

2.5.1 802.11 compatible implementation

The implementation of d-AdaptOR, analogous to any opportunistic scheme, involves the selection of a relay node among the candidate set of nodes that have received and acknowledged a packet successfully. One of the major challenges in the implementation of an opportunistic routing algorithm, in general, and d-AdaptOR algorithm in particular, is the design of an 802.11 compatible acknowledgement mechanism at the MAC layer. Below we propose a practical and simple way to implement an acknowledgement architecture.

The transmission at any node i is done according to 802.11 CSMA/CA mechanism. Specifically, before any transmission, transmitter i performs channel sensing and starts transmission after the backoff counter is decremented to zero. For each neighbor node $j \in \mathcal{N}(i)$, the transmitter node i then reserves a virtual time slot of duration $T_{ACK} + T_{SIFS}$, where T_{ACK} is the duration of the acknowledgement packet and T_{SIFS} is the duration of Short InterFrame Space (SIFS) [33]. Transmitter i then piggy-backs a priority ordering of nodes $\mathcal{N}(i)$ with each data packet transmitted. The priority ordering determines the virtual time slot in which the candidate nodes transmit their acknowledgement. Nodes in the set S^i that have successfully received the packet then transmit acknowledgement packets sequentially in the order determined by the transmitter node.

After a waiting time of $T_{wait} = |\mathcal{N}(i)|(T_{ACK} + T_{SIFS})$ during which each node in the set S^i has had a chance to send an ACK, node i transmits a FOrwarding control packet (FO). The FO packets contain the identity of the next forwarder, which may be node i again or any node $j \in S^i$. If T_{wait} expires and no FO packet is received (FO packet reception is unsuccessful), then the corresponding candidate nodes drop the received data packet. If the transmitter i does not receive any acknowledgement, node i retransmits the packet. The backoff window is doubled after every retransmission. Furthermore, the packet is dropped if the retry limit (set to 7) is reached.

In addition to the acknowledgement scheme, d-AdaptOR requires modifications to the 802.11 MAC frame format. Fig. 2.2 shows the modified MAC frame formats required by d-AdaptOR. The reserved bits in the type/subtype fields of the frame control field of the 802.11 MAC specification are used to indicate whether the rest of the frame is a d-AdaptOR data frame, a d-AdaptOR ACK, or a FO.[8] The data frame contains the candidate set in priority order, the payload, and the 802.11 Frame Check Sequence. The acknowledgement frame includes the data frame sender's address and the feedback EBS Λ_{max}. The FO packet is exactly the same as a standard 802.11 short control frame which uses a different subtype value.

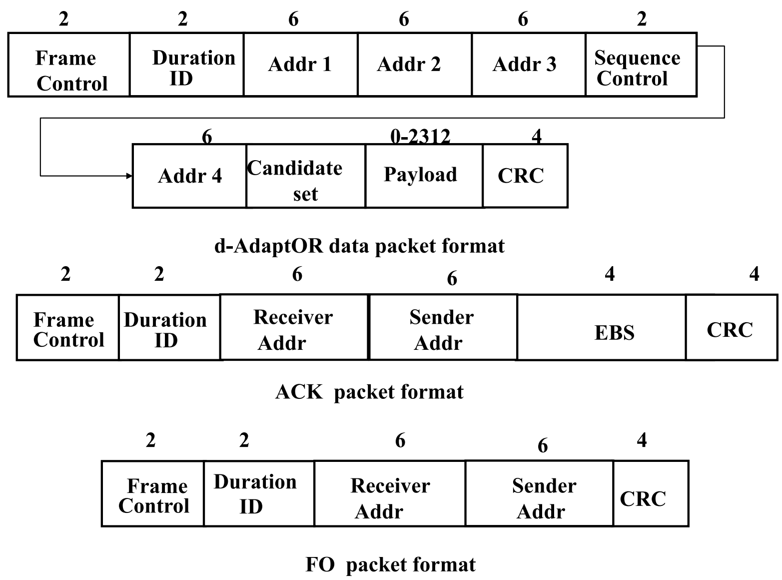


Figure 2.2: Frame structure of data packets, acknowledgement packets, and FO packets.

2.5.2 d-AdaptOR in a realistic setting

Loss of ACK and FO packets

Interference or low signal to noise ratio (SNR) can cause loss of ACK and FO packets. Loss of an ACK packet results in an incorrect estimation of nodes that have received the packet and thus affects the performance of the algorithm. Loss of FO packet negatively impacts the throughput performance of the network. In particular, loss of FO packet can result in the drop of a data packet at all the potential relays, reducing the

[8]This enables the d-AdaptOR to communicate and be fully compatible with other 802.11 devices.

throughput performance. Hence, in our design, FO packets are transmitted at lower rates to ensure a reliable transmission.

Increased overhead

As it is the case with any opportunistic scheme, d-AdaptOR adds a modest additional overhead to the standard 802.11 due to the added acknowledgement/handshake structure. This overhead increases linearly with the number of neighbors. Assuming a 802.11b physical layer operating at 11 Mbps with an SIFS time of 10 μs, preamble duration of 20 μs, Physical Layer Convergence Protocol (PLCP) header duration of 4 μs, and 512 byte frame payloads, Table 2.2 compares the overhead in the data packet due to piggy-backing and the control overhead due to ACK and FO packets for unicast 802.11, genie-aided opportunistic scheme, and d-AdaptOR. d-AdaptOR requires a communication overhead of 4 extra bytes (for EBS) per ACK packet compared to the genie-aided opportunistic scheme, while unicast 802.11 does not require such overhead.

Table 2.2: Overhead comparisons

	Data Frame	Control packets	Total
802.11	397 μs	40 μs (ACK)	437 μs
Genie-aided opportunistic scheme	400 μs	115 μs + 40 μs (ACK+FO)	555 μs
d-AdaptOR	400 μs	124 μs +40 μs (ACK+FO)	564 μs

Note that the overhead cost can be reduced by restricting the number of nodes in the candidate list of MAC header to a given number, MAX-NEIGHBOUR. The unique ordering for the nodes in the candidate set is determined by prioritizing the nodes with respect to $\Lambda_n(i, \{i, j\}, j), j \in \mathcal{N}(i)$ and then choosing the MAX-NEIGHBOUR highest priority nodes.[9] Such a limitation will sacrifice the diversity gain and hence, the performance of any opportunistic routing algorithm for lower overhead. In practice, we have seen that limiting the neighbor set to 4 provides most of the diversity gain.

[9]In case of ambiguity, the node with the smallest index is chosen.

2.6 Simulations

In this section, we provide simulation studies in realistic wireless settings where the theoretical assumptions of our study do not hold. These simulations not only demonstrate a robust performance gain under d-AdaptOR in a realistic network, but also provide significant insight of the appropriate choice of the design parameters such as damping sequence $\{\alpha_n\}$, delivery reward R, etc. We first investigate the performance of d-AdaptOR with respect to the design parameters and network parameters in a grid topology of 16 nodes. We then use a realistic topology of 36 nodes with random placement to demonstrate robustness of d-Adaptor in splite of the violation of the analytic Assumptions 1 and 2.

2.6.1 Simulation Setup

In Subsections 2.6.2 and 2.6.3 using appropriate choice of the design parameters, we compare the performance of d-AdaptOR against suitably chosen candidates. As a benchmark, when appropriate, we have compared the performance against a genie-aided policy which relies on full network topology information when selecting routes. This is nothing but π^* discussed in Subsection 2.4.2. We also compare the performance against Stochastic Routing (SR) [7] (SR is the distributed implementation of policy π^*) and ExOR [10] (an opportunistic routing policy with the ETX metric) in which the empirical probabilistic structure of the network is used to implement opportunistic routing algorithms. As a result, their performance will be highly dependent on the precision of empirical probability associated with link, p_{ij}. To provide a fair comparison, we have considered simple greedy versions of SR and ExOR. These algorithms adapt $\{p_{ij}\}$ to the history of packet reception outcomes and rely on the updates to make routing decisions assuming error free $\{p_{ij}\}$. We have also compared our performance against a conventional routing SRCR [6] with full knowledge of topology. In this setting, a conventional route is selected with perfect knowledge of link success probability at any given node. This comparison in effect provides a simple benchmark for all learning-based conventional routing policies in the literature such as Q-routing [23] and predictive Q-routing [25] when congestion is taken to be small enough (such that finding least congested paths coincides with finding the path with the minimum expected number of transmissions).

Our simulations are performed in QualNet. We consider two sets of topologies in our experimental study:

1. Grid Topology: In Subsection 2.6.2, we study a grid topology consisting of 16 indoor nodes such that the nearest neighbors are separated by distance L m. If unspecified, L is chosen to be 25 meters. The source and the destination are chosen at the maximal distance (on the diagonal) from each other.

2. Random Topology: In Subsection 2.6.3, we study a random topology consisting of 36 indoor nodes placed in an area of 150m$\times$150m. Here, we investigate the performance under a multi-source multi-destination setting as the number of flows in the network is varied and each flow is specified via a randomly selected pair of source and destination nodes.

The nodes are equipped with 802.11b radios placed in an indoor environment transmitting at 11 Mbps with transmission power of 15 dBm. Note that the choice of indoor environment is motivated by the findings in [34] where opportunistic routing is found to provide significant diversity gains. The wireless medium model includes Rician fading with the K-factor of 4 and Log-normal shadowing with a mean of 4dB. The path loss follows the two-ray model in [35] with a path exponent of 3. The acknowledgement packets are short packets of length 24 bytes transmitted at 11 Mbps, while FO packets are of length 20 bytes and transmitted at a lower rate of 1 Mbps to ensure reliability. If unspecified, packets are generated according to a constant bit rate (CBR) source at a rate of 20 packets/sec. The packets are assumed to be of length 512 bytes equipped with simple cyclic redundancy check (CRC) error detection. The cost of transmission is assumed to be one unit and the reward R is set to 40. We have chosen $\alpha_n^1 = \frac{1}{\sqrt{n}\log n}$ as the exploration parameter of choice.

2.6.2 Effects of Design and Network Parameters

In this subsection, we investigate the role and criticality of various design parameters of d-AdaptOR with respect to criterion of the expected number of transmissions.

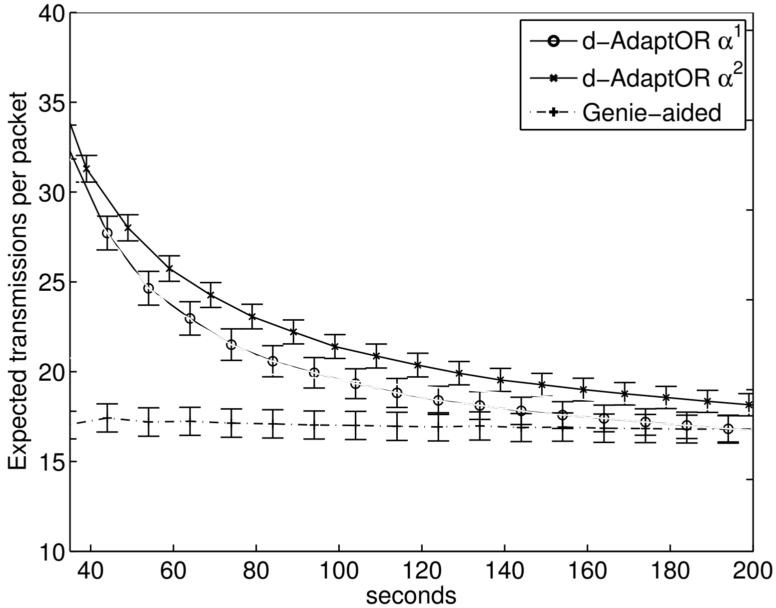


Figure 2.3: Comparison for $\alpha_n^1 = \frac{1}{\sqrt{n}\log(n)}$, $\alpha_n^2 = \frac{1}{n\log(n)}$.

Let us start with design parameters $\{\alpha_n\}$ and R

Exploration Parameter Sequence α_n

The convergence rate of stochastic recursion (2.2) depends strongly on the choice of sequence $\{\alpha_n\}$. Convergence is slower with a faster decreasing sequence $\{\alpha_n\}$ and results in less variance in the estimates of Λ_n, while with a slow decreasing sequence of $\{\alpha_n\}$, convergence is fast but results in large variance in the estimates of Λ_n. In Fig. 2.3, we have plotted the effect of the choice of α_n sequence by comparing two sequences $\{\alpha_n^1 = \frac{1}{\sqrt{n}\log n}\}$ and $\{\alpha_n^2 = \frac{1}{n\log n}\}$. Note that under sequence $\{\alpha_n^2 = \frac{1}{n\log n}\}$, d-AdaptOR is slower to adapt to the optimal performance while it shows a slightly smaller variance. This is because the choice of α_n controls the rate with which greedy versus (randomly chosen) exploration actions are utilized. The optimization of the choice of $\{\alpha_n\}$ is an interesting topic of study in stochastic approximation [36]- [37], far beyond the scope of this work.

Per Packet Delivery Reward R

To ensure an acceptable performance of d-AdaptOR, the value of delivery reward, R, must be chosen sufficiently high. This would ensure the existence of routes under which the value of delivering a packet (as represented in R) is worth (i.e. larger than) the cost of relaying and routing that packet. A reasonable choice of R is any value larger than the worst case expected transmission cost. Increasing R beyond such a value does not affect the asymptotic optimality of the algorithm. Next, we study the performance of d-AdaptOR with respect to the convergence rate and delivery ratio.

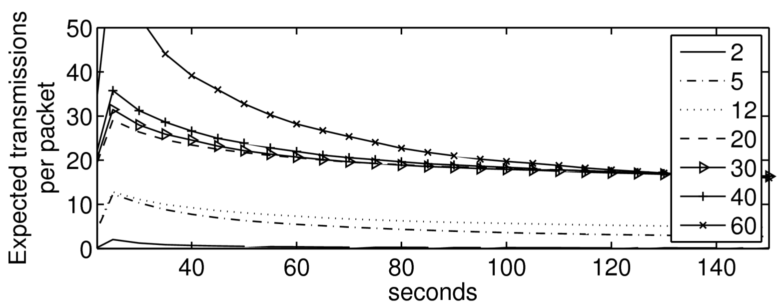


Figure 2.4: Expected number of transmissions versus time as R is varied.

Fig. 2.4 plots the expected number of transmissions rate as time progresses for various values of R. As seen in Fig. 2.4, if R increases beyond a threshold R_0 (in the example provided here, this threshold is 18, but in general it depends on the network diameter), the expected number of transmissions per packet achieve the optimal value of R_0. In contrast, for $R < R_0$, the expected number of transmissions approaches zero as the packets not worth obtaining a routing reward are dropped.[10] Fig. 2.4 also shows that the convergence rate of the expected number of transmissions for routing per packet under d-AdaptOR decreases as R increases. The slow convergence for $R > R_0$ for large R is due to the flexibility of exploring longer paths. The slow convergence to zero for $R < R_0$ near R_0 is attributed to the fact that it takes longer time for d-AdaptOR to realize that the packet is not worth relaying.

Fig. 2.5 plots the delivery ratio as R is varied. Fig. 2.5 shows that as R increases beyond a threshold R_0, the delivery ratio remains fixed. However, for sufficiently small

[10]For $R < R_0$, we have plotted negative of the expected per packet reward as the expected number of transmissions.

R, nearly all the packets are dropped as the cost of transmission of the packet as well as relaying is not worth the obtained delivery reward. Due to very slow convergence rate around R_0 for $R < R_0$, we observe that non-negligible number of packets are delivered in the duration of experiment.

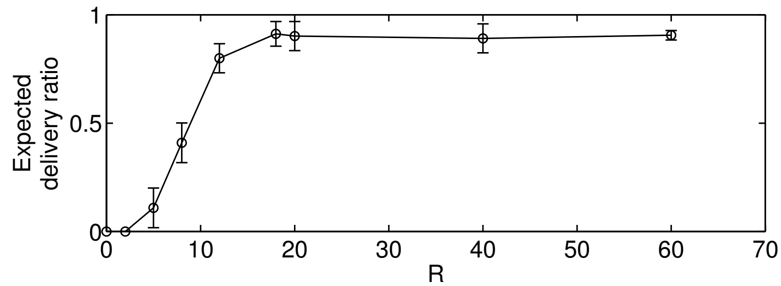


Figure 2.5: Delivery ratio as R is varied.

Next, we investigate the performance of d-AdaptOR with respect to other candidate protocols for the network parameters such as packet length, traffic rate, neighbor distance, and time varying costs.

Packet Length

We have repeated our simulations for 1024 byte packets. Fig. 2.6 plots the performance as the packet length is varied from 512 to 1024 bytes. Note that due to the decreasing packet transmission reliabilities, the expected routing cost per packet is increased with the packet size; however, the optimality of d-AdaptOR does not depend on the packet length.

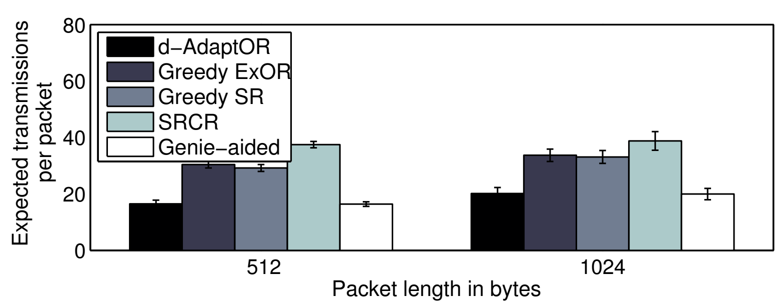


Figure 2.6: d-AdaptOR performance as packet length is varied.

Traffic rate

Fig. 2.7 plots the mean number of transmissions versus CBR rate for candidate algorithms. Even though the performance gain for d-AdaptOR decreases somewhat with increase in the load, there is always a non-negligible advantage over greedy solutions.

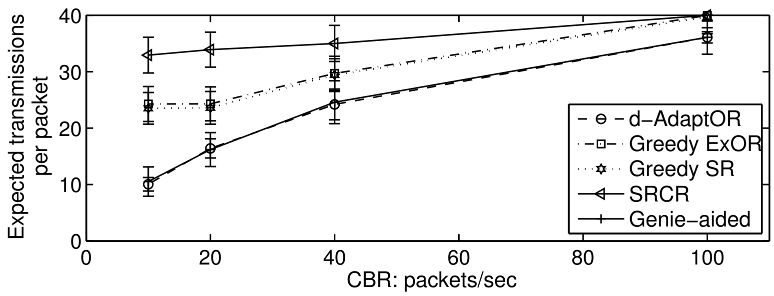


Figure 2.7: Performance of d-AdaptOR as CBR traffic is varied.

Average Hop Length L

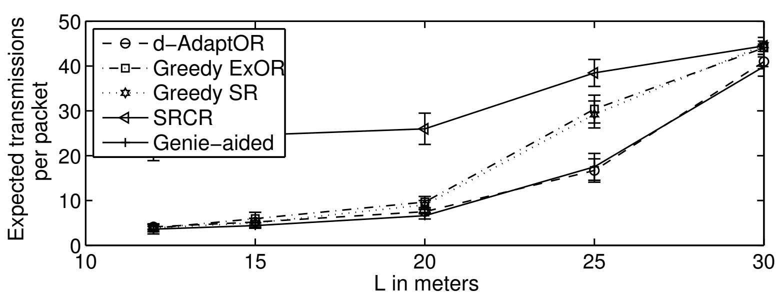


Figure 2.8: Small hops provide significant receiver diversity gain.

In an attempt to understand the performance gap between various opportunistic algorithms, specifically the gap between d-AdaptOR versus learning-based conventional routing algorithms [23–26] whose performance is bounded by SRCR, one needs to gain insight about the diversity gain achieved by opportunistic routing. Fig. 2.8 compares the expected transmission cost for the three opportunistic routing algorithms (d-AdaptOR, ExOR, and SR) and SRCR as the distance between the neighboring nodes in the grid topology, measured in L meters, is varied from 10 meters to 30 meters. Note that for high values of L, the receiver diversity is low due to retransmission packet losses giving nearly similar performance for candidate protocols, while small L corresponds to a network with large receiver diversity gain. As expected, when L is small, all opportunistic routing schemes provide a significant improvement over conventional routing, but perhaps what is more interesting is the performance gain of learning-based d-AdaptOR over the greedy-based solutions in medium ranges.

Time Varying Cost

In our analytical setup we assume the transmission costs are fixed. Next, we discuss a simple scenario where the nodes have time varying transmission costs. Consider a network in which nodes may go into an energy saving mode when they do not participate in routing (e.g. to recharge their energy sources). Assume that upon entering the energy saving mode, a node announces a high cost of transmission (100 instead of usual transmission cost of 1). Fig. 2.9 plots the expected average cost of d-AdaptOR when two nodes at the center of the grid move into an energy saving mode. It shows that d-AdaptOR can track the genie aided solution after the nodes move into the energy saving mode.

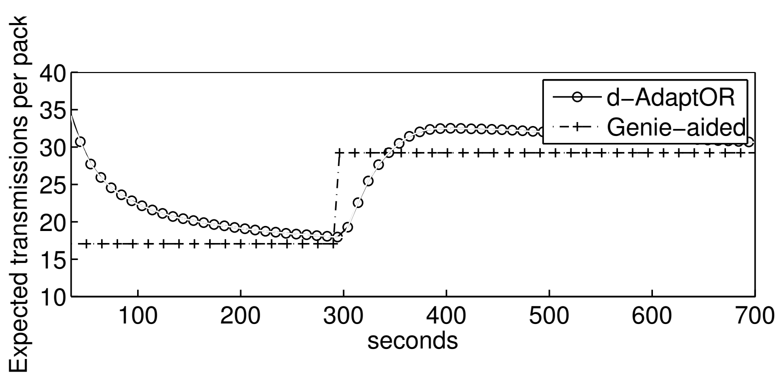


Figure 2.9: Time varying cost: Nodes go into sleep mode at time 300 seconds.

2.6.3 Case Study: Random Network

In this subsection, we study a random network scenario consisting of 36 wireless nodes placed randomly, with the remaining parameters kept the same as the default parameters.

Fig. 2.10 plots the expected number of transmissions and the expected average per packet reward for the candidate routing algorithms versus network operation time when a single flow is present in the random topology. We first note that, as expected, SRCR performs poorly compared to the opportunistic schemes as it fails to utilize the receiver diversity gain. This underlines our contribution over all existing learning-based solutions [23–26] which ignore receiver diversity. Furthermore, Fig. 2.10 shows that d-AdaptOR algorithm outperforms the greedy opportunistic schemes given sufficient number of packet deliveries. This is because the greedy versions of SR and ExOR fail to explore possible choices of routes and often result in strictly suboptimal routing policies. Fig. 2.10 also shows that the randomized routing decisions employed by d-AdaptOR work as a double-edged sword. On the one hand, they form a mechanism through which network opportunities are exhaustively explored until the globally optimal decisions are constructed, resulting in an improved long term performance while these randomized decisions lead to a short-term performance loss. This in fact is reminiscent of the well-known exploration/exploitation trade-off in stochastic control and learning literature.

Next we study the performance of d-AdaptOR as the number of flows in the network is varied where each flow is specified via a randomly selected pair of source and destination. Fig. 2.11 plots the expected number of transmissions and expected average reward for the candidate routing algorithms for the random topology. As seen in Fig. 2.11, d-AdaptOR maintains an optimal performance. However, Fig. 2.11 also shows that the gap between d-AdaptOR and the greedy version of SR significantly decreases with an increase in number of flows where the natural pattern of traffic flow renders the (randomized) exploration phase less critical. In other words, while Fig. 2.11 is consistent with the Remark 1 in Section 2.2 regarding the decomposition of multiple-flow scenario to multiple single-flow scenarios, it also suggests that a joint design in which the multiplicity of flows provide a natural (and greedy) exploration of the network might be beneficial with regard to the transient/short-term performance measures of interest.

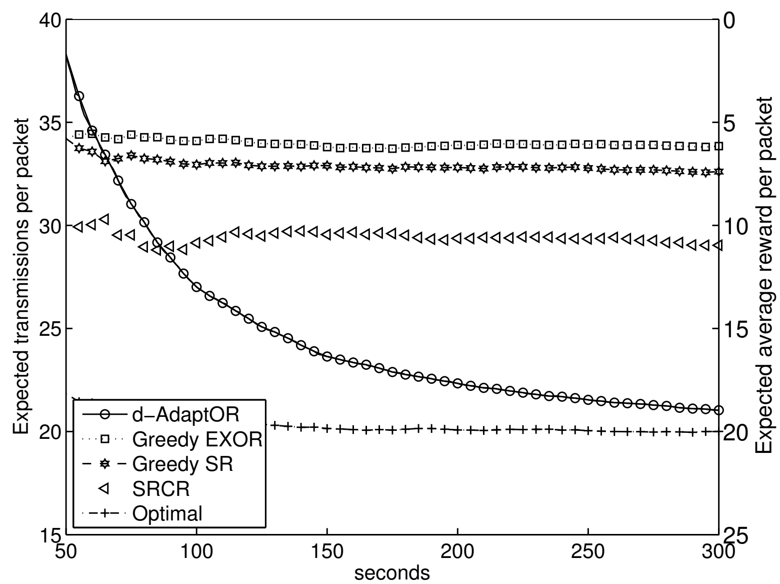


Figure 2.10: The expected number of transmissions as function of operation time.

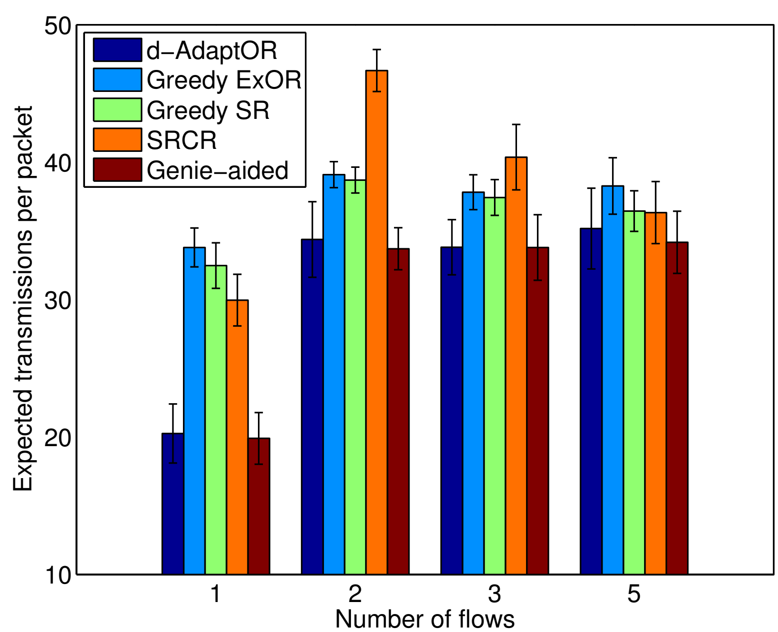


Figure 2.11: d-AdaptOR vs. distributed SR, ExOR, and SRCR performance.

2.7 Summary

In this chapter, we proposed d-AdaptOR, a distributed, adaptive, and opportunistic routing algorithm whose performance is shown to be optimal with zero knowledge regarding network topology and channel statistics. More precisely, under idealized assumptions, d-AdaptOR is shown to achieve the performance of an optimal routing with perfect and centralized knowledge about network topology, where the performance is measured in terms of the expected per packet reward. Furthermore, we show that d-AdaptOR allows for a practical distributed and asynchronous 802.11 compatible implementation, whose performance was investigated via a detailed set of QualNet simulations under practical and realistic networks. Simulations show that d-AdaptOR consistently outperforms existing adaptive routing algorithms in practical settings.

Acknowledgement

This chapter, in part, appears in the following publications. The dissertation author was the primary investigator and author of these papers.

- A. Bhorkar, M. Naghshvar, T. Javidi and B. Rao, An Adaptive Opportunistic Routing Scheme for Wireless Ad-hoc Networks,, IEEE Transaction on Networking, volume 12, issue 1, 2012.

- A. Bhorkar, M. Naghshvar, T. Javidi and B. Rao, AdaptOR An Adaptive Opportunistic Routing Scheme for Wireless Ad-hoc Networks, ISIT 09.

- A. Bhorkar, M. Naghshvar, T. Javidi and B. Rao Exploration vs Exploitation in wireless Ad-hoc networks, CDC 09.

I would like to thank my co-authors Mohammad Naghshvar, Prof. Tara Javidi, and Prof. Bhaskar Rao.

Chapter 3

No Regret Routing protocol

Abstract

In this chapter, we consider the problem of adaptive routing in wireless ad-hoc networks with respect to the regret (or learning loss) criterion. The regret criterion accounts for the loss of performance incurred because of the implicit on-line learning task involved when the link probabilities are unknown. In particular, we construct asymptotically efficient adaptive control schemes (that minimize the rate at which regret accumulates with time) while coming arbitrarily close to the minimum rate. The intent is to capture the conflict between learning and control.

3.1 Introduction

The opportunistic algorithms proposed in [7,9,10,12] depend on a precise probabilistic model of wireless connections and the network topology. We refer to the solution given in [12] as the genie aided routing algorithm. In a practical setting, however, these probabilistic models have to be "learned" and "maintained.". In Chapter 2, an adaptive distributed opportunistic routing algorithm (d-AdaptOR) was proposed. AdaptOR relies on a large number of packets routed in the network to implicitly learn the optimal routing decisions. In particular, d-AdaptOR achieves an optimal per packet expected cost even when zero knowledge of the transmission success probabilities and network topology is available. The optimality of AdaptOR in terms of its per packet expected cost criterion ensures that the accumulated loss relative to the genie aided routing algorithm (known as *regret*) grows sub-linearly with regard to the number of packets in the network. However, this per packet optimality criterion does not account for the rate of learning among policies with sub linear growth in their regret. The optimal rate of growth of regret (the accumulated loss relative to the genie aided solution) is the topic of this chapter. In particular, we devise an opportunistic routing scheme under which the first N packets see a regret (relative to the genie-aided solution) on the order of $O(\log(N))$. This is shown, in an order sense, to achieve a lower bound on the performance of any uniformly good policy (a policy that is not tuned in for any given probability model).

The remainder of the chapter is organized as follows. In Section 3.2, we provide the system model. Section 3.3 introduces our proposed routing algorithm, NRR. Section 3.4 discuses the outline of proof for the optimality of NRR. Finally we conclude in Section 3.5 and discuss future work.

3.2 System Model

We now consider the problem of routing packets from a source node 0 to a destination node d in a wireless ad-hoc network of $d+1$ nodes denoted by the set $\Theta = \{0, 1, 2, \ldots, d\}$. Associated with each node i, there is a non-negative transmission cost c_i. The transmission cost c_i models the amount of energy used for transmission, the expected time for transmission of a packet, or the hop count when c_i=1. The time is slotted and indexed by $n \geq 0$ (this assumption is not technically critical and is only assumed for ease of exposition).

We consider an opportunistic routing setting with no duplicate copies of the packets. In other words, at a given time the node responsible for routing a given packet transmits the packet and is received by a sub-set of neighbors; the successful reception of the packet by the neighbors is assumed to be known with zero error and delay at the transmitter; and finally, the routing algorithm determines the choice of the next relay among these nodes. More precisely, given a successful packet transmission from node i to the set of neighboring nodes S, the next (possibly randomized) routing decision includes 1) retransmission by node i, 2) relaying the packet by a node $j \in S$, or 3) dropping the packet altogether. If node j is selected as a relay, then it transmits the packet at the next slot, while other nodes $k \neq j, k \in S$, expunge the packet.

We define the termination event for packet m to be the event that packet m is either received at its destination or is dropped by a relay before reaching the destination. We denote this termination action by T. We define the termination time τ_T^m to be the stopping time when packet m is terminated. We discriminate amongst the termination events as follows: We assume that upon the termination of a packet at its destination (successful delivery of a packet to its destination), a fixed and given positive delivery reward r is obtained, while no reward is obtained if the packet is terminated before it reaches the destination. Let r_m denote this random reward obtained at the termination time τ_T^m, i.e. either zero if the packet is dropped prior to reaching the destination node or r if the packet is received at the destination node.

In this work, we formulate the problem in terms of a specific packet generation process at source node 0. Let M_n denote the packet generation process at source node 0 (by definition $M_0 = 1$) where packet m is generated at the source at time $\tau_s^{m+1} = \tau_T^m + 1$, i.e. a new packet is generated at the source when the previous packet has terminated. Note that the assumptions are for analytical tractability, the protocol implementation is not required to abide to this specific traffic generation pattern.

To facilitate learning, we assume that each node also generates probe packets independently of the process M_n. We assume that the probe packets are generated at each node such that the number of the probe packets generated in time slot N are order of $\log(N)$ for unit cost.

An admissible routing policy π is a sequence of routing decisions, i.e. for each packet, it includes the selection of the next hop. The total expected reward obtained upto time N under policy π is given as

$$J^\pi(0, N) = \mathbf{E}^\pi \left[\sum_{m=1}^{M_{N \wedge \tau_T^m - 1}} \left\{ r_m - \sum_{n=\tau_s^m}^{\tau_T^m - 1 \wedge N} c_{i_n^m} \right\} \right], \tag{3.1}$$

where $\mathbf{E}^\pi$ denotes the expectation taken with respect to policy π, M_N denotes the number of packets generated at the source upto time N, i_n^m denotes the index of the node which transmits packet m at time n, Let $J^*(0, N)$ be an optimal total reward such that

$$J^*(0, N) = \sup_{\pi} J^{\pi}(0, N). \tag{3.2}$$

After accounting for the extra overhead of probe packets, we define the actual regret achieved for policy π as

$$R^{\pi}(0, N) = J^*(0, N) - J^{\pi}(0, N) + (d + 1)\log(N). \tag{3.3}$$

Here, we are interested in minimizing the regret $R^{\pi}(0, N)$.

Let π be a uniformly good policy if [38]

$$R^{\pi}(0, N) = O(N^{\alpha}), \forall \alpha > 0. \tag{3.4}$$

We restrict attention only to the class C_U of asymptotically uniformly good adaptive strategies. The regret of uniformly good policies grow slowly and are asymptotically efficient [38].

We will first prove that uniformly consistent policies have an lower bound of $O(\log N)$ i.e.,

$$\liminf_{N \to \infty} \frac{R^{\pi}(0, N)}{\log N} \geq C_0, \quad C_0 > 0, \pi \in C_U. \tag{3.5}$$

Our objective now is to design policy a ϕ^* for which there exists $0 < C_1 < \infty$ such that

$$\limsup_{N \to \infty} \frac{R^{\phi^*}(0, N)}{\log N} \leq C_1. \tag{3.6}$$

3.3 Algorithm: Construction of Efficient strategies

Next, we provide the No Regret Routing (NRR) algorithm ϕ^* which is based on efficiently routing the packets based on the observations so far. Algorithm ϕ^* uses an index function Λ_n and rank-orders the nodes based on index Λ_n. The node with the highest value of Λ_n is chosen for the transmission. Index function Λ_n takes into account how much weight we should give for a current estimate of the model (and hence the greedy decisions) versus how much random exploration we allow for. For instance, if a particular node is not chosen too often for transmission then the estimates of the probability model are not credible enough to justify greedy routing decisions.

In contrast, as the number of packets routed through a given node increases, keeping track of the transmission outcomes enables the algorithm to learn a reliable model to be exploited.

Associated with each node i, the algorithm uses two counting variables $\mathfrak{N}_n(i)$, $\mathfrak{N}_n(S,i)$: $\mathfrak{N}_n(i)$ denotes number of times node i is used for transmission while $\mathfrak{N}_n(S,i)$ denotes number of times a set of nodes S are reached when node i is used for transmission. These counters help to gather the knowledge of the structure of the genie-aided solution. Next, we give detailed description of NRR.

No Regret Routing (NRR) algorithm ϕ^*

0. **Initialization Stage**:
   $n = 0$, $\mathfrak{N}_n(i) = 0$, $\mathfrak{N}_n(S,i) = 0$, for all $i \in \Theta$.

1. **Probing Stage**:
   This stage occurs at time 2^j, $j \geq 0$. In this stage probe packets are generated by each node. Let S_n^i denote the (random) set of nodes that have successfully received the packet from node i. The counting variables are then incremented as follows:
$$\mathfrak{N}_n(i) = \begin{cases} \mathfrak{N}_{n-1}(i) + 1 & \text{if } i \text{ is a transmitter} \end{cases},$$
$$\mathfrak{N}_n(S,i) = \begin{cases} \mathfrak{N}_{n-1}(S,i) + 1 & \text{if } S = S_n^i \\ \mathfrak{N}_{n-1}(S,i) & \text{if } S \neq S_n^i \end{cases}.$$
   In this stage, the empirical probabilities are updated as:
$$P_n(S|i) = \frac{\min\{\mathfrak{N}_n(S,i), 1\}}{\sum_S \min\{\mathfrak{N}_n(S,i), 1\}}.$$
   The value function V_n at time n is obtained using the procedure in stage 5 with input (P_n). At time n, $V_n(i) = V_{n-1}(i)$ if $n \neq 2^j, j \geq 0$.

2. **Reception and Acknowledgment Stage**:
   This stage is assumed to occur at time n. Let i be the node which has transmitted at time n^-. Let S_n^i denote the (random) set of nodes that have successfully received the packet from node i. In the reception and acknowledgment stage, the successful reception of the transmitted packet is acknowledged by all the nodes in the set S_n^i. These nodes form the set of potential relays for node i. Upon reception and acknowledgement, the corresponding counting random variables are incremented.

3. **Indexing Stage:**

   This stage is assumed to occur at n^+. An index for each decision $k \in \Theta \cup \{T\}$ is calculated at time n as

$$\Lambda_n(k) = V_n(k) + \sqrt{\frac{2 \log n}{\mathfrak{N}_n(k)}}. \tag{3.7}$$

4. **Relay/Transmission Stage:**

   This stage is assumed to occur at $(n+1)^-$. In this stage, the next set of relay nodes (actions) are selected. A node k is selected such that[1]:

$$k \in \arg\max_{j \in S \cup \{T\}} \Lambda_n(j).$$

5. **Value function computation**

   This stage is conducted periodically (at a rate much slower than stages 2-4).

   (a) Let $\mathcal{A} = d$ and let $\mathcal{X} = \Theta \backslash \mathcal{A}$.

   Initialize: $V(d) = 0$, $V(T) = 0$ and for $\forall i \in \mathcal{X}$ $V(i) = -\infty$.

   (b) For each $i \in \mathcal{X}$, compute $V(i)$ as

$$V(i) = \max \left\{ -c_i + \sum_{S \in \mathcal{N}(i)} P(S|i) \max_{j \in \mathcal{A} \cap S} V(j), r_i \right\},$$

   where $r_i = r > 0$ for $i = d$, else $r_i = 0$.

   (c) Append node $i \in \mathcal{X}$ with the highest value of $V(i)$ to the ordered set $\mathcal{A}$ and remove node i from $\mathcal{X}$.

   (d) If $\mathcal{X}$ is empty stop or else go to step 5b.

   This stage rank-orders the nodes assuming that the current estimate of the broadcast model is accurate. In [12] this procedure is shown to result in the optimal opportunistic routing algorithm when the network topology and the local broadcast model are available.

[1] In case of the ambiguity, the node with smallest index is chosen.

3.4 Optimality of NRR

We will now state the main result establishing the optimality of the proposed NRR algorithm under the assumptions of a time-invariant model of packet reception and a centralized controller. More precisely, we have the following assumptions:

Assumption 3. *The probability of a successful reception of a packet transmitted by node i at set $S \subseteq \mathcal{N}(i)$ of nodes is $P(S|i)$, independent of time and all other routing decisions.*

Assumption 4. $P(\cdot|j) = 0 \Rightarrow P_n(\cdot|j) = 0, \forall n \geq n_0, n_0 \geq 0.$

Assumption 5. *The successful reception at set S due to transmission from node i is known to the centralized controller.*

The probabilities $P(\cdot|\cdot)$ in Assumption 3 characterize a packet reception model referred as the *local broadcast model*. Note that for all $S \neq S'$, successful reception at S and S' are mutually exclusive and $\sum_{S \subseteq \Theta} P(S|i) = 1$. Furthermore, logically node i is always a recipient of its own transmission, i.e. $P(S|i) = 0$ iff $i \notin S$. Assumption 3 is in line with the experimentally tested routing protocol ExOR [10] which indicates that reasonably simple probabilistic models provide good abstractions of the media access control (MAC) and physical (PHY) layers at the routing layer. Assumption 4 holds valid in practice, since it determines that the existence of neighbours using probe packets can be achieved reliably. Assumption 5 is for ease of exposition and can be relaxed. In fact [12] provides a distributed massage passing which can compute $V_n(i)$ using Bellman-ford iteration.

3.4.1 Lower bound on regret for NRR

We first find the lower bound $R^{\pi}(0, N)$, $\pi \in C_U$ by constructing a specific example of routing. In this example, we decompose the routing problem into a multi-arm bandit problem. Before we discuss the relationship, we briefly describe the multi-arm bandit setting.

Multi-arm Bandit Problem: A two multi-arm bandit problem is defined by reward random variables $Y_n^i, i = 1, 2$, where i is the arm. Successive plays of arm i yield reward Y_n^i which are independent and identically distributed and generated according a Bernoulli distribution with unknown parameter. Let Z_n denote the random variable denoting the arm pulled at time n. Let $p^i, p^1 > p^2$ be the mean of the unknown distribution. The regret until time N following a policy π is given by,

$$R_B^{\pi}(N) = Np^1 - E^{\pi} \sum_{i=1}^{2} \sum_{n=1}^{N} I(Z_n = i) Y_n^i \tag{3.8}$$

Fact 2 (Theorem 3.1 [38]). If the policy $\pi \in C_U$, then a lower bound for R_B^{π} is

$$\inf_{\pi \in C_U} \liminf_{N \to \infty} \frac{R_B^{\pi}(N)}{\log N} \geq \frac{1}{D(p^1 \,||\, p^2)}, \tag{3.9}$$

where $D(p^1 \,||\, p^2)$ is the Kullback-Leibler divergence between binomial distributions with mean p^1 and p^2.

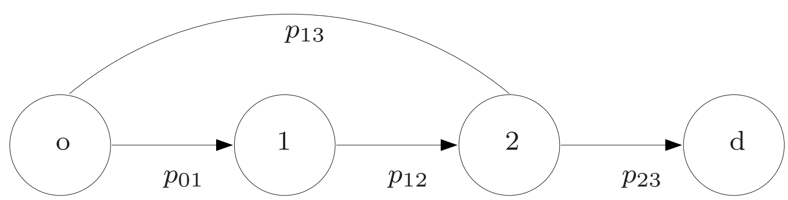


Figure 3.1: With probability p_{ij}, a packet transmitted by node i is received by node j.

Consider a topology as shown in Figure 3.1 with the link success probabilities as shown. Let $\lambda(i) = 0, i \neq s$. At time τ_s^m node s transmits a packet which is successfully received by nodes 1 and 2. At time slot $\tau_s^m + 1$ a node from 1 and 2 is chosen for next transmission. The problem is equivalent to a multi arm bandit problem, in which an arm coincides with the choice of a node as the next relay. Thus,

$$\inf_{\pi \in C_U} \liminf_{N \to \infty} \frac{R^{\pi}(N)}{\log N} \geq C(\text{ for some } C > 0 \text{ }). \tag{3.10}$$

3.4.2 Upper bound on regret for NRR

To obtain the upper bound on regret we will need the following definition of the reward function associated with each packet m. Let the termination action be denoted by T and the termination state be F. The termination state F is the state visited by the system when the termination action T is chosen, i.e. $P(F|T) = 1$. Given a set S

of nodes that have received a packet from one of the nodes in Θ, the set of allowable actions is denoted by $A(S) = S \cup \{T\}$. The allowable action in the termination state F as well as all states containing the destination is T, i.e. $\{T\} = A(F) = A(S)$ for all S containing d.

It remains to define the reward function $g : \mathfrak{S} \times \mathcal{A} \to \mathbb{R}$ to represent the reward obtained from taking an action at a given state. In summary, $g(S, a)$ is given as:

$$g(S,a) = \begin{cases} -c_i & a = i \in S \\ r & a = T\ ,\ d \in S \\ 0 & a = T\ ,\ d \notin S \end{cases} .$$

In view of this model, we can replace regret as following

$$R^{\pi}(0, N) = \mathbf{E}^{*}[\sum_{n=0}^{N} g(S_n, a_n)] - \mathbf{E}^{\pi}[\sum_{n=0}^{N} g(S_n, a_n)]. \tag{3.11}$$

Using the proof technique used in [39], [40] we now reorganize the terms in Lemma 4 and restate $R^{\pi}(0, N)$ as:

Lemma 4.

$$R^{\phi^*}(0, N) = \sum_{S} \sum_{a:\rho^*(S,a)\neq 0} \mathbf{E}^{\phi^*}[\mathfrak{N}_N(S, a)]\rho^*(S, a), \tag{3.12}$$

where $\rho^*(S, a)$ is a constant for each S and a.

Finally, we show that when $\rho^*(S, a) \neq 0$, $E[\mathfrak{N}_N(S, a)] = O(\log(N))$, which proves an achievable upper bound for NRR.

3.5 Summary

In this paper, we proposed No Regret Routing NRR, an adaptive opportunistic routing algorithm whose performance is shown to be optimal with respect to a performance loss criterion known as *regret* in the absence of knowledge regarding network topology and/or channel statistics. More precisely, under idealized assumptions the accumulated loss under NRR relative to the genie aided solution is shown to grow logarithmically with time which, in an order sense, coincides with the lower bound on the performance of any uniformly good policy.

Acknowledgement

This chapter, in part, appears in the following publications. The dissertation author was the primary investigator and author of this paper.

- A. Bhorkar, T. Javidi, No regret routing in wireless Ad-hoc networks, Asilomar November, 10

I would like to thank my co-author Prof. Tara Javidi.

Chapter 4

Opportunistic Routing with Congestion Diversity protocol

Abstract

We consider the problem of routing packets across a multi-hop network consisting of multiple sources of traffic and wireless links while ensuring bounded expected delay. Each packet transmission can be overheard by a random subset of receiver nodes among which the next relay is selected opportunistically. The main challenge in the design of minimum-delay routing policies is balancing the trade-off between routing the packets along the shortest path to the destination and distributing the traffic according to the maximum backpressure. Combining important aspects of shortest path and backpressure routing, this paper provides a systematic development of a distributed opportunistic routing policy with congestion diversity (D-ORCD).

D-ORCD uses a measure of draining time to opportunistically identify and route packets along the paths with an expected low overall congestion. D-ORCD is proved to ensure a bounded expected delay for all networks and under any admissible traffic. Furthermore, this paper proposes a practical implementation which empirically optimizes critical algorithm parameters and their effects on delay as well as protocol overhead. Realistic Qualnet simulations for 802.11-based networks demonstrate a significant improvement in the average delay over comparative solutions in the literature.

4.1 Introduction

This chapter provides a distributed opportunistic routing policy with congestion diversity (D-ORCD) under which the congestion information is integrated with the distributed shortest path computations of [12]. The main contribution of this chapter is to design a distributed routing policy that improves the delay performance over existing routing policies while ensuring throughput optimality. In [41], ORCD, a centralized version of D-ORCD, is shown to be throughput optimal without any discussion of system implications. In this chapter, we extend the throughput optimality proof for the distributed version and discuss implementation issues in detail. We also tackle some of the system level issues observed in realistic settings via detailed Qualnet simulations. We then show that D-ORCD exhibits better delay performance than state of the art routing policies, namely, EXOR, DIVBAR and E-DIVBAR.

Before we close, we emphasize that some of the ideas behind the design of D-ORCD have also been used as guiding principles in many routing solutions: some in an opportunistic context [42, 43] and some in a conventional context [44]. Below, we detail the similarity and difference between these solutions and our work for the sake of completeness, even though, in our study, we have chosen to focus only on solutions with a comparable overhread and similar degree of practicality. In [42], perhaps the work most related to ours, the authors consider a flow-level model of the network and propose a routing policy referred to as *min-backlogged-path* routing, under which the flows are routed along the paths with minimum total backlog. In this light, D-ORCD can be viewed as a packet-based version of min-backlogged-path routing without a need for the enumeration of paths across the network and costly computations of the total backlog for all paths. In [43], the authors propose a modified version of backpressure which uses the shortest path information to minimize the average number of hops per packet delivery, while keeping the queues stable. In [44], a modified throughput optimal backpressure policy, LIFO-Backpressure, is proposed using LIFO discipline at layer 2. Neither of these approaches lend themselves to practical implementations: [43] requires maintaining large numbers of virtual queues at each node thus increasing the implementation complexity, while [44] uses an atypical LIFO scheduler resulting in significant reordering of the packets. Furthermore, while LIFO-Backpressure policy guarantees stability with minimal queue-length variations, realistic bursty traffic in large multi-hop wireless networks may result in queue-length variations and thus high delay.

The chapter is organized as follows. In Section 4.2, we describe the D-ORCD routing algorithm. In Section 4.3, we discuss various of D-ORCD protocol implementation issues. Section 4.4 describes in detail our simulation results, and we compare the performance of various routing policies with D-ORCD. We then discuss theoretical guarantees of D-ORCD in Section 4.5. We provide concluding remarks and discuss directions for future research in Section 4.6. The appendix contains proofs of the throughput optimality of D-ORCD performance under certain assumptions on the model.

4.2 Opportunistic Routing with Congestion Diversity

The goal of this work is to design a routing policy with improved delay performance over existing opportunistic routing policies. In this section, we describe the guiding principle and the design of Opportunistic Routing with Congestion Diversity (D-ORCD). We propose a time-varying distance vector, which enables the network to route packets through a neighbor with the least estimated delivery time.

D-ORCD opportunistically routes a packet using three stages: (a) a transmission, (b)an acknowledgment, and (c) further relaying. During the transmission stage, a node transmits a packet. During the acknowledgment stage, each node that has successfully received the transmitted packet, sends an acknowledgment (ACK) to the transmitter node. D-ORCD then makes routing decisions based on a congestion-aware distance vector metric referred to as the *congestion measure*. Specifically, during the relaying stage, the relaying responsibility for the packet is shifted to a node with the least congestion measure among the nodes that have received the packet. The congestion measure of a node associated with a given destination node provides an estimate of the best possible draining time of a packet arriving at that node for reaching to destination. Each node is responsible for updating its congestion measure and for transmitting this information to its neighbors. Next, we detail D-ORCD design and the computations performed at each node to update the congestion measure.

4.2.1 D-ORCD Design

We consider a network of D nodes labelled by $\Omega = \{1, \ldots, D\}$. Let p_{ij} be the probability that the packet transmitted by node i is successfully received by node j. Node j is said to be *reachable* by node i, if $p_{ij} > 0$. The set of all nodes in the network which are reachable by node i is referred to as the *neighborhood* of node i and is denoted by $\mathcal{N}(i)$.

D-ORCD relies on a routing table at each node to determine the next best hop. The routing table at node i consists of a list of neighbors $\mathcal{N}(i)$ and a structure consisting of an estimated congestion measure for all these neighbors in $\mathcal{N}(i)$ associated with different destinations. The routing table acts as a storage and decision component at the routing layer. The routing table is updated using a "virtual routing table" at the end of every "computational cycle": an interval of T_c units of time. During the progression of the computation cycle, the nodes exchange and compute the temporary congestion measures to update the virtual routing table. The temporary congestion measures are computed in a fashion similar to the distributed stochastic routing computation of [12] using the backlog information at the beginning of the computation cycle (generalizing the computations of distributed Bellman-Ford). We conceptualize this in terms of a virtual routing table updating and maintaining these temporary congestion measures. We assume that each node has a common global time to ensure that the nodes update the routing table at roughly the same time.

We denote the temporary congestion measure associated with node $i \in \Omega$ at time t and destination $d \in \Omega$ as $V_i^d(t)$. Each node i computes $V_i^d(t)$ based on congestion measures $\tilde{V}_k^{(i,d)}(t)$ obtained via periodic communication with its neighbours $k \in \mathcal{N}(i)$ and the queue backlog at the start of the computation cycle. D-ORCD stores these temporary congestion measures $\{V_i^d(t)\}_{d\in\Omega}$ and $\{\tilde{V}_k^{(i,d)}(t)\}_{d\in\Omega,k\in\mathcal{N}(i)}$ in the virtual routing table. More precisely, node i periodically computes its own congestion measure and subsequently advertises it to its neighbors using control packets at intervals of $T_s \leq T_c$ seconds. Finally the actual routing table is updated using the entries in the virtual routing table every T_c seconds. The sequence of operations performed by D-ORCD are shown in Figs. 4.1,4.2.

Meanwhile, for routing decisions, node i uses the entries in the actual routing

Table 4.1: Notations used in the description of the algorithm

Symbol	Definition
$\mathcal{N}(i)$	Neighbours of node i
$V_i^d(t)$	Congestion measure at node i at time t
$\tilde{V}_k^{(i,d)}(t)$	Congestion measure obtained at node i from node k
$T(t)$	Ending time of the latest computation cycle before time t
T_c	Duration of the computation interval
T_s	Control packet transmission interval
$L_i(t)$	Local congestion at node i
$D_i(t)$	Congestion down the stream for node i
$K_{D-ORCD}^{(i,d)}(t)$	Selected relay for transmission at node i
$S_i(t)$	Set of nodes receiving the packet transmitted by node i
$Q_i^d(t)$	Queue-length at node i destined for d at time t
$\bar{Q}_i^d(t)$	Average queue-length at node i destined for d
$P_{succ-k}^{(i,d)}(t)$	Probability that the highest priority node k receives the packet
$P^{(i,d)}(t)$	Probability that at-least one higher priority node receives the packet
$H^{(i,d)}(t)$	Set of higher priority nodes than node i

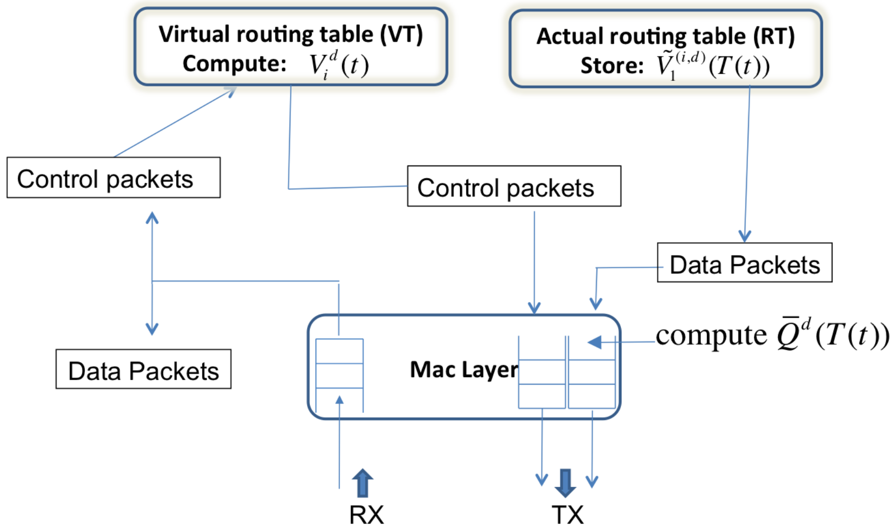


Figure 4.1: Operation of D-ORCD

routing table (computed during the last computation cycle). Let $T(t) = \max_n\{nT_c : nT_c \le t, n \in \mathrm{Z}\}$ be the ending time of the latest computation cycle. Then node i stores $\tilde{V}_k^{(i,d)}(T(t))$ in the actual routing table and selects the next best hop $K_{D-ORCD}^{(i,d)}$ to minimize the packet's draining time, i.e.

$$K_{D-ORCD}^{(i,d)}(t) = \arg\min_{k \in S_i(t) \cup i} \tilde{V}_k^{(i,d)}(T(t)), \tag{4.1}$$

where $S_i(t)$ denotes a random set of nodes receiving the packet transmitted by node i at time t.

Next, we describe the distributed computations performed during each computation cycle.

4.2.2 Congestion Measure Computations

The congestion measure associated with node i for a destination d at time t is the aggregate sum of the local draining time at node i (denoted by $L_i^d(t)$) and the draining time from its next hop to the destination (denoted by $D_i^d(t)$), i.e.

$$V_i^d(t) \quad = \quad L_i^d(t) + D_i^d(t). \tag{4.2}$$

Assuming a FIFO discipline at layer-2, we proceed to decompose the local draining time. This relies on the observation that when a packet arrives at a node, i, its waiting

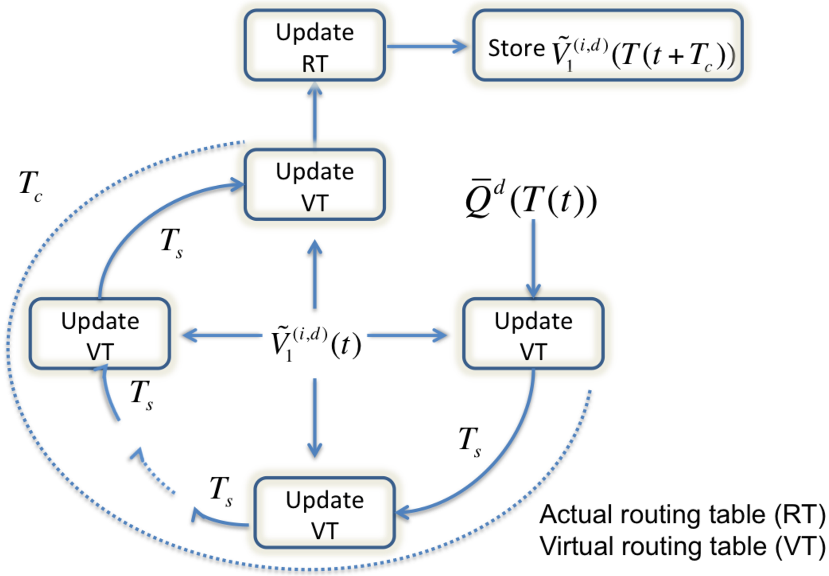


Figure 4.2: Actual routing table and virtual routing table updates.

time is equal to the time spent in draining the packets that have arrived earlier plus its own transmission time. If $P^{(i,d)}(t)$ denotes the probability that the packet transmitted by node i is successfully received by a node with a lower congestion measure, then the packet's expected transmission time at node i is given by $\frac{1}{P^{i,d}(t)}$. Let $\bar{Q}_i^d(t)$ denote the number of packets destined for destination d averaged over the previous computation cycle. $\bar{Q}_i^d(t)$ is updated as

$$\bar{Q}_i^d(t) = \frac{T_s}{T_c} \sum_{l=0}^{\frac{T_c}{T_s}-1} Q_i^d(T(t) - l).$$

The local draining time for node i to destination d at time t is approximated as,

$$L_i^d(t) = \frac{1}{P^{(i,d)}(t)} + \sum_{d' \in \Omega} \frac{\bar{Q}_i^{d'}(T(t))}{P^{(i,d')}(t)}. \tag{4.3}$$

D-ORCD computes the expected congestion measure "down the stream" for each node $i \in \Omega$ using the latest congestion measure $\tilde{V}_k^{(i,d)}(t)$ received from nodes $k \in \Omega$ with lower congestion measure. With respect to the destination d, a node $k \in \Omega$ is defined as a higher priority node than node i if $\tilde{V}_k^{(i,d)}(t) < V_i^d(t)$ and the set of higher priority nodes is defined as $H^{(i,d)}(t)$. Let $P_{succ-k}^{(i,d)}(t)$ be the probability that node k is the highest priority node to successfully hear node i at time t and $k \in H^{(i,d)}(t)$. As a result, the expected congestion "down the stream" $D_i^d(t)$ can be given as

$$D_i^d(t) = \sum_{k \in \Omega} P_{succ-k}^{(i,d)}(t) \tilde{V}_k^{(i,d)}(t). \tag{4.4}$$

Remark 5. In each computation cycle, assuming T_c is large, D-ORCD computations converge to the Bellman equation associated with the minimum cost ("shortest path") route in a network, where the link costs are given in terms of the queue length $\bar{Q}_i^d(t)$.

Remark 6. If the link success probabilities have independent realizations, then for all $S \subseteq \Omega$, $P(S_i(t) = S) = \prod_{k \in S} \prod_{l \notin S} p_{ik}(1 - p_{il})$. The success probabilities $P^{(i,d)}(t)$ and $P_{succ-k}^{(i,d)}(t)$ can be calculated as

$$P^{(i,d)}(t) = \sum_{S: \exists k \in S \cap H^{(i,d)}(t)} P(S_i(t) = S), \tag{4.5}$$

$$P_{succ-k}^{(i,d)}(t) = \frac{1}{P^{(i,d)}(t)} \times \sum_{\substack{S: \tilde{V}^{(k,d)}(t) < \tilde{V}^{(k',d)}(t) \\ k', k \in S \cap H^{(i,d)}(t)}} P(S_i(t) = S). \tag{4.6}$$

4.2.3 Opportunistic Routing with Partial Diversity

The three-way handshake procedure discussed in Section 4.2.1 to achieve receiver diversity gain in an opportunistic scheme is achieved at the cost of an increase in the control overhead. In particular, it is easy to see that this overhead cost, which is the total number of ACKs sent per data packet transmission, increases linearly with the size of the set of potential forwarders. Thus, we consider a modification of D-ORCD in the form of opportunistically routing with partial diversity (P-ORCD). This class of routing policies is parametrized by a parameter M denoting the maximum number of forwarder nodes. This is equivalent to a constraint on the maximum number of nodes allowed to send an acknowledgment per data packet transmission. Such a constraint will sacrifice the diversity gain, and hence the performance of any opportunistic routing policy, for lower communication overhead.

In order to implement opportunistic routing policies with partial diversity, before the transmission stage occurs, we find the set of "best neighbors" for each node i at any time t, denoted by $B_i^*(t)$, where $|B_i^*(t)| \leq M$. After transmission of a packet from node i at time t, the routing decision is made as follows: 1) among the nodes in $B_i^*(t) \cap S_i(t)$, select a node with the lowest congestion measure as the next forwarder; or 2) retain the packet if none of the nodes in the set $B_i^*(t)$ has received the packet. Next we present a mathematical formulation for modification of D-ORCD with partial diversity.

Let $\mathcal{B}$ be the collection of all subsets of Ω of size less than or equal to M, i.e. $\mathcal{B} = \{B \subseteq \Omega : |B| \leq M\}$.

In the D-ORCD protocol with partial diversity, (PD-ORCD), the corresponding quantities $\bar{V}_i^d(t)$ are updated as

$$\bar{V}_i^d(t) = \min_{B \in \mathcal{B}} \left\{ L_i^d(t) + \sum_{k: k \in \mathcal{B}} P_{succ-k}^{(i,d)}(t) \tilde{\bar{V}}_k^{(i,d)}(t) \right\}, \tag{4.7}$$

while the next hop is selected as

$$K_{PD-ORCD}^{(i,d)}(t) = \underset{k \in \{S_i(t) \cap B\} \cup i}{\arg\min} \tilde{\bar{V}}_k^{(i,d)}(T(t)). \tag{4.8}$$

We carry out a simulation study for the delay performance of D-ORCD with these modifications and compare it to the delay performance of the other routing policies in Section 4.4.

Remark 7. When $M = 1$, each node can send packets to only one of its neighbors. Therefore, this routing policy cannot take the advantage of the broadcast nature of wireless transmissions any longer, and is classified as conventional routing.

In the next section, we discuss the practical issues associated with computation of the time-varying congestion measures $V_i^d(t)$, $i \in \Omega$. Furthermore, we propose practical implementations and heuristics.

4.3 Implementation Details: Protocol Components

In this section, we discuss the implementation issues of D-ORCD which involve distributed and asynchronous iterative computations of $V_i^d(t)$'s. We provide a brief discussion of the basic challenges of D-ORCD including the three-way handshake procedure employed at the MAC layer, link quality estimation, and the avoidance of loops while routing.

4.3.1 802.11 Compatible Implementation

Three way Handshake

The implementation of D-ORCD, analogous to any opportunistic routing scheme, involves the selection of a relay node among the candidate set of nodes that have received and acknowledged a packet successfully. One of the major challenges in the implementation of an opportunistic routing algorithm, in general, and D-ORCD in particular, is the design of an 802.11 compatible acknowledgement mechanism at the MAC layer. Below we propose a practical and simple way to implement an acknowledgement architecture.

The transmission at any node i is done according to 802.11 CSMA/CA mechanism. Specially, before any transmission, transmitter i performs channel sensing and starts transmission after the backoff counter is decremented to zero. For each neighbor node $j \in \mathcal{N}(i)$, the transmitter node i then reserves a virtual time slot of duration $T_{ACK} + T_{SIFS}$, where T_{ACK} is the duration of the acknowledgement packet and T_{SIFS} is the duration of Short Inter Frame Space (SIFS) [33]. Transmitter i then piggy-backs a priority ordering of nodes $\mathcal{N}(i)$ with each data packet transmitted. The priority ordering determines the virtual time slot in which the candidate nodes transmit their acknowledgement. Nodes in the set S_i that have successfully received the packet then transmit acknowledgement packets sequentially in the order determined by the transmitter node.

After a waiting time of $T_{wait} = |\mathcal{N}(i)|(T_{ACK} + T_{SIFS})$ during which each node in the set S_i has had a chance to send an ACK, node i transmits a FOrwarding control packet (FO). The FO packets contain the identity of the next forwarder, which may be node i itself (i.e. node i retains the packet) or any node $j \in S_i$. If T_{wait} expires and no FO packet is received (FO packet reception is unsuccessful), then the corresponding candidate nodes drop the received data packet. If transmitter i does not receive any acknowledgement, it retransmits the packet. The backoff window is doubled after every retransmission. Furthermore, the packet is dropped if the retry limit (set to 7) is reached.

4.3.2 Control Packets Fidelity

D-ORCD depends on a reliable, frequent, and timely delivery of the control packets. As documented in [45], the loss of control packets may destabilize the algo-

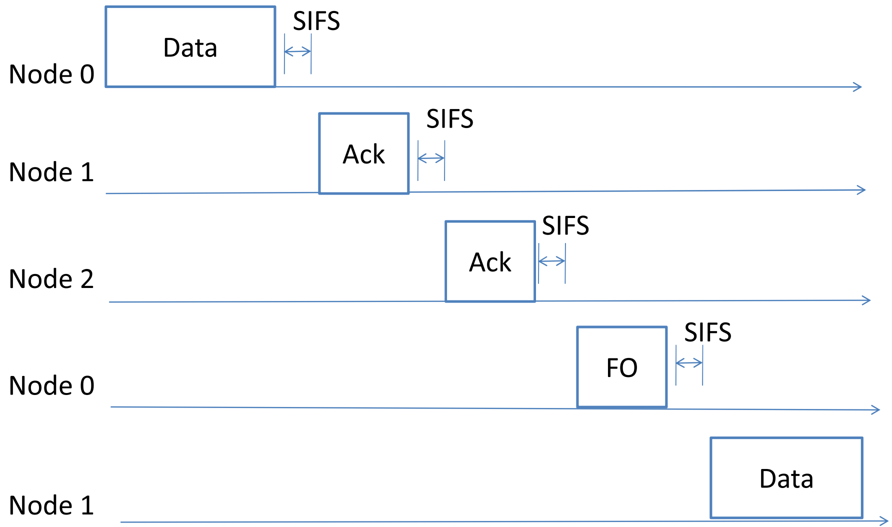


Figure 4.3: Typical packet transmission sequence for D-ORCD

rithm operation and cause significant performance degradation for many well known routing algorithms. In our implementation, we have taken advantage of the priority-based queuing to implement this component of the control plane. D-ORCD prioritizes the control packets by assigning them the highest strict priority, reducing the probability that the packets are dropped at the MAC layer and ensuring a timely delivery of the control packets. In particular, D-ORCD utilizes priority queues: data packets are assigned to the lower priority queue and control packets are assigned to the higher priority queue. Moreover, D-ORCD scheduler assigns a sufficiently lower PHY rate for the control packets to improve the probability of successful reception.

4.3.3 Link Quality Estimation Protocol

D-ORCD computations given by (4.2) utilize link success probabilities p_{ij} for each pair of nodes i, j. We now a describe method to determine the probability of successfully receiving a data packet for each pair of nodes $i, j \in \Omega$. It consists of two phases: active probing phase and passive probing phase. In the active probing phase, dedicated probe packets are broadcast periodically to estimate link success probability. In passive probing, we utilize the overhearing capability of the wireless medium. The nodes are configured to a promiscuous mode, hence enabling them to hear the packets from neighbors. In passive probing, the MAC layer keeps track of the number of packets received from the neighbors including the retransmissions. Finally, a weighted average is used to combine the active and passive estimates to determine the link success probabilities. Passive probing does not introduce any additional overhead cost but can be slow, while the active probing rate is set independently of the data rate but introduces costly overhead.

4.3.4 Loop Avoidance Heuristic

D-ORCD approximates the solution to the fixed point equation via a distributed distance vector approach. The classical problem of counting to infinity [46] in distance vector routing can affect D-ORCD performance due to the time varying nature of the congestion metric. The problem is most acute when there is a sudden burst of traffic.[1] and can cause severe transient effects due to slow updates of the control packets. The looping results in large delays, increased interference and loss of packets.[2]

In our experiments, to address this issue, we utilize an extension of the Split-horizon with poison reverse solution [47] to avoid loops. In Split-horizon with poison reverse, a node advertises routes as unreachable to the node through which they were learned. We have extended the rule to D-ORCD by advertising the routes as unreachable to higher ranked nodes. This removes most looping routes before they can propagate through the network.

4.4 Simulations

In this section, we compare the expected delay encountered by the packets in the network using Qualnet simulations under various opportunistic routing policies: ExOR, DIVBAR, E-DIVBAR and D-ORCD. We first investigate the performance of D-ORCD

[1] Similar to the broken link scenario in a typical distance vector routing.

[2] Packet loss occurs when time to live (TTL) value exceeds the number of allowed hops (typically 64).

with respect to a canonical example to demonstrate D-ORCD gains [1]. We then use a realistic topology of 16 nodes placed in a grid topology to demonstrate the robust performance improvement achievable in practical settings.

4.4.1 The Simulation Setup

Our simulations are performed in QualNet. We consider two set of topologies in our experimental study:

1. Canonical Example: In this example, we study the canonical example in Fig. 4.4. We motivate the performance improvement for D-ORCD by a scenario which exemplifies the need to avoid congestion in the network by highlighting the shortcomings of the existing routing paradigms: shortest path and backpressure.

2. Grid Topology: We study an outdoor wireless settings of grid topology consisting of 16 nodes separated by a distance of 200 meters. These simulations demonstrate a robust performance gain under D-ORCD in a realistic network.

We now describe parameters settings in the simulation. The nodes are equipped with 802.11b radios transmitting at 11 Mbps with a transmission power of 15 dBm. The wireless medium model includes Rician fading with a K-factor of 4 and Log-normal shadowing with a mean of 4dB. In the canonical example path loss is determined by pathloss matrix which gives the attenuation of the received signal power with distance from the transmitter for every pair of network nodes, while for a grid topology the path loss follows the ITM model in [35]. The antenna model is the standard omnidirectional antenna model with the default settings of the simulator. The network queues are FIFO with finite buffer size of 750 KB.

The acknowledgement packets are short packets of length 24 bytes transmitted at 11 Mbps, while FO packets are of length 20 bytes and transmitted at the lower rate of 1 Mbps to ensure reliability. If unspecified, packets are generated according to a poission modulated Markov traffic. The packets are assumed to be 512 bytes in length and equipped with simple cyclic redundancy check (CRC) error detection. The control packets are transmitted periodically at an interval of $T_s = 0.5$ second.

We have chosen partial diversity $M = 4$ and an update frequency $T_c = T_s = 0.5$ second in our experiments. A discussion on the choice of parameters in the design of D-ORCD is provided in Section 4.4.4.

In our study, we have compared the performance of D-ORCD against the state of the art routing algorithms. Before we proceed, we describe these candidate algorithms as well as our implementation of them.

- DIVBAR [48]: We implemented DIVBAR to select the next hop based on a weighted differential backlog. Specifically, let $\tilde{Q}_k^{(i,d)}(t)$ denote the latest information at node i about the number of packets buffered in queue k for destination d. For any destination d, DIVBAR chooses the next hop $K_{DIVBAR}^{(i,d)}(t)$, such that

$$K_{DIVBAR}^{(i,d)}(t) = \arg\min_{k \in S_i(t) \cup i} (\tilde{Q}_k^{(i,d)}(t) - Q_i^d(t)). \tag{4.9}$$

  We have created virtual queues for each destination to identify differential backlog associated with different destinations. Note that original backpressure algorithm proposed in [48] is utilized in conjunction with a scheduler to maximize the network's overall weighted differential backlog as well as a mechanism to choose the destination queue to be served. In our implementation, we serve the packets in a prioritized manner based on the destination using 802.11 MAC. Specifically, a packet with destination $m(t)$ is selected among all possible virtual queues such that

$$m(t) = \arg\min_{d} \{\min_{k} (\tilde{Q}_k^{(i,d)}(t) - Q_i^d(t))\}. \tag{4.10}$$

  In order to implement a priority scheduling we utilize a priority scheduler such that the packet destined for $m(t)$ is assigned to a higher priority queue. We have implemented the DIVBAR algorithm using a structure similar to D-ORCD (in which $V_i^d(t)$ is replaced with $Q_i^d(t)$).

- ExOR [10] : ExOR uses an ETX metric when routing the packet without considering queuing information at the nodes. Specifically, for a packet destined for node d, the next hop $K_{ExOR}^{(i,d)}$ is chosen such that

$$K_{ExOR}^{(i,d)}(t) = \arg\min_{k \in S_i(t) \cup i} ETX^{(k,d)}, \tag{4.11}$$

where $ETX^{(k,d)}$ is the minimum number of transmissions from node k to destination d given by,

$$ETX^{(k,d)} = \min_{j} \{ \frac{1}{p_{kj}} + ETX^{(j,d)} \}. \tag{4.12}$$

We have used our distributed architecture for the calculation of the ETX metric by taking $Q_i(t) = 1$ for all $i \in \Omega$ and $M = 1$ in the calculation of $V_i(t)$, even though, in principle, the overhead can be held much lower due to the time invariant nature of node ordering.

- E-DIVBAR [13]: E-DIVBAR is a variant of DIVBAR, where along with the queue information, ETX metric is used for path selection. In particular, for a packet destined for d, the next hop $K^{(i,d)}_{E-DIVBAR}$ is chosen such that

$$K^{(i,d)}_{E-DIVBAR}(t) \quad = \quad \arg\min_{k \in S_i(t) \cup i} \left\{ (\tilde{Q}^{(i,d)}_k(t) - Q^d_i(t)) ETX^{(k,d)} \right\}.$$

  The E-DIVBAR algorithm is also implemented using a structure identical to D-ORCD and DIVBAR, however, the control packets contain information about the queue-length as well as the ETX for a given destination. The commodity selection is performed using the same equation (4.10) as for DIVBAR.

Next, we study the canonical example where we compare the average delay encountered by packets in the network under various routing policies: ExOR, DIVBAR, E-DIVBAR and D-ORCD. The choice of the canonical network enables us to clearly reveal the high capability of D-ORCD in balancing the traffic taking advantage of path diversity in the network.

4.4.2 Performance of D-ORCD: Canonical Example

Consider the network shown in Fig. 4.4 which is parameterized by N. Nodes $12, 13, \ldots, 12 + (N-1)$ form a "hole" in the network whose size is controlled by the parameter N. We now discuss the delay gains under D-ORCD as parameters N and λ_1 (the incoming traffic rate at node 1) are varied and verify them in this section.

Note that the source node 1 can route packets either through node 2 or node 3. Since only node 1 has a routing choice, we focus on the delay experienced by packets

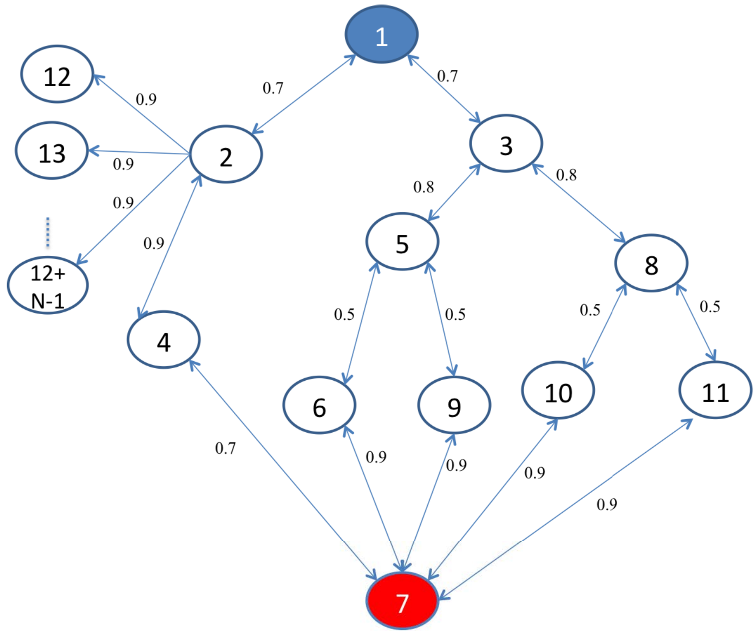
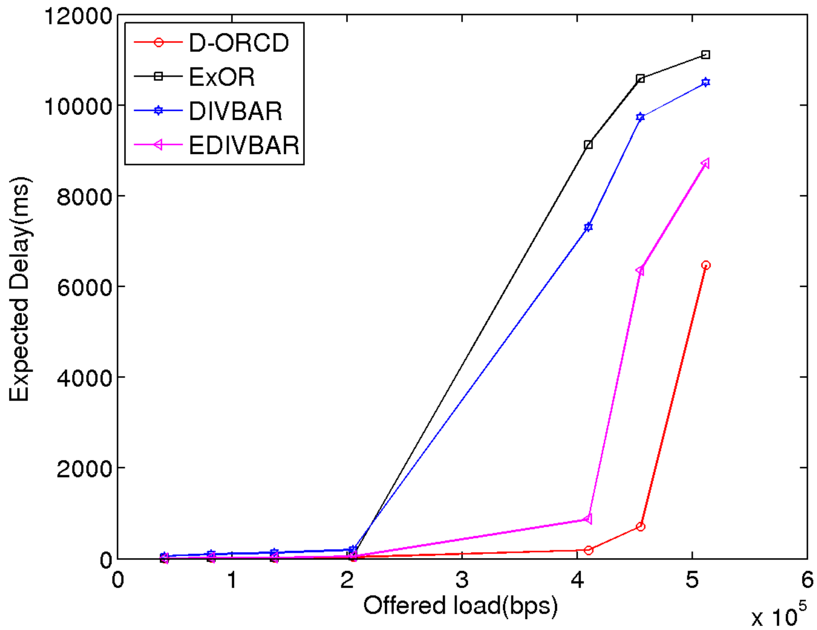


Figure 4.4: Structure of the canonical network from [1]

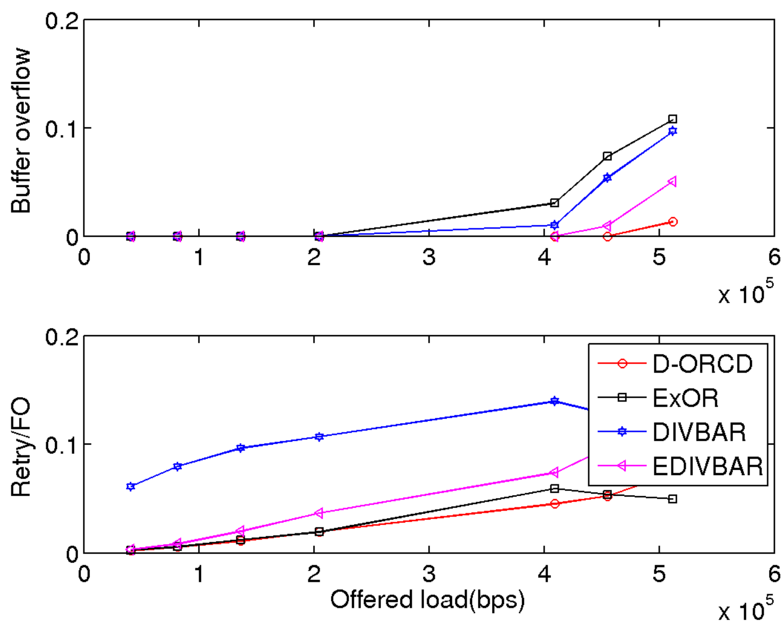
originating in node 1. Fig. 4.5 provide plots of the average end-to-end packet delay and the buffer overflow ratios for all the routing algorithms as the arrival rate λ_1 is varied. We observe that D-ORCD has better delay performance than the other algorithms over the range of incoming traffic rates considered. Fig. 4.6 plots the highest priority next hop for node 1 under the candidate protocols throughout the duration of the experiments.

ExOR gives higher priority to node 2 than node 3 independent of the congestion at intermediate nodes ($ETX^{(2,7)} = 2.53$ and $ETX^{(3,7)} = 4.36$). ExOR can thus suffer from poor delay performance as the arrival rate at node 2 approaches capacity. ExOR has the worst delay performance among all the algorithms as seen in Fig 4.6 particularly when the traffic load on the network is high. In Fig. 5.15 we observe that DIVBAR and E-DIVBAR forward significant number of packets into 12,13 and 14 increasing the interference and packet drops as well as delay.

Next, we study the impact of the size of the "hole"; i.e. N on the expected per packet delay. Under DIVBAR the packets that arrive at node 2 from source 1 are likely to be forwarded and wander between nodes $12, 13 \ldots, 12 + (N - 1)$ before eventually forwarding to 4. In contrast, increasing N has no effect on the performance of D-ORCD. This is because $V_2^7(t) < V_{12+i}^7(t)$, $i = 0, 2, \ldots, N - 1$, for all time slots t, in



(a) Delay



(b) Fraction of packet loss

Figure 4.5: Performance for Canonical Example for N=2

effect, preventing the packets to enter the "hole". Fig. 4.7 provides the expected delay encountered by the source packets under various routing policies, as the size of the "hole", N, increases and the arrival rate is set to low value of $\lambda_1 = 200$ kbps. The figure shows that the average delay under D-ORCD is significantly lower than other candidate protocols as N increases from 1 to 5.

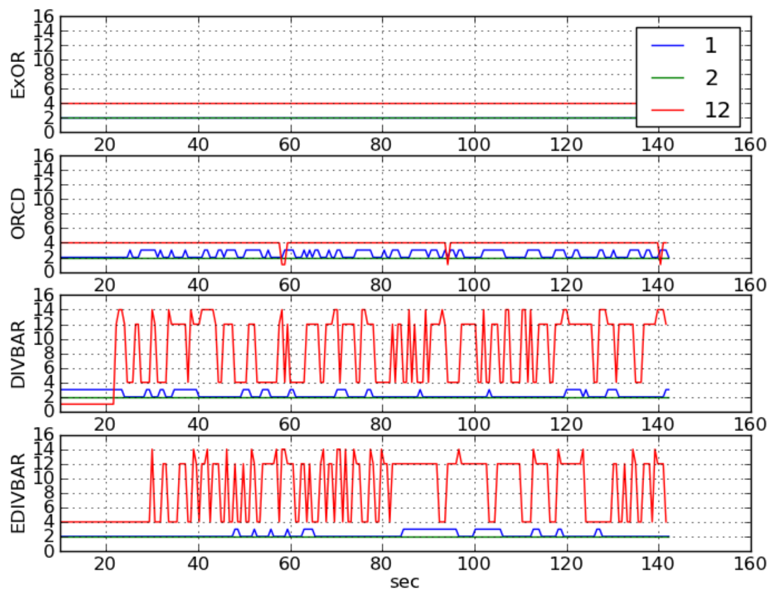


Figure 4.6: Highest priority nodes for Canonical Example.

4.4.3 Performance of D-ORCD: Grid Topology

We perform simulations for the grid networks of 16 nodes in Fig. 4.8(a) and 4.8(b). UDP Traffic is injected at each node $i \in \Omega$, with Poisson distributed packet arrivals. Figure 4.9(a) shows the expected delay versus the arrival rate under various routing policies for the network in Fig. 4.8(a). Under ExOR, packets are always routed opportunistically along the "shortest path" to the destination which results in high delay under heavy traffic scenarios. On the other hand, DIVBAR, E-DIVBAR, and D-ORCD are throughput optimal and hence, they distribute the traffic to ensure bounded average delay for all traffic rates inside the stability region. The performance gap between DI-

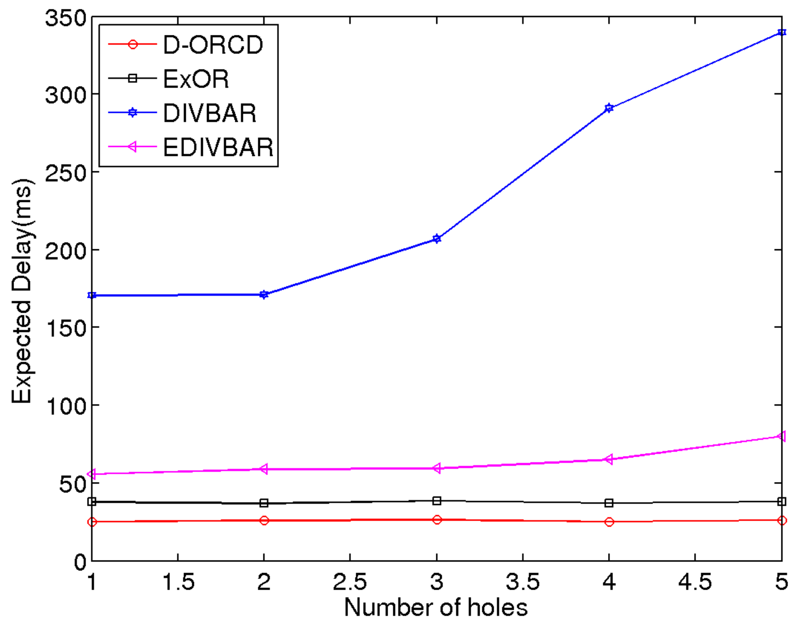
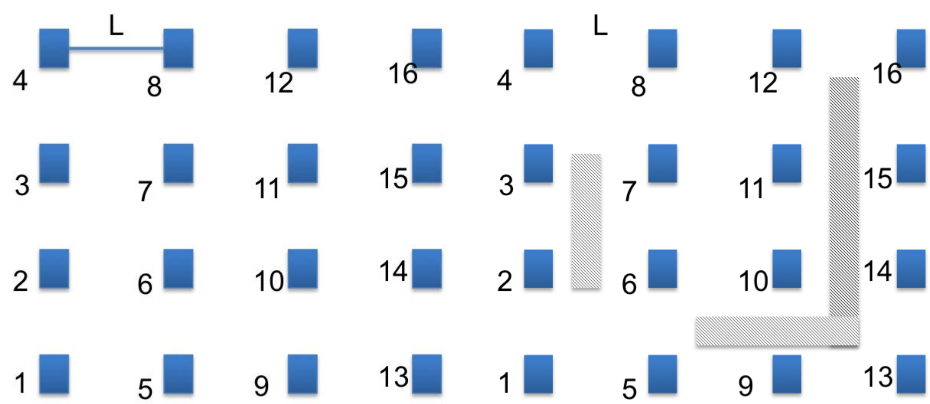


Figure 4.7: Performance for Canonical Example for λ=200 kbps



(a) Grid topoloogy. All nodes have the same arrival rate.

(b) Modifications to grid topology with blockage.All nodes have the same arrival rate, except node 10 does not generate traffic.

Figure 4.8: Grid topology of 16 nodes (4 x 4). Node 1 is assumed to be the destination

VBAR and E-DIVBAR follows from the fact that DIVBAR does not use any metric of closeness to the destination when routing the packets; while E-DIVBAR takes into account the ETX of the nodes. A more interesting observation is the comparable performance of D-ORCD and E-DIVBAR. In other words, in network 4.8(a) the mere addition of ETX and queue measures in E-DIVBAR perform sufficiently well.

Next, we consider the network shown in Fig.4.8(b), a modification of the network shown in Fig.4.8(a) in which link qualities are changed due to the existing barriers in the network. Figure 4.10(a) shows the delay performance of the candidate routing policies for this network as the traffic load varies. Again, ExOR and DIVBAR show large delay. But, unlike in the case of the network shown in Fig.4.8(a), the performance gap between D-ORCD and E-DIVBAR is now rather significant. The reason is that D-ORCD always route packets along the least congested paths to the destination (without assuming the network topology and the arrival traffic). In other words, the performance of E-DIVBAR exhibits high dependence on the underlying network topology and the arrival traffic: E-DIVBAR performs well in symmetric networks with equal arrival rate to all nodes (e.g. the network of Fig.4.8(a)), while, it performs poorly in non-symmetric networks under non-uniform traffic patterns.

4.4.4 Choice of Parameters

Next, we investigate the performance of D-ORCD with respect to the design parameters in the grid topology of 16 nodes in Figure 4.8(a). It provides significant insight to the appropriate choice of the design parameters such as the choice of partial diversity M and choice of computation cycle T.

Choice of partial diversity, M

We focus on characterizing the trade-off between performance and overhead cost for D-ORCD. We consider modifications of D-ORCD with partial diversity to decide on the number of neighbors M which acknowledge the reception of the packet. In particular, we compare the delay performance as well as the overhead cost of D-ORCD. Figure 4.11 shows the average delivery time of each packet versus the number of M for Network shown in Fig.4.8(a). Figure 4.11 illustrates the trade-off between the delay performance and overhead cost D-ORCD. We note that limiting the size of the neighbor set to 4 provides the best trade-off.

Choice of computation cycle interval T_c

D-ORCD throughput optimality as we will discuss in Section 4.5 requires that computation cycle interval be sufficiently large. However, to ensure a better delay performance, T_c must be chosen sufficiently low to make the routing decisions more responsive to the instantaneous congestion. In particular, as T_c increases, the chosen routing path i) utilizes outdated queue lengths and ii) keeps the routing policy fixed for longer durations independent of current queue-lengths. In Figure 4.12, we plot the performance of D-ORCD as T_c varies in terms of multiples of T_s. We observe that for a high load, the choice $T_c = T_s$ outperforms other values for T_c. We have chosen a more responsive version of (4.2) at the cost of provable throughput optimality, where T_c is set to $T_s = 0.5$ second.

4.5 Theoretical Guarantees

In this section, we provide a theoretical guarantee regarding the throughput optimality of D-ORCD under the assumptions that i)the flows in the network are destined for the single destination node D (for multi-destination extensions see [49]), ii) link probabilities are time invariant, iii) the routing decisions and the successful reception at set S due to transmission from node i is acknowledged perfectly to node i.

Before we precisely state the optimality, we define the notations. We define a *routing decision* $\mu_{ij}(t)$ to be the number of packets (upto 1 packet) whose relaying responsibility is shifted from node i to node j during time slot t ($\mu_{ii}(t) = 1$ means that i retains the packet). Note that $\mu_{ij}(t)$ forms the departure process from node i, while it creates an endogenous arrival to node j. Without loss of optimality, we assume that $p_{ii} = 1$ and $\mu_{iD}(t) = 1$, if $D \in S_i(t)$.

Definition 1. A *routing policy* is a collection of causal routing decisions $\cup_{i,j\in\Omega} \cup_{t=0}^{\infty}$ $\{\mu_{ij}(t)\}$.

Let $A_i(t)$ represent the amount of data that exogenously arrives to node i during time slot t. Arrivals are assumed to be i.i.d. over time and bounded by a constant A_{max}. Let $\lambda_i = \mathbb{E}[A_i(t)]$ denote the exogenous arrival rate to node i. We define $\boldsymbol{\lambda} =$

$[\lambda_1, \lambda_2, \ldots, \lambda_N]$ to be the arrival rate vector. Let $Q_i(t)$ denote the queue backlog of node i at time slot t. We assume any data that is successfully delivered to the destination D will exit the network and hence, $Q_D(t) = 0$ for all time slots t. We define $\boldsymbol{Q}(t) = [Q_1(t), Q_2(t), \ldots, Q_{D-1}(t)]$ to be the vector of queue backlogs of nodes $1, 2, \ldots, D-1$.

The selection of routing decisions under a routing policy Π together with the exogenous arrivals impact the queue backlog of node i, $i \in \Omega$ as:

$$\begin{aligned} Q_i^{\Pi}(t+1) &= [Q_i^{\Pi}(t) - \sum_{j \in \Omega} \mu_{ij}^{\Pi}(t)]^+ \\ &\quad + \sum_{j \in \Omega} \mu_{ji}^{\Pi}(t) \mathbf{1}_{\{Q_j^{\Pi}(t) \geq \mu_{ji}^{\Pi}(t)\}} + A_i(t), \end{aligned}$$

where the superscript Π emphasizes the dependence of queue backlog dynamics on the choice of policy Π.

Definition 2. Given an ergodic exogenous arrival process with rate $\boldsymbol{\lambda}$, a routing policy Π is said to *stabilize* the network if $Q_{tot}^{\Pi}(t)$ is ergodic and $\mathbb{E}[Q_{tot}^{\Pi}(t)]$ remains bounded when packets are routed according to Π. The *stability region* of the network (denoted by $\mathfrak{S}$) is the set of all arrival rate vectors $\boldsymbol{\lambda}$ for which there exists a routing policy that stabilizes the network.

Definition 3. A routing policy is said to be *throughput optimal* if it stabilizes the network for all arrival rate vectors that belong to the interior of the stability region.

Fact 3 (Corollary 1 in [13])**.** An arrival rate vector $\boldsymbol{\lambda}$ is within the stability region $\mathfrak{S}$ if and only if there exists a stationary randomized routing policy that makes routing decisions $\{\tilde{\mu}_{ij}(t)\}_{i,j \in \Omega}$, solely based on the collection of potential forwarders at time t, $\{S_i(t)\}_{i \in \Omega}$, and for which

$$\mathbb{E}\left[\sum_{j \in \Omega} \tilde{\mu}_{kj}(t) - \sum_{i \in \Omega} \tilde{\mu}_{ik}(t)\right] \geq \lambda_k.$$

We are ready to present Theorem 2 regarding the optimality of D-ORCD.

Theorem 2. *Suppose $T_c = O(D)$ and $M = D$. Then D-ORCD is throughput optimal.*

The proof of Theorem 2 is based on the Foster-Lyapunov Theorem. For completeness, the structure of the Lyapunov function and a sketch of the proof is provided in the Appendix. By Theorem 2, under D-ORCD, the average total queue backlog remains bounded. Little's theorem implies that under D-ORCD, the expected delay is bounded.

Remark 8. The assumptions for the optimality of D-ORCD could be relaxed in many cases.

1. The packet transmission on a link (i, j) is assumed to be successful with probability p_{ij}, and transmissions on links are assumed to be independent of each other. The computations in (4.2) and (A.24) can be generalized to incorporate correlated link qualities. by replacing the term $\big(\prod_{k\in S} p_{ik}\big)\big(\prod_{l\notin S}(1-p_{il})\big)$ with $P(S|i)$ in the definition of $P^{(i,d)}(t)$ and $P^{(i,d)}_{succ-k}(t)$, where $P(S|i)$ denotes the probability of the event $\{S_i(t) = S\}$. Furthermore, it is straight forward to show that the throughput optimality of D-ORCD is robust to all channel estimation errors, even though, erroneous link models, in general, can significantly degrade its delay performance.

2. In this chapter, we assumed that the network topology and the probability of successful transmissions are time-invariant. The generalization to the case of time-varying network topology with stationary transmission probabilities is straight forward [50].

4.6 Summary

In this chapter, combining the important aspects of shortest path routing with those of backpressure routing, we provided a distributed opportunistic routing policy with congestion diversity (D-ORCD). Under this policy packets are routed according to a rank ordering of the nodes based on a congestion cost measure. Furthermore, we show that D-ORCD allows for a practical distributed and asynchronous 802.11 compatible implementation, whose performance was investigated via a detailed set of QualNet simulations for practical and realistic networks. Simulations show that D-ORCD consistently outperforms existing routing algorithms in practical settings.

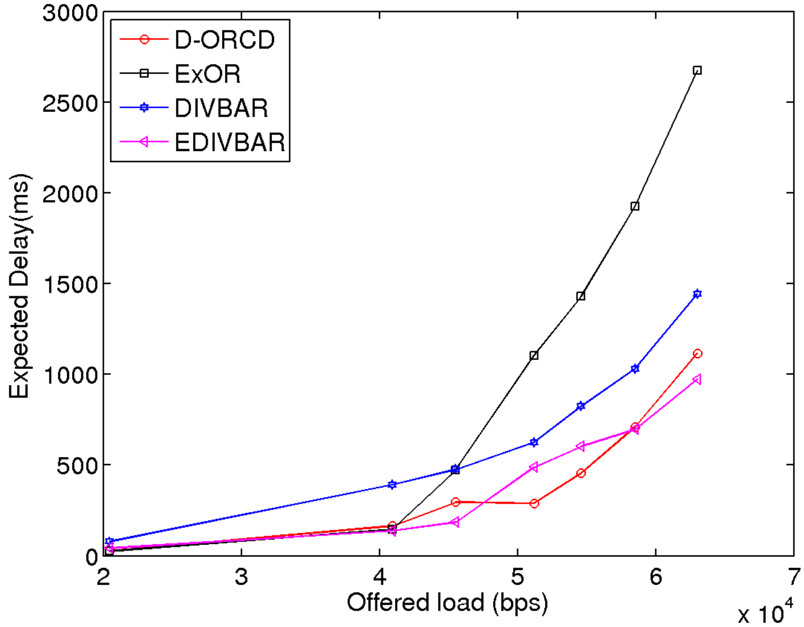
Acknowledgement

This chapter, in part, appears in the following publications.

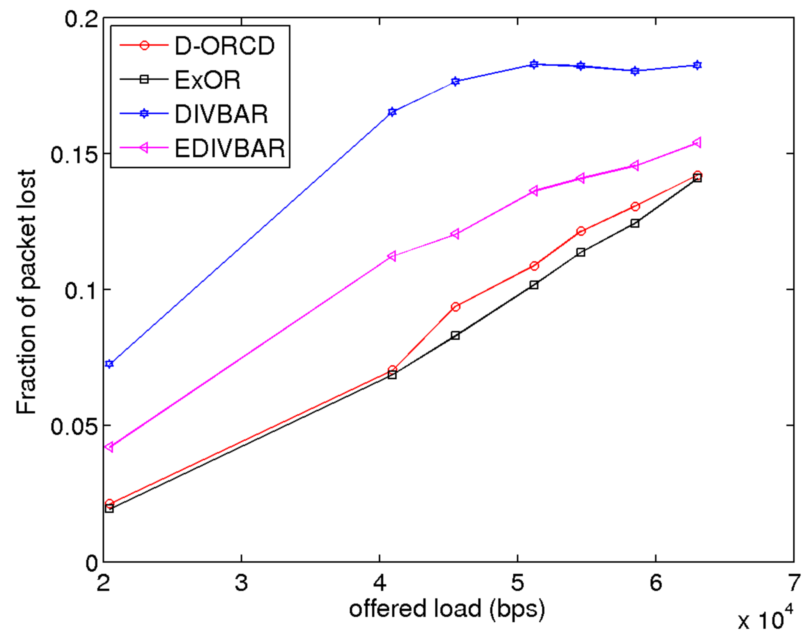
- A. Bhorkar, T.Javidi, Opportunistic routing for congestion diversity in wireless ad-hoc networks, under review IEEE Transaction on Networking.

- A. Bhorkar, A. Nilson, P. Johansoon, Common Opportunistic Routing and Forwarding, VTC10

The dissertation author was the primary investigator and author of these papers. I would like to thank my co-authors Mohammad Naghshvar, Prof. Tara Javidi, A. Nilson and P. Johansoon.

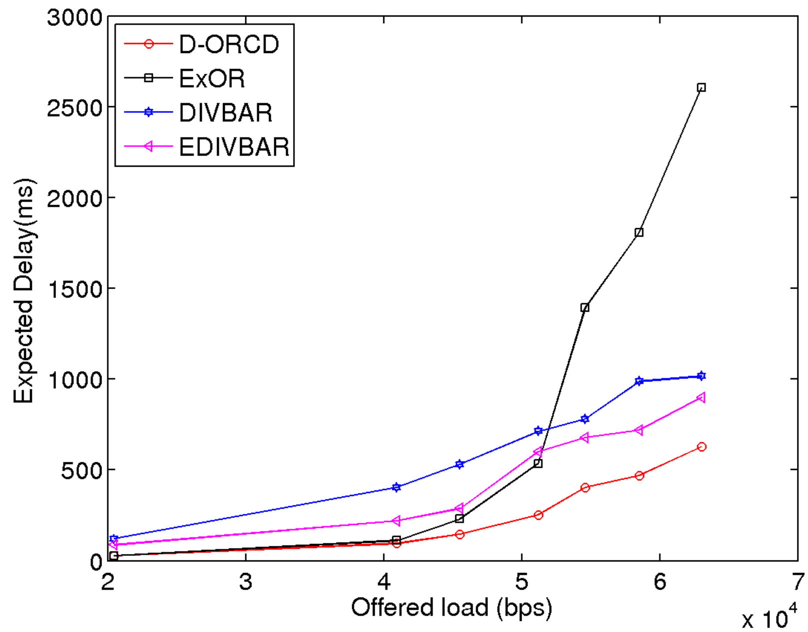


(a) Average delay per packet delivery for the network shown in Fig.4.8(a)

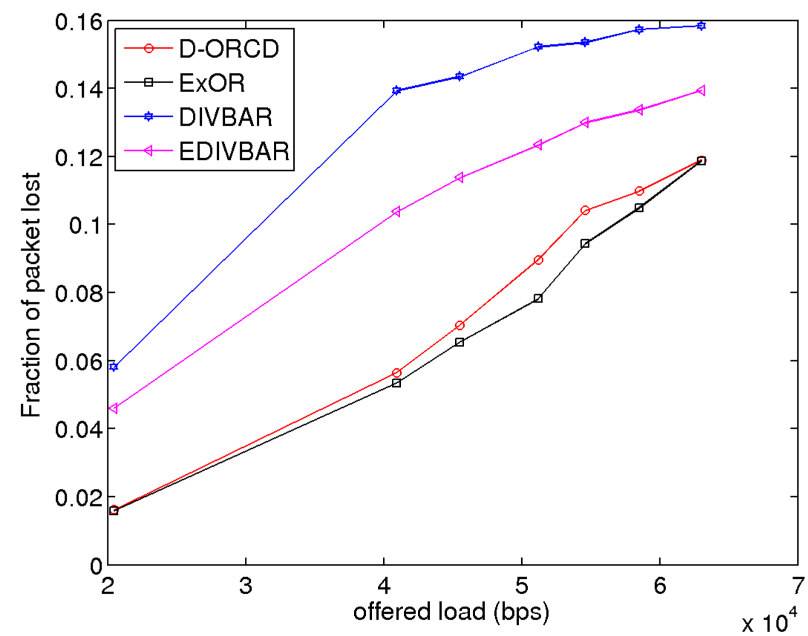


(b) Fraction of the packets lost is dominated by FO packet loss. (Packet loss due to buffer overflow is negligible)

Figure 4.9: Performance results for the grid topology.



(a) Average delay per packet delivery for the network shown in Fig.4.8(b).



(b) Fraction of the packets lost is dominated by FO packet loss. (Packet loss due to buffer overflow is negligible)

Figure 4.10: Performance results for the grid topology in Fig.4.8(b)

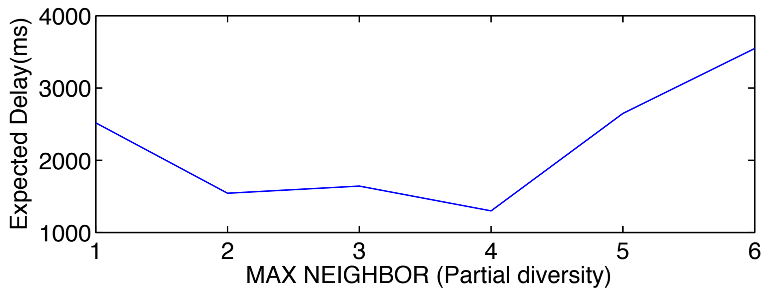


Figure 4.11: Delay performance for D-ORCD for Network shown in Fig.4.8(a)

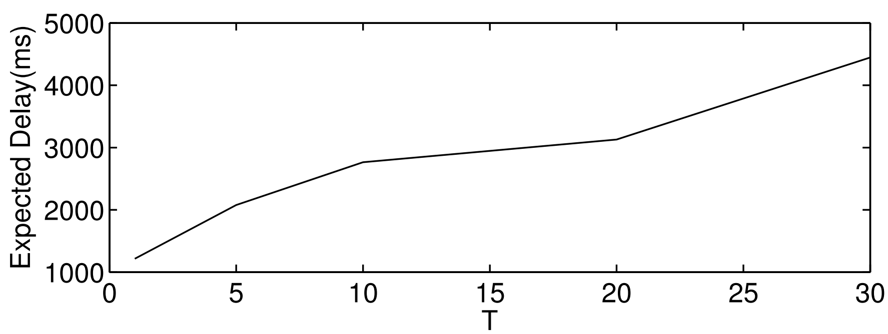


Figure 4.12: Delay Performance for D-ORCD for Fig.4.8(a) as T_c varies

Chapter 5

Congestion Diversity Protocol

Abstract

This work reports on a first comprehensive study of congestion-aware routing algorithms for wireless mesh networks. In particular, 802.11 compatible implementations of a set of congestion-aware routing protocols are compared against the state of the art shortest path routing protocol (Srcr): Backpressure (BP), Enhanced-Backpressure (E-BP) are adapted from [13, 48] suitably adjusted for 802.11 networks, while Congestion Diversity Protocol (CDP) is adapted from [1] recognizing the limitations of BP and E-BP for 802.11-based networks.

A small testbed consisting of twelve 802.11g nodes was deployed to empirically compare the performance of congestion-aware routing protocols (BP, E-BP and CDP) against benchmark Srcr. Surprisingly, backpressure-based routing algorithms (BP and E-BP) show significant degradation with respect to the Srcr benchmark for both UDP and TCP traffic. In contrast, CDP performs comparable to Srcr when the traffic is TCP or has low intensity. For medium to high load UDP traffic, CDP exhibits significant improvement with respect to both end-end delay and throughput over other protocols in 60-80% of the network configurations. The cases where CDP performs poorly are analyzed carefully. These cases are shown to be the negative side-effect of a modular approach to congestion-aware routing design in which the MAC layer is left intact. Furthermore, it is shown that these cases are indeed rare in any practical setting.

5.1 Introduction

Traditionally in communication networks, end-end rate adaptation, traffic engineering, and transport layer signaling have been widely developed to prevent network congestion. On the other hand, the routing layer is tasked to identify shortest-paths to the destination independent of the congestion in the network. In wireless context, variants of shortest-path [3–6] have been proposed, without modifications to the functionalities of the traditional layers, relying various notions distance metric to the destination. A well known example of this approach is Srcr proposed in [6], whose distance metric is based on the number of transmissions required to relay a packet. However, such a shortest path approach to routing traffic falls short in providing acceptable service in wireless networks as the traffic demand approaches the network capacity and when UDP flows constitute a significant proportion of the traffic. More precisely, shortest path solutions are susceptible to an underutilization of path diversity resulting in increased delay, congestion, buffer overflows, and queue instability. In contrast, going back to the seminal work on Backpressure (BP) routing of Tassiulas and Ephremides [14], a slew of theoretical and simulation-based studies [13, 43, 48, 50, 51] have argued for congestion-aware routing protocols: protocols that route packets using only neighbors' congestion levels [48] or overall congestion along the path [13, 43, 50] to the destination.

This work provides a comprehensive approach to the design, implementation and experimental evaluation of the congestion-aware routing against state of art routing Srcr in multi-hop 802.11-based (WiFi-based) wireless networks. The salient feature of our approach is our equal treatment of theory and experimentation in the design of congestion-aware routing algorithms. The design and the choice of the routing protocols in this study are inspired by the theoretical studies in wireless networks [13, 14, 48, 50]. We have, however, refrained from a redesign of the network at all layers and functionalities as suggested by these studies. Instead, we have devised a solution on a testbed consisting on commercially available 802.11-based wireless radios to shed light on the implications of incorporating the congestion information at the routing layer. More precisely, we have restricted our focus to 1) a low overhead implementation of the proposed protocols in the literature, and 2) a modular solution, where only the functionalities of the routing layer have been modified, leaving the physical (PHY) and the media access control (MAC) layers untouched. This pragmatic approach has allowed us to test the basic promise of the congestion-aware routing including and beyond backpressure. This work provides with the first study which carefully investigates the advantages and pitfalls of backpressure routing and a novel design for end-end delay improvements on a network consisting of inexpensive commercially available components. Secondly, our modular approach allows us to investigate the advantages of congestion-aware routing at the routing layer isolated from the benefits of generalized scheduling [14] or receiver diversity gain [12].

5.1.1 Overview of Results

In this work, we provide a comprehensive evaluation of various congestion-aware solutions in the context of a 802.11-based networks with minimal modifications to the MAC or PHY operations, the only such study to the best of our knowledge. First, we provide a low overhead practical implementation of backpressure-based algorithms BP and Enhanced-BP(E-BP) motivated by the theoretical studies proposed in the opportunistic routing context [13, 48].[1] In particular, we modify DIVBAR [48] and EDIVBAR [13], which rely on the receiver diversity and arrive at designs of BP and E-BP consistent with the widely used MAC of 802.11 wireless cards. The backpressure-based algorithms provide an important theoretical guarantee of throughput optimality (bounded expected delay for all stabilizable traffic) by effectively balancing the queue backlogs at every location. In contrast to the shortest path routing, backpressure-based routing uses differential backlogs at the nodes to make routing decisions. BP (and DIVBAR) selects the neighbor with the most negative differential backlogs at each node (in the absence of any such neighbor, the node retains the packet). Since BP ignores the distance of the potential forwarders to the destination, it can lead to a performance worse than Srcr particularly at the low traffic [1]. To address the shortcomings of BP (and DIVBAR), E-BP (EDIVBAR) has been proposed in order to integrate the two approaches [13] by linearly combining the backlog information of BP and distance information of Srcr. Motivated by a theoretical and simulation study in the opportunistic

[1] In opportunistic setting, the next hop is chosen after the receiving nodes of a packet are known at the transmitter.

routing context [50, 51], we also propose a novel alternative Congestion Diversity Protocol (CDP). CDP combines the backlog information of BP and distance information of Srcr, by estimating and utilizing the total draining draining times of nodes. The design of BP, E-BP and CDP is unified via a class of asynchronous distributed routing algorithms and corresponding congestion measures. In all these cases, the routing decisions as well as the distributed computations of congestion measures are based on an asynchronous exchange of congestion information amongst neighboring nodes. This unified approach enables a fair comparison amounts the candidates of interest.

The main contributions of our work include:

- We have implemented and studied the algorithms: Srcr, BP, E-BP and CDP. We have taken a pragmatic approach by implementing these solutions on the existing off-the-shelf embedded Alix nodes [52] at the routing layer, leaving rest of the radio functionalities untouched.
  - For TCP traffic, where the transport layer responds to the congestion, CDP shows a comparable performance with respect to Srcr. In contrast, BP improves the total throughput by significantly increasing the throughput for short flows at the cost of severe disruption to long flows and hence fairness.
  - For UDP traffic, at low traffic loads, CDP performs identical to Srcr, while BP and E-BP show poor performance due to occasional busrtiness and random walk effects in the network.[2]
  - For UDP traffic in medium to high loads, CDP reduces delay, decreases packet drop rate, and increases throughput in comparison with BP, E-BP (congestion-aware schemes) and Srcr (congestion blind scheme) in at least 60% of the scenarios.
- As a by-product of the modularity of our approach, we identify intra-flow and inter-path interference as the main potential drawback of the congestion-aware routing.

[2]The desirable performance of Srcr under low traffic indicates the sufficiency of shortest path solutions in a network with a significant gap between link capacities and ingress traffic, as it is the case with wired networks.

- We provide pathological examples where the intra-flow and inter-path interference significantly over-shadows the benefits of congestion awareness and path diversity. Furthermore, these pathological behaviors are shown to be the inevitable side effects of the modular approach to routing where the MAC layer is kept intact.
- In real networks consisting of low rate background traffic, a modular implementation is sufficient to capture the benefits of congestion diversity. In other words, we show that the pathological examples are not likely to arise in practically relevant situations.

5.1.2 Related Work

We close this section with a note on the related work. While, much experimental research has shown the value of using the differential backlog information in wireless networks for scheduling and rate allocation at the MAC layer [53], [54], [55] and the transport layer [56], there are very few experimental studies that have dealt with a practical implementation of backpressure as a routing solution with commercially available radios. For example, [55] incorporates the backlog information at all layers making it hard to come to the conclusions regarding the value of the congestion information at the routing layer. The authors in [57] have proposed Backpressure Collection Protocol (BCP) for sensor networks on top on 802.15.4, to enhance data collection. The BCP implementation requires a LIFO discipline at the MAC layer leading to significant reordering, unsuitable for various applications.

In literature, many heuristic load balancing and multipath routing solutions [50, 51, 58–62] have been proposed for wireless networks in manners that are reminiscence of our approach. To the best of our knowledge, these approaches are studied in terms of simulations and no real implementation is known. Recently, inspired by the theoretical framework of [63], Horizon [64], a practical joint load balancing algorithm with TCP rate control is implemented for 802.11-based networks. Our work supplements the design and analysis of Horizon by considering both UDP and TCP traffic for a possible class of distributed asynchronous congestion aware routing algorithms without any changes at the transport layer.

The remainder of this work is organized as follows. Section 5.2 introduces the routing algorithms, CDP, BP, E-BP and Srcr. In Section 5.3, we discuss various implementation issues for these protocols. Section 5.4 provides the performance results for UDP and TCP traffic. In Section 5.6, we analyse the performance of the congestion-aware routing algorithms closely with respect to and determine the causes of performance gains and losses by analysing various scenarios. Finally, we conclude the chapter in Section 5.7.

5.2 Routing with Congestion Diversity

In this section, we start with our 802.11-compatible design for the congestion-aware routing algorithms BP, E-BP and CDP. Next, we summarize the exiting design and our implementation for state of the art protocol Srcr.

5.2.1 Congestion-aware routing

The guiding principle of congestion-aware routing has been congestion avoidance in the network taking into account the queue backlog information $q_t^d(n)$ at each node n destined for node d at time t and the link qualities $W(n,k)$ between each pair of nodes n, k.

BP, E-BP and CDP take routing decisions by exchanging a time-varying congestion aware metric, referred to as the *congestion measure*. For a set of nodes Ω, we denote the congestion measures for destination $d \in \Omega$ at node $n \in \Omega$ as $V_{X,t}^d(k)$, where X is the protocol of interest in the set {BP, E-BP, CDP}. In practice $V_{X,t}^d(k)$ is only known at node n via periodic updates received from node k. Let $\tilde{V}_{X,t}^{(n,d)}(k)$ be the latest congestion measure advertised by neighbor k and received at node n. Based on the received congestion-measure $\tilde{V}_{X,t}^{(n,d)}(k)$, each node n in the network updates its routing table. In particular, the routing table determines the next-hop $K_{X,t}^{(n,d)}$ for a packet at node n destined for node d. After each successfully acknowledged transmission, the routing responsibility is then transferred to the next hop. Table 5.1 provides notations used in the description of these algorithms.

The design of these congestion-aware algorithms rely on a routing table at each

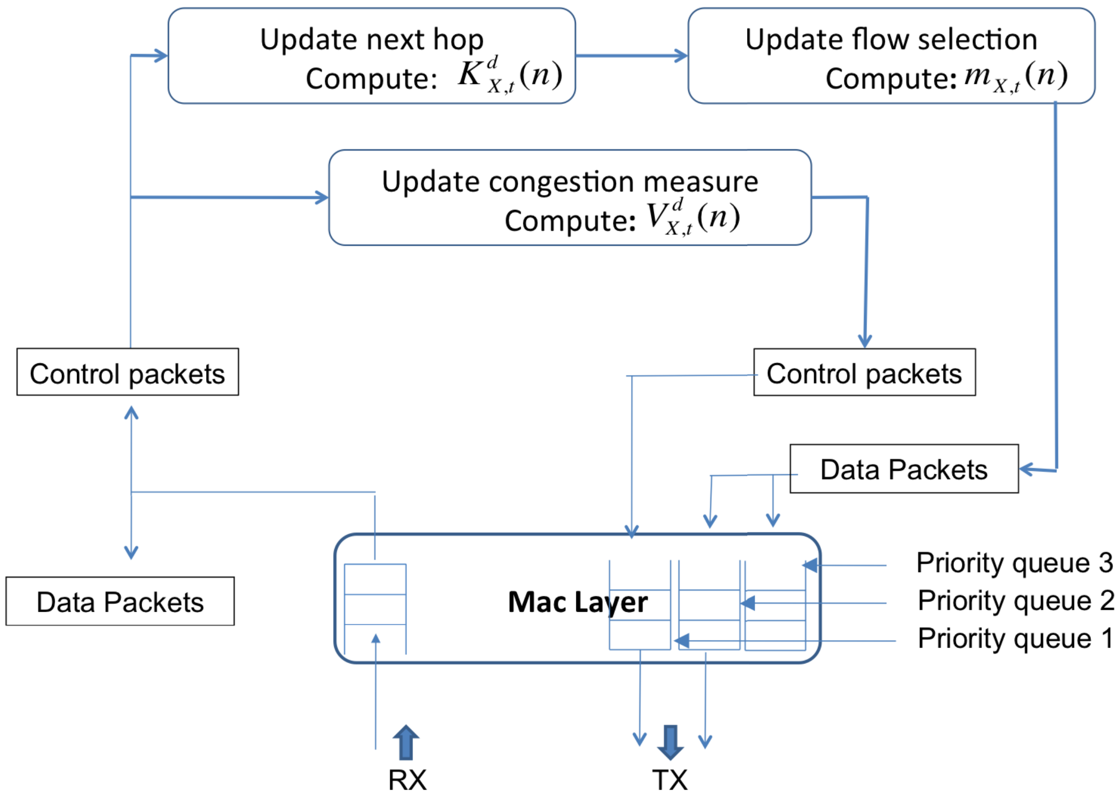


Figure 5.1: Design of congestion-aware routing algorithms.

node to determine the next best hop. The routing table at node n consists of a list of neighbors $\mathcal{N}(n)$, a structure consisting of estimated congestion measures $\tilde{V}^{(n,d)}_{X,t}(k)$ for all neighbors $k \in \mathcal{N}(n)$ associated with different destinations, and the best next hop vector $\{K^{(n,d)}_{X,t}\}_{d\in\Omega}$. Node n periodically advertises the entries of the its computed congestion measures to its neighbors at intervals of T seconds using control packets. Thus, the periodic computation and communication of congestion-measures propagates routing information across the neighbors. The sequence of operations performed are shown in Figure 5.1.

BP, E-BP and CDP have different notions of measuring the effective congestion in the network and thus determining next hop selections. Next, we detail the computations performed at each node to determine the congestion measures and next hops for BP, E-BP and CDP respectively.

Table 5.1: Notations used in the description of the algorithms

Symbol	Definition
$\mathcal{N}(n)$	Neighbours of node n
$W(n,k)$	Transmission time on the link from node n to k
$q_t^d(n)$	Queue-length at node n destined for d at time t
$ETX^{(k,d)}$	ETX from node k to destination d
$V_{X,t}^d(n)$	Congestion measure computed for protocol X at node n $X \in \{$BP, E-BP, CDP$\}$
$\tilde{V}_{X,t}^{(n,d)}(k)$	Latest congestion-measure obtained at n from node k
$K_{X,t}^{(n,d)}$	Selected relay by protocol X at node n

Backpressure Protocol (BP)

For BP, the congestion measure is simply the queue backlog information. In particular, each node advertises its current queue-backlog information for each destination as a congestion measure in the control packet. The congestion measure $V_{BP,t}^d(n)$ for node n, $n \neq d$ is given as $V_{BP,t}^d(n) = q_t^d(n)$. Thus, effectively, the estimated congestion measures $\tilde{V}_{BP,t}^{(n,d)}(k)$ at node n denotes the latest queue length information at its neighbor $k \in \mathcal{N}(n)$.

The BP then selects the next hop based on a weighted differential backlog. For any destination d, BP chooses the next hop $K_{BP,t}^{(n,d)}$, such that:

$$K_{BP,t}^{(n,d)} = \arg\min_{k \in \mathcal{N}(n)} \frac{1}{W(n,k)} \left(\tilde{V}_{BP,t}^{(n,d)}(k) - V_{BP,t}^d(n) \right). \tag{5.1}$$

The original backpressure [48] assumes a globally synchronized time-slotted MAC protocol as well as a controller that computes and schedules the nodes in centralized manner. Note that our implementation of BP is an approximate variant of the original backpressure proposed in [48] adjusted for the distributed implementation on 802.11-based networks.

Enhanced Backpressure Protocol (E-BP)

E-BP is a variant of BP, where along with the queue information, the ETX metric is used for path selection. E-BP, similar to BP, uses queue backlog information as the congestion measure and $V^d_{E-BP,t}(n) = q^d_t(n)$. Furthermore, for a packet destined for d, E-BP chooses its next hop $K^{(n,d)}_{E-BP,t}$ such that:

$$\begin{aligned} K^{(n,d)}_{E-BP,t} &= \arg\min_{k \in \mathcal{N}(n)} \Big\{ ETX^{(k,d)} \\ &+ \frac{1}{W(n,k)} \left(\tilde{V}^{(n,d)}_{E-BP,t}(k) - V^d_{E-BP,t}(n) \right) \Big\}, \end{aligned} \tag{5.2}$$

where $ETX^{(k,d)}$ is the minimum transmission time from node k to destination d defined as

$$ETX^{(k,d)} = \min_j \Big\{ ETX^{(j,d)} + W(k,j) \Big\}. \tag{5.3}$$

Note that, for E-BP, the control packet needs extra overhead to compute the ETX (see Section 5.2.2) along with the transmission of the congestion-measure.

Congestion Diversity Protocol (CDP)

The congestion measure for CDP is the aggregate sum of the local draining time at the node n and the draining time from its next hop till the destination. In CDP, when relaying packets destined for node d, node n selects the targeted receiver $K^{(n,d)}_{CDP,t}$ to minimize the packet's delivery time, i.e.

$$K^{(n,d)}_{CDP,t} = \arg\min_{k \in \mathcal{N}(n)} \Big\{ W(n,k) + \tilde{V}^{(n,d)}_{CDP,t}(k) \Big\}. \tag{5.4}$$

Assuming a FIFO discipline at layer-2, we proceed to describe the computations of congestion measure for CDP. The congestion measure associated with node n for a destination d at time t is the aggregate sum of the local draining time at node n and the estimated draining time from its next hop, $\tilde{V}^{(n,d)}_{CDP,t}(K^{(n,d)}_{CDP,t})$. The local draining time for a packet destined for d arriving at n at time t is equal to the duration of the time spent draining the packets that arrived earlier plus its own packet delivery time. In other words, if $q^j_t(n)$ is the number of packets destined for j queued at node n at time t, the local draining time is equal to

$$\sum_{j \in \Omega} q^j_t(n) W(n, K^{(n,j)}_{CDP,t}) + W(n, K^{(n,d)}_{CDP,t}). \tag{5.5}$$

The congestion measure for node n, $n \neq d$ is then given as

$$V^d_{CDP,t}(n) = W(n, K^{(n,d)}_{CDP,t}) + \sum_{j \in \Omega} q^j_t(n) W(n, K^{(n,j)}_{CDP,t}) + \tilde{V}^{(n,d)}_{CDP,t}(K^{(n,d)}_{CDP,t}). \quad (5.6)$$

Thus, in CDP, the congestion measures are computed in a fashion similar to distributed Bellman-Ford computations [12].

5.2.2 Shortest path routing: Srcr

For the sake of completeness, we describe the design of congestion unaware, shortest path routing protocol Srcr. This implementation acts as a benchmark for the comparison purposes with respect to the congestion-aware routing. Srcr proposed in [6] uses the ETX metric when routing the packet considering only the link quality information at the nodes. The ETX to reach the destination is computed using the transmission duration $W(i, j)$ between each pair of nodes i and j. Specifically, for a packet destined for node d, the next hop $K^{(n,d)}_{Srcr,t}$ is chosen such that

$$K^{(n,d)}_{Srcr,t} = \arg\min_k ETX^{(k,d)} + W(n, k), \quad (5.7)$$

where $ETX^{(k,d)}$ is the minimum transmission time from node k to destination d computed as (5.3). We use the distributed architecture of CDP for the calculation of ETX metric by taking $q^d_t(n) = 1$ in the calculation of $V^d_{CDP,t}(n)$.

In the next section, we discuss the practical issues associated with computation of the congestion measures for these congestion-aware routing algorithms. Furthermore, we propose practical implementations and heuristics.

5.3 Implementation Details

In this section, we provide the elements of the congestion aware routing responsible for the computation of the congestion measure including reliability of control packets, link quality estimation, neighbor discovery, flow selection, and avoidance of loops while routing.

5.3.1 Control Packet Reliability

An important component of the implementation is the exchange of congestion-measures using control packets. In particular, BP, E-BP and CDP depend on a reliable, frequent, and timely delivery of the control packets. As documented in [45], the loss of control packets can be the cause of instability in many well known routing algorithms. Thus, it is important that our implementation ensures a high reliability for the delivery of control packets. In our implementation, we have taken advantage of the priority-based 802.11e to implement this component of the control plane. In other words, the MadWifi scheduler assigns highest strict priority to the control packets. It reduces the probability of control packet drop at the MAC layer and also ensures their timely delivery. In the context of 802.11e, we utilize two of the four priority queues: data packets are assigned to the lower priority queue (WME-AC-BK) and control packets are assigned to the higher priority queue (WME-AC-VO) [65]. In our implementation, the scheduler assigns a sufficiently reliable and low PHY rate (11 Mbps in our testbed) for the control packets. These design choices ensure high reliability and speedy delivery of the control packets.

5.3.2 Link quality estimation

The computations given by (5.1)-(5.7) utilize the transmission time $W(i,j)$ for each pair of nodes i,j. In this work, we measure the transmission time $W(i,j)$ by taking the difference between the instant when a packet enters the hardware at node i and when the acknowledgement is received from node j at node i.[3]

The transmission time $W(i,j)$ for a packet from node i transmitted to node j can be written as the interval between the transmission of the packet from node i's interface and the reception of an acknowledgement (ACK) at node i from node j. The available drivers and hardware differ in functionalities to accurately measure the transmission time (up to the accuracy of the operating system scheduling delay). Specifically, in the context of atheros cards, we will detail the computation of $W(i,j)$. For atheros-based

[3] Different hardware-independent proxies for transmission times exists in the literature and these can be of interest for easy implementation. For example, link success probabilities or the number of retries [66] can be used to obtain approximate transmission times.

wireless interfaces and the MadWifi driver used in our experiments, we can easily determine the instant of time when i receives an ACK from j. Now, we are left to determine the time instant at which packet enters the interface. Unfortunately, the MadWifi driver does not provide the time of entry at the interface directly. Instead, the driver provides information on when the packet enters the queue at the MAC layer. With such limited information at hand, we devise the following algorithm to determine drain time. Timestamp T_1 when the packet enters the MAC queue, timestamp T_2 when the previous packet exits the interface and timestamp T_3 when the current packet exists the interface. The transmission time is then given as $T_3 - \max(T1, T2)$.

At the system level, we combine the link quality measurements using actual data packets at the nodes (passive probing) with dedicated probe packets transmitted to each neighbor when a node does not engage in data transmission (active probing). These estimates are combined using a weighted average.

5.3.3 Neighbor Discovery

Each node needs to maintain information on the cost vectors along with the link quality information for all of its neighbors to efficiently route the packets. In order to reduce the overhead, we restrict the set of neighbors by using a sufficiently reliable link. Specifically, the neighbors are defined so that the link success probability for each neighbor is above a threshold γ. Defining neighbors using a delivery ratio eliminates any artifacts due to external interference. Out of the many possibilities to implement this procedure, we have used dedicated probe packets to obtain the relevant information.

5.3.4 Loop Avoidance

Unless carefully designed, distributed computations of any time-varying distance vector routing algorithm are likely to suffer from the classical problem of counting to infinity [46]. Looping can result in large delays, increased interference and loss of packets. The problem is most acute when there is a sudden burst of traffic,[4] resulting in a transient build-up of queue and an overestimation of the quality of the route. Such

[4] Similar to the broken link scenario in a typical distance vector routing.

transient effects can be severe due to the resulting slow exchange of control packets.

To address this issue, in case of CDP, we utilize Split-horizon with Poison reverse solution [47] to avoid loops. CDP uses control packet information received from neighbors to gather appropriate information for the split horizon implementation. In Split-horizon with poison reverse, a node advertises routes as unreachable to the node through which they were learned. Note that, we apply loop avoidance methods only for CDP, while BP and E-BP are left in their original form in [13].

5.3.5 Flow Selection

In our design, we have implemented the flow selection criterion proposed in [13, 48], for BP and E-BP. This means that both BP and E-BP perform flow selection among the packets associated with different destinations using virtual queue mechanism at layer-2. In particular for protocol X, $X \in$ BP, E-BP, each node n at time t selects the oldest packet destined for node $m_{X,t}(n)$ among available packets and transmits the packet to the PHY layer. BP performs a flow selection by selecting destinations with minimum queue differentials, i.e.

$$m_{BP,t}(n) = \arg\min_{d} \min_{k \in \mathcal{N}(n)} \frac{1}{W(n,k)} \left(\tilde{V}_{BP,t}^{(n,d)}(k) - V_{BP,t}^{d}(n) \right). \tag{5.8}$$

On the other hand, E-BP selects destination which minimizes sum of queue differential and ETX, i.e.

$$\begin{aligned} m_{E-BP,t}(n) &= \arg\min_{d} \Big\{ \min_{k \in \mathcal{N}(n)} ETX^{(k,d)} + \frac{1}{W(n,k)} \\ &\times \ (\tilde{V}_{E-BP,t}^{(n,d)}(k) - V_{BP,t}^{d}(n)) \Big\}. \end{aligned} \tag{5.9}$$

In our implementation, we approximate the above packet level flow selection in (5.8) and (5.9) using the priority queueing available on the atheros cards. In order to implement priority scheduling, we utilize the 802.11e-based priority scheduler [52] at the MAC layer and the highest destination packet ($m_{BP,t}(n)$ or $m_{E-BP,t}(n)$) is assigned to the higher priority hardware queue.

CDP takes a simpler approach in flow selection. In particular, rather than using virtual queues for different destinations, CDP uses a simplistic FIFO discipline at layer-

2 for all packets. We have found that this simple approach is satisfactory for the CDP's performance.

5.4 The Experimental Setup

Our testbed consists of 12 wireless Alix [52] nodes with 512 MB RAM and a 500 MHz processor on Linux version 2.6.22. The nodes placed as shown in Figure 5.2, are distributed in Atkinson Hall, UCSD, in about 215,000 sq. ft. of space. Each node is equipped with an Atheros-based 802.11 a/b/g wireless interface (AR 5213) connected with omni directional antennas. All nodes are connected to Ethernet ports for maintenance and data collection. All nodes are configured in the 802.11g ad-hoc mode with RTS/CTS disabled and the transmission power is set to 13 dBm. In addition to human inhabitants, the building contains hundreds of workstations and a large variety of electronics operating in the same 2.4 GHz unlicensed frequency band as 802.11, resulting in a highly variable channel quality in different portions of the building and during different times of the day. For consistent data, we performed our experiments during the night when the variability of the channel is least.

All of the above routing algorithms have been implemented in user space with appropriate calls to the MadWifi driver, which is supported by the Linux kernel (2.6.22 onwards). These algorithms perform queuing and scheduling on every packet being transmitted or received by the driver. We have used a transmission rate of 11 Mbps for the control packets while the data packets are sent at 48 Mbps. The packet size for data packets is 512 bytes. For each algorithm, each iteration of traffic generation is executed for 180 seconds.

We study the choice of the parameters in the subsequent analysis. Specifically, we need to set the following parameters: control packet interval T, probe interval duration, and neighbor probability threshold γ.

1) Choice of control packet interval T: In our setup, we transmit control packets periodically at intervals of 200 ms. Each control packet of roughly 200 bytes is broadcasted at a rate of 11 Mbps.

We need to tradeoff the overhead of the control packets and the need to obtain the

Figure 5.2: Testbed: Node locations

accurate congestion measures from neighbours. Since broadcast packets do not undergo a backoff mechanism, the broadcasting cost for control packets is negligible compared to data transmission. In order to study the interference effects of control packets on the performance, we vary the control packet transmission interval for Srcr and compare the performance with various intervals of 50, 100, 200, 300 and 500 ms. We observe that the performance does not vary for intervals greater than 200 ms, which implies that the control packets do not have a significant effect on the performance.

2) *Choice of link probe interval:* In our setup, we transmit probe packets at regular intervals to probe the channel and learn the link quality. The choice of probing interval should trade-off the added overhead with the ability to track channel variations. We set the probe interval to 1 second in accordance with the value chosen in [10, 67] consistent with indoor environment's fading parameters. Furthermore, probe packets of length 512 bytes are selected to match the data packet size (also set at 512 bytes).

3) *Choice of probability threshold γ:* We have chosen $\gamma = 0.4$ for trading off link reliability with network connectivity. The decision whether a node is a neighbor, is

based on the initial non-interfering condition of the network.

With the above setup, we are ready to evaluate the performance of congestion-aware routing protocols for TCP and UDP traffic as reported next.

5.5 Performance Results

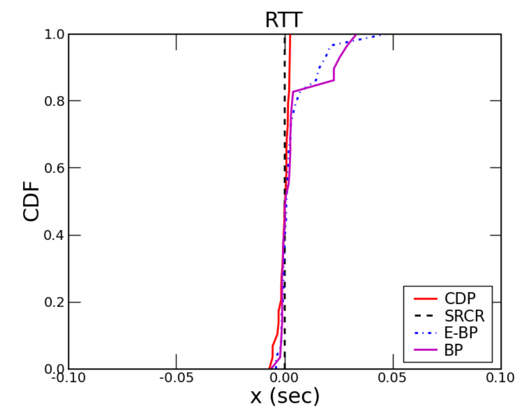
In this section, we perform a comparative study of various routing protocols under TCP and UDP traffic. In our comparative analysis, we investigate the following performance measures:

1. End-End delay: For M packets, we define the mean delay $D = \frac{1}{M}\sum_{j=1}^{M}(\tau_A^j - \tau_D^j)$, where τ_A^j is the arrival time at the destination and τ_D^j is the departure time for packet j at the source. For TCP traffic, we consider mean delay as mean Round Trip Time (RTT). We are interested in the distribution of per packet delay, e.g. the Cumulative Distribution Function (CDF) of D with respect to the random choice of network topology. For illustration purposes, we consider a differential delay measure which consists of the difference between CDP and the candidate protocol. Specifically, we plot the difference $D_{candidate} - D_{Srcr}$, where D_{Srcr} is the mean delay for CDP and $D_{candiate}$ is the mean delay for the comparative protocol.

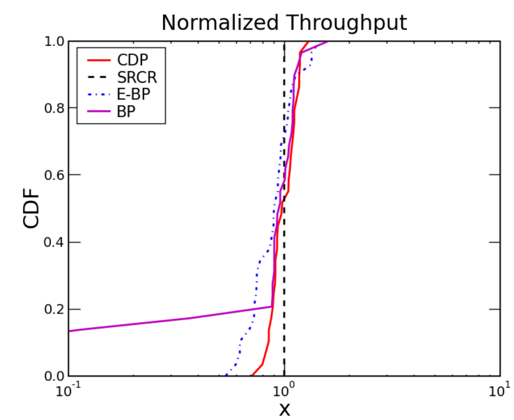
2. Throughput ratio: The throughput is the number of bytes received at the destination for the duration of the experiment. Again when investigating the CDP performance with respect to the network topology, we use the normalized throughput ratio as a measure of performance, where the normalized throughput ratio is defined as the ratio between the throughput of the candidate protocol versus the Srcr.

5.5.1 Experiments with TCP

In this section, we study the performance of congestion-aware routing algorithms for TCP used for reliable communications. We report the comparative performance of the candidate routing protocols (relative to Srcr) under reliable transfer control algorithms TCP-Veno [68] by selecting a configuration of two TCP flows with randomly selected source and destination pairs. We do not expect to see significant improvement for congestion-aware routing protocols with respect to Srcr when the sources of traffic are TCP. The first reason for the insignificant performance gain is that the current implementations of TCP are non-aggressive. TCP tries to avoid congestion in the network and thus makes the congestion routing insensitive to congestion routing. Secondly, TCP is known to have performance degradation with respect to the packet reordering. (Figure 5.4 shows that BP, E-BP and CDP suffer from the reordering of packets introduced by the dynamic route selection.)



(a) Differential RTT



(b) Normalized throughput

Figure 5.3: Performance for TCP traffic

Given the above considerations, the best we can hope is that the proposed congestion aware routing does not degrade the performance. We show that this is the case for CDP and E-BP. Figures 5.3(a) and 5.3(b) compare the CDF of the Round Trip Time

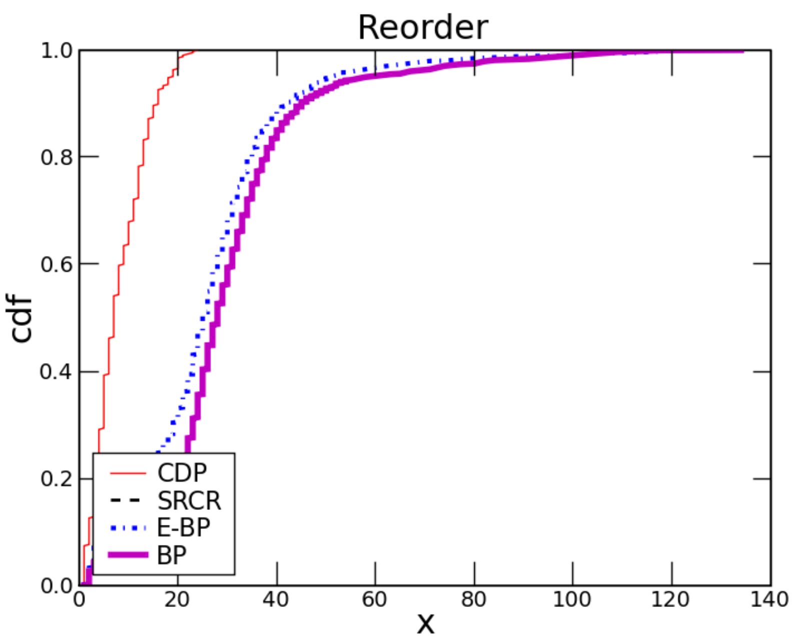


Figure 5.4: CDF of reordering

(RTT) and the normalized throughput for multi-hop flows for CDP, Srcr, BP and E-BP. These set of experiments show that CDP and E-BP exhibit a comparable performance with Srcr for TCP flows. For multi-hop flows, the performance of TCP for BP suffers from the loops and "dead ends", resulting in timeouts and very low throughput for routes with multiple hops.

Next, we dissect and study the more interesting case of UDP traffic.

5.5.2 Experiments with UDP

In this section we report on the performance of BP, E-BP, CDP, and Srcr in our network with two randomly selected flows with two sets of source-destination pairs. We inject Poisson traffic at each source node with a randomly selected average load between 0 Mbps and 7 Mbps. We repeat the experiment for 100 such configurations. Among these random experiments, we consider configurations for comparison where at least one algorithm delivers 80% of packets. (15% of the scenarios were overloaded). Furthermore, uninteresting cases with non-overlapping single hop routes are also excluded from the analysis (15% of the scenarios contained only single hops). Figures 5.5 plots the offered loads for the remaining configurations in our analysis. We classify the configurations under considerations into two sets: low load configurations when the observed delay for Srcr is sufficiently small (less than 0.1 second) and high load constituting the remaining configurations. We report the Cumulative Distribution Function (CDF) of the performance metrics for various protocols for low and high load scenarios in Figures 5.6-5.9.

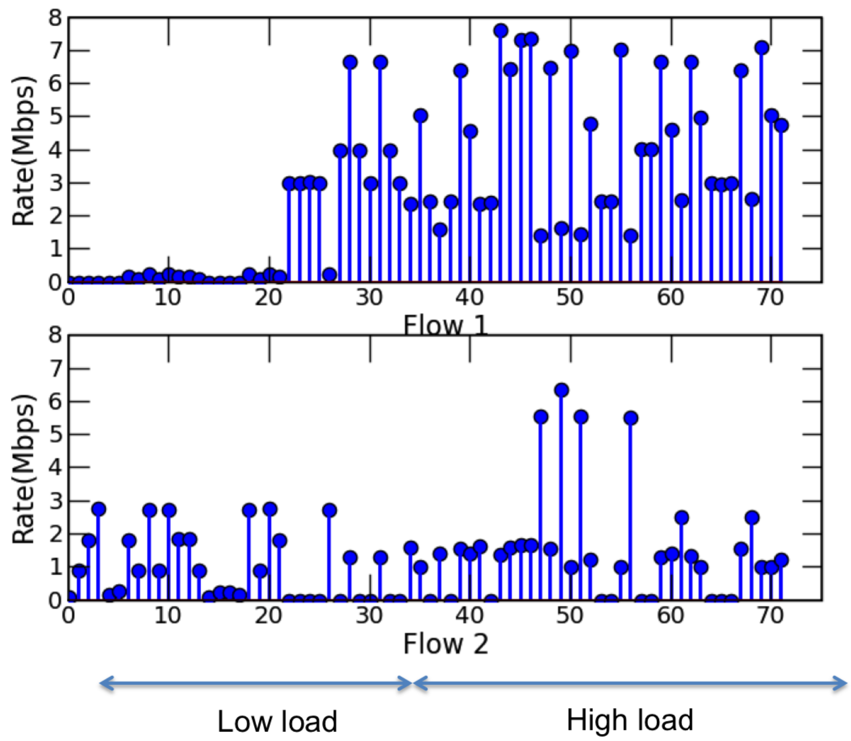


Figure 5.5: Offered load

In Figures 5.6 and 5.7, we plot the delay differential and normalized throughput for the candidate protocols under a low load scenario. Here, as we expected, CDP performs similar to Srcr while performing significantly better than BP and E-BP. This is because in absence of congestion, the distributed computation in (5.6) reduces to computation of the ETX, while BP and E-BP reduce to a near random walk in the network.

Figure 5.8 compares the CDF of the delay differential, while Figure 5.9 shows the CDF of the normalized throughput ratio for high traffic load. Figures 5.8, and 5.9

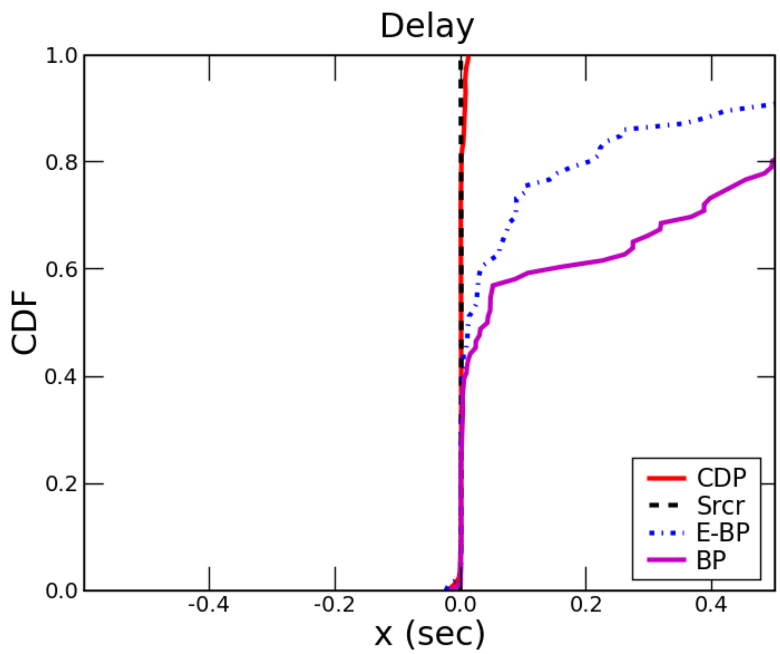


Figure 5.6: CDF of delay differential for a low load

show that for about 60% of the network configurations selected, CDP delivers packets with significantly less delay compared to other protocols[5]. Figures 5.8 and 5.9 also show CDP performs worse than Srcr in 30% of the scenarios. In Section 5.5.3 we dissect and isolate the sources of loss in these scenarios.

5.5.3 Congestion awareness : Pros and Cons

In this section, we investigate the root causes of CDP's performance losses and gains as illustrated in Figures 5.8 and 5.9. To gain insight on the strengths and weaknesses of congestion-aware routing, we consider two example topologies associated with delay differentials on both ends of the spectrum on the CDF plot in Figure 5.8. The performance of these examples with respect to Srcr is marked as point A and point B respectively in Figure 5.8. The configurations associated with these points A and B are shown in Figure 5.10. Our first example topology (point A) consists of one high (4 Mbps) unicast flow between nodes 10-17 and another low load flow (1 Mbps) between

[5]The candidate protocol performs poorly if the CDF lies to the left of the Srcr

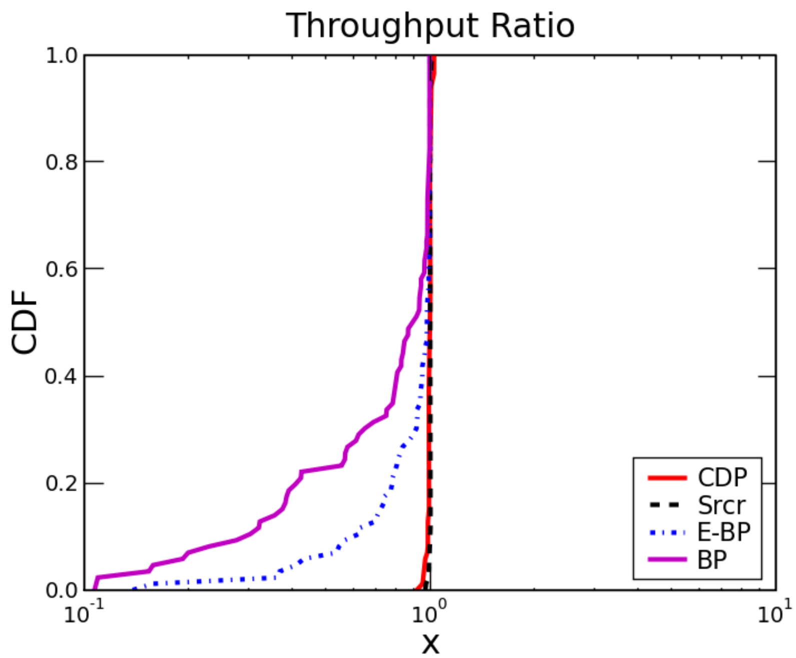


Figure 5.7: CDF of normalized throughput for low load

nodes 14-16. In this setting, the delay performance of CDP is significantly worse than Srcr. The second topology consists of the same flows 10-17 and 14-16 but flow rates of 1 Mbps and 4 Mbps are swapped. Note that this second topology coincides with the case when the CDP significantly outperforms Srcr.

Congestion-aware Routing: Cons (Topology A)

We study the flow configuration at point A consisting of flow 10-17 with high load and 14-16 with low load. Figure 5.11 plots the end-end delay for the individual flows. We observe that Srcr performs best, while the delay for CDP is 100 ms poorer than Srcr. The delay performance of BP and E-BP is order of seconds.

To understand the sources of loss of performance under congestion-aware routing policies we have illustrated the next hop selections by node 10 in Figure 5.12 where we plot $K_{X,t}^{(10,17)}$ under the candidate protocols throughout the duration of the experiments. We observe that Srcr maintains node 14 as the next hop in a static manner while CDP switches its next hop from node 14 to node 16. BP and E-BP forward significant number of packets into nodes 5, 7 and 11 increasing the interference in the network.

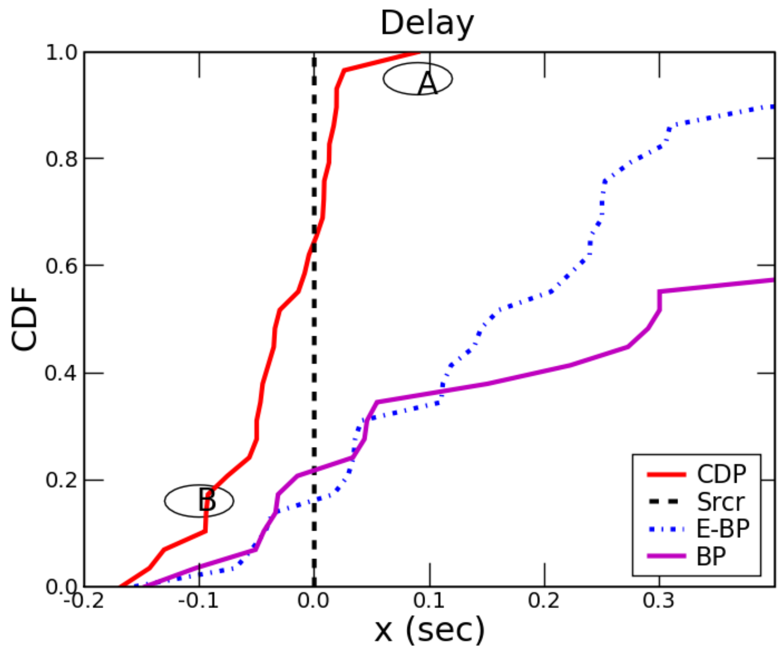


Figure 5.8: CDF of delay differential for high load

Figure 5.13 decomposes the loss incurred by each protocol into buffer overflow, retry loss and loop loss.[6] In this example, we observe flow 10-17 suffers a higher queue loss for CDP compared to Srcr. Other losses remain fairly negligible (even though higher for BP and E-BP).

Congestion-aware Routing: Pros (point B)

We now study the flow configuration at point B consisting of flow 10-17 with low load and 14-16 with high load. Figure 5.14 plots the end-end delay for the individual flows. The delay performance under CDP shows a improvement over the other candidate protocols by at-least 100 ms.

Figure 5.15 shows the next hop selections by node 10. We observe that Srcr persistently relies on routing via node 14, resulting in severe congestion and packet drops for the flow 10-17, reducing the throughput and increasing the delay. BP and E-BP still forward packets to nodes 5, 7 and 11 resulting in increased interference.

[6]Loop loss occurs due to the presence of loops in routes resulting in packet drop if the Time to Live (TTL) value reaches 0.

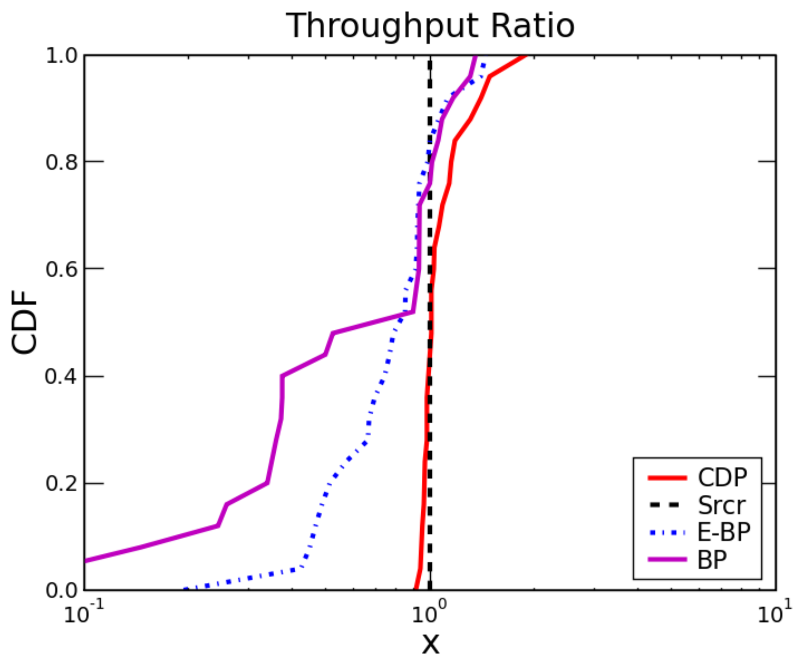


Figure 5.9: CDF of normalized throughput for high load

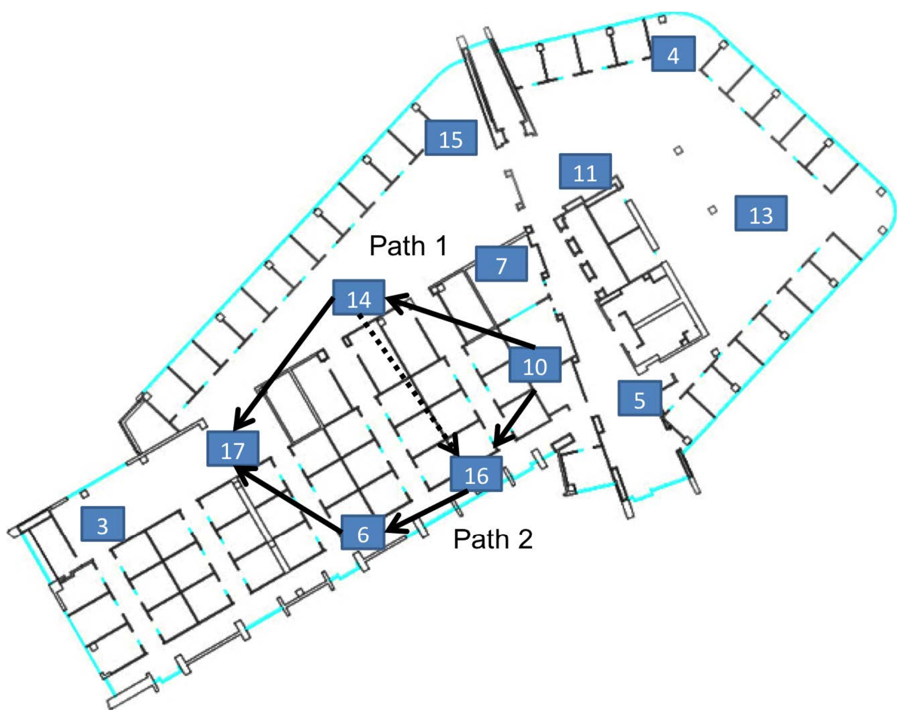


Figure 5.10: Topology used to analyse the performance results.

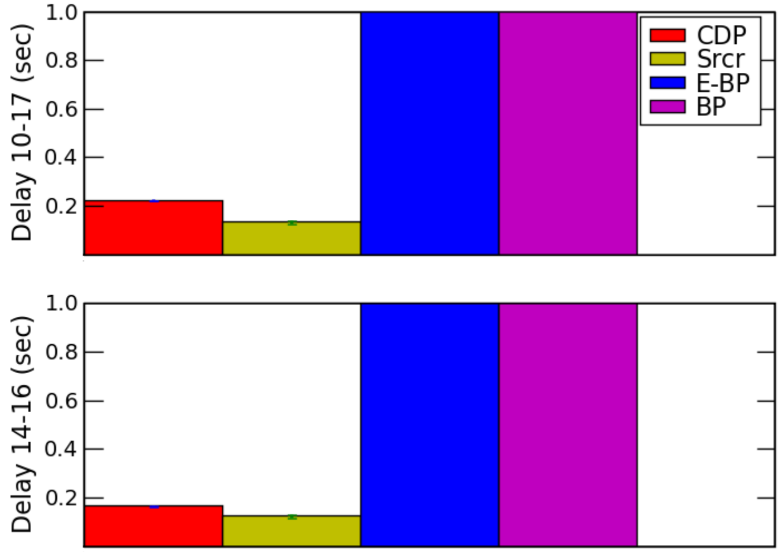


Figure 5.11: Mean delay (point A)

Figure 5.16 shows the decomposition of sources of loss under each protocol. In contrast to the previous example, the number of packet drops for flow 10-17 is significant for Srcr, E-BP and BP i.e. 20%, 25% and 40% respectively; while the packet loss under CDP is less than 1% (mostly due to buffer overflow). The retry losses and loop losses for all protocols are negligible.

The overall performance gain of CDP over Srcr in this network Topology B, suggests that the multi-path load balancing and congestion avoidance inherent to CDP can result in significant performance gains. However, the overall performance degradation in the network topology A suggests that the gains can be overshadowed by the intra-flow and inter-path interference.

5.6 Modular Approach and Multi-path Diversity

In this work, we have taken a modular approach of separating MAC from routing in designing congestion-aware routing. It is believed that a modular approach can be a significant limitation in the wireless networking setting. In this section, we relate the

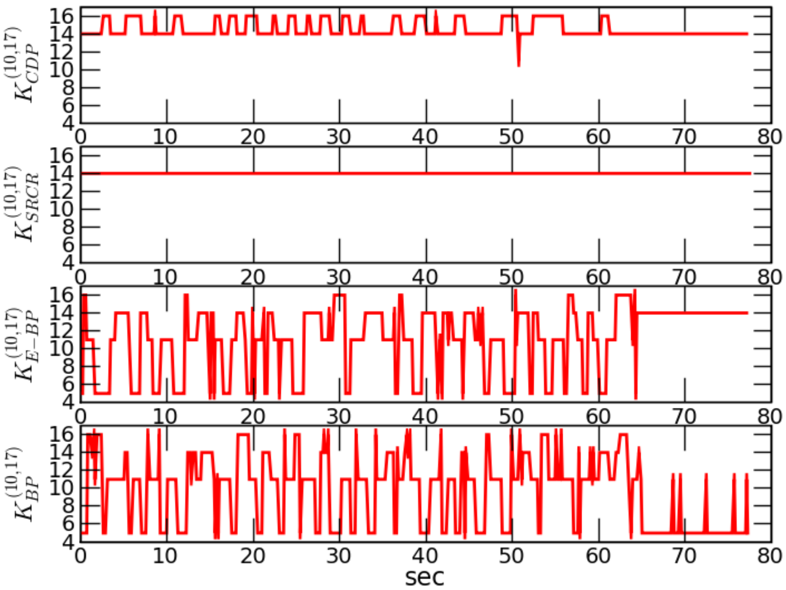


Figure 5.12: Routing paths taken by node 10 for the flow 10-17 (point A)

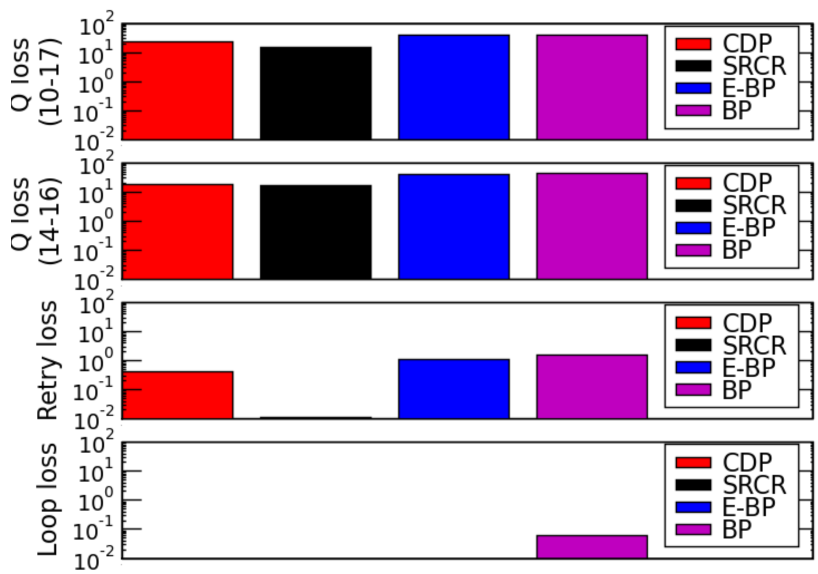


Figure 5.13: Loss decomposition percentage (point A)

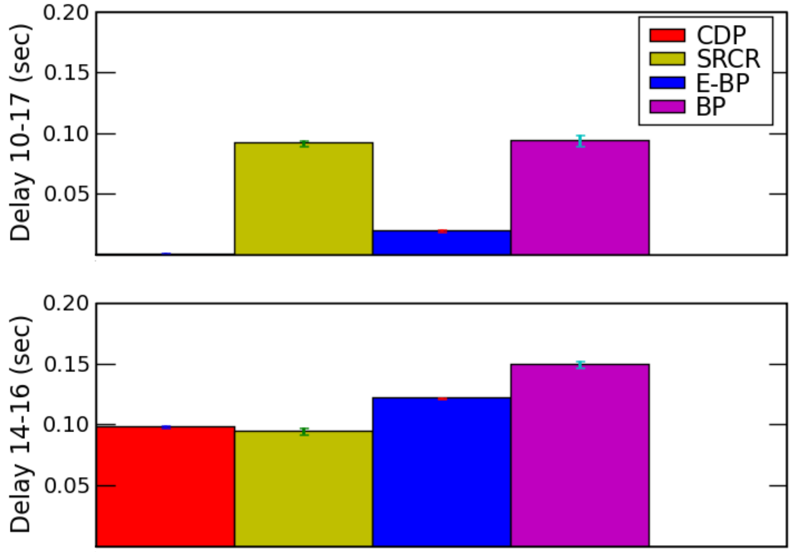


Figure 5.14: Mean delay (point B)

loss of the performance in the network topology A, suffering from infra-flow and inter-path interference to the modularity of our solution. As such, we show that,

- The degraded behavior is an inevitable side effect of the modular approach to routing where MAC layer is kept intact.
- In real networks consisting of low rate background traffic, a modular implementation is sufficient to capture the benefits of congestion diversity.

5.6.1 Case studies

In this section, we consider a set of test topologies which contain a single flow possibly routed using two paths. We analyse the effect of multi-path and congestion diversity using a set of randomized protocols where the packets at the source node is routed via Path-1 with probability α. We then compare the delay under CDP as well as the proposed randomized algorithms as α is varied from 0 to 1.

Next, we consider the examples in Figure 5.17: example 1 and 2, consisting of a single flow 10-17, with the difference in the quality of link 14-17. In example 3, we

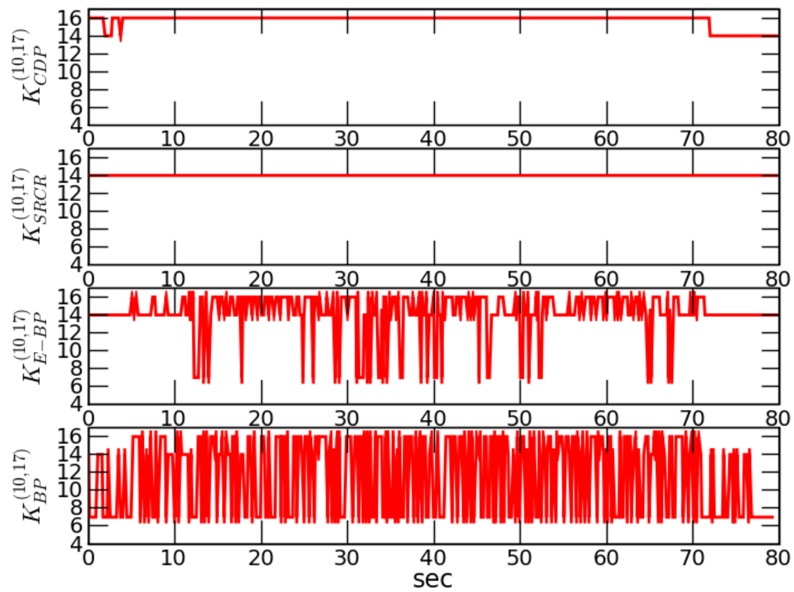


Figure 5.15: Routing paths taken by node 10 for the flow 10-17 (Topology B)

consider the longer flows 13-3 shown in Figure 5.17. Note that in these examples, Srcr is a trivial case of our class of randomized policies with $\alpha = 1$.[7]

Example 1

We analyse the topology in Figure 5.17 for a high load of 4 Mbps for flow 10-17.[8] Figure 5.18 shows the end-end delay performance for CDP and the randomized routing policy for various quantities of α. As seen in the network topology A in Section 5.5, the best performance is achieved at $\alpha = 1$ (hence Srcr). The reason for this observation is that any simultaneous utilization of Path-1 and Path-2 results in nodes 10, 14, 16 and 6 contending for the wireless channel concurrently. Hence, the gain achieved by reducing the congestion and load balancing Path-1 and Path-2 is not significant enough to compensate for the increase in the channel access times and interference.

[7]Note that we have not analysed the performance of BP and E-BP due to their consistently poorer performance.

[8]This topology is similar to the topology for topology A in Figure 5.10.

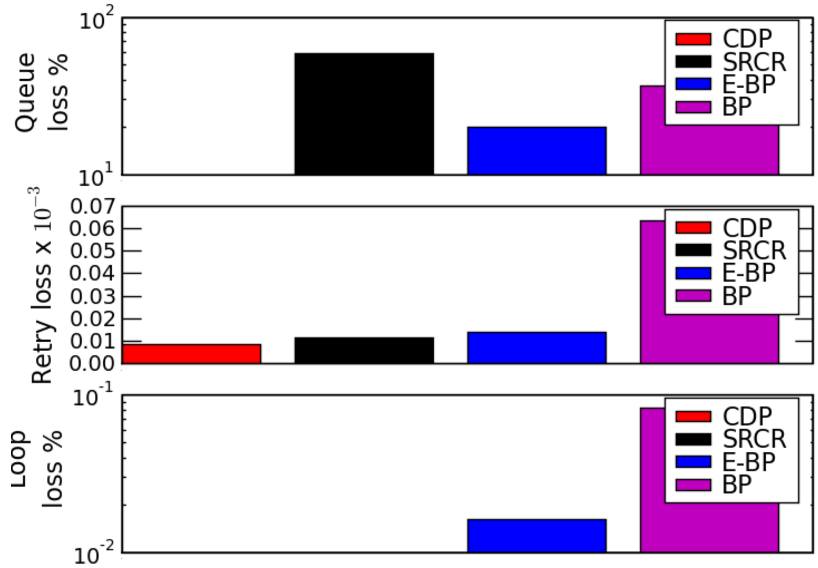


Figure 5.16: Loss decomposition percentage (point B)

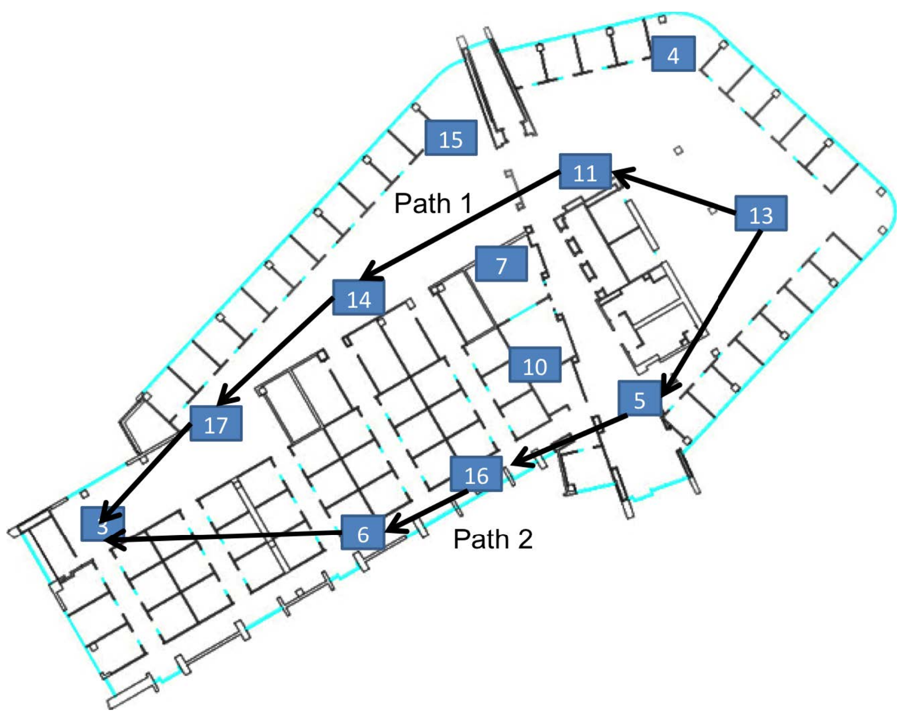


Figure 5.17: Topology used to study the gains of multipath diversity

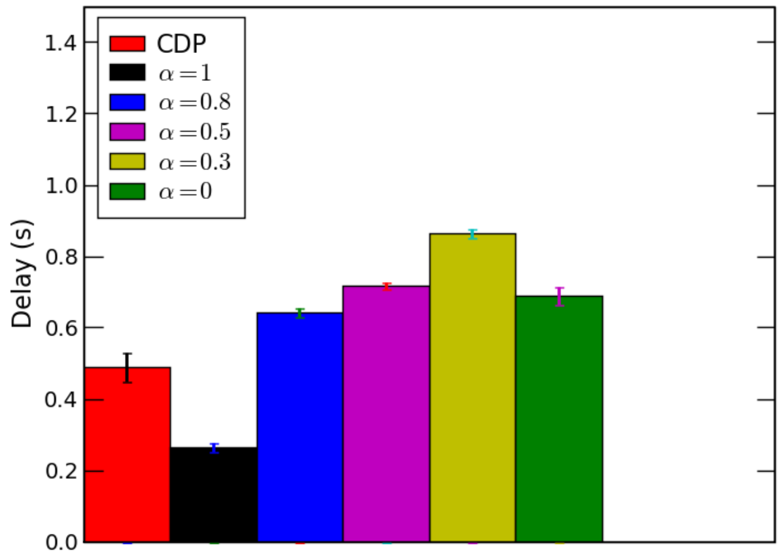


Figure 5.18: Delay performance as α is varied for the topology in Figure 5.10

Example 2

We consider the same topology as example 1, however link 14-17 is low in quality. In practice, low link quality for 14-17 occurs when there is an obstruction between 14-17. Figure 5.19 shows the delay performance for the set of routing protocols as α is varied. It is not surprising that the delay is minimized when $\alpha = 0$ when static use of longest path is employed.

Example 3

We now turn attention to another special topology with flow 13-3 shown in Figure 5.17, where the self-interference is significant on both paths and the number of hops are high along those paths. Figure 5.20 shows the delay performance of the set of routing protocols as α is varied for flow 13-3. In contrast to the first 2 examples, the mean delay is minimized for $\alpha = 0.5$ when congestion diversity is utilized. Note that the CDP is unable to follow an optimal $\alpha = 0.5$ (it tracks $\alpha = 0.38$). This we believe is due to the lag between the ideal time to switch between paths and the actual time it takes for

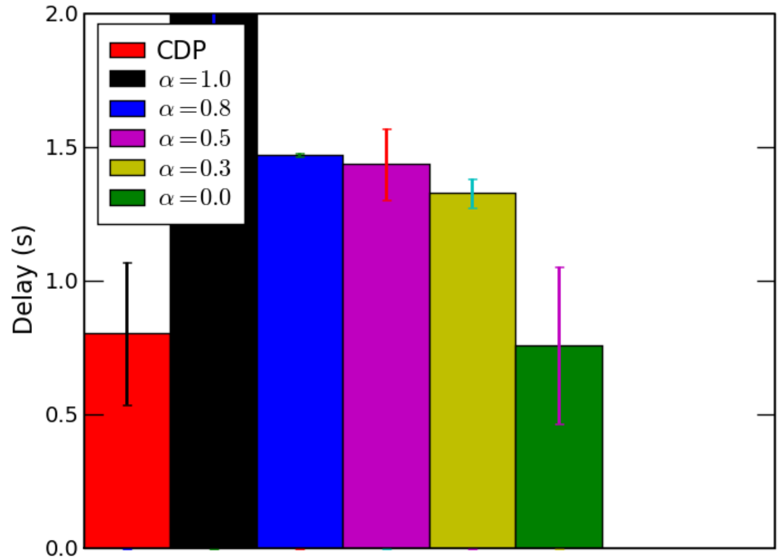


Figure 5.19: Delay performance as α is varied for case 1

CDP to switch paths.

The above examples reveal that exploiting multi-path diversity is a complicated function of the intra-flow and inter-path interference in the network. Even in very simple topologies, the interplay of congestion diversity gain and interference is too complicated to account for. In other words, a modular approach is not supported to address both effects simultaneously and thus necessitating a cross layer re-design of MAC and routing layer.

5.6.2 Performance in high interference scenarios

Next, we study the effectiveness of multipath routing in networks with non-negligible interference. In real deployments, concurrent flows already exist in the network. Unlike the test configurations in Section 5.5, deployed networks are likely to see a sufficiently high "interference floor". In presence of such steady interference floor, the contribution of the intra-flow and inter-path interference is likely to be minimal. In other words, we argue that in most practical scenarios, even with a modular design of MAC

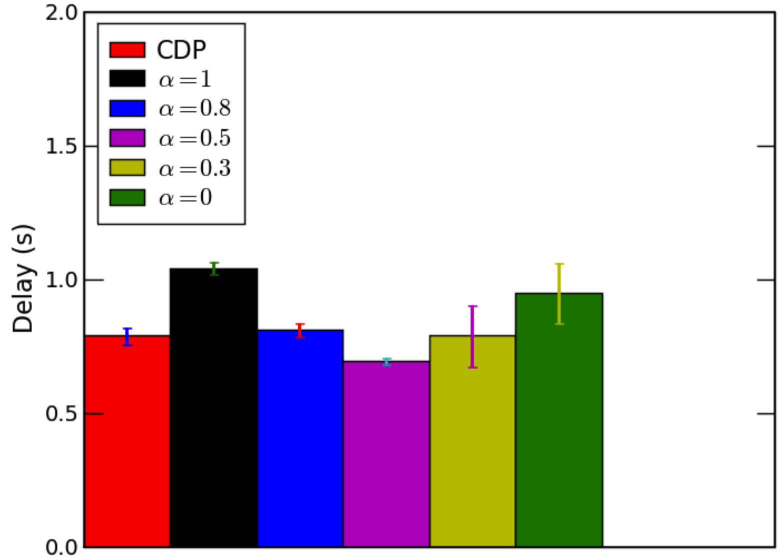


Figure 5.20: Delay performance as α is varied for case 2

and routing, the benefits of congestion-aware multi-path diversity are significant.

We test this conjecture by repeating the experiments of Section 5.5, where in addition to the two randomly selected heavy and long UDP flows, nodes in the network are engaged in the transmission of low intensity traffic at the rate of 10 packets/sec. Note that such background traffic can often be attributed to the control plane packets.

Figures 5.21 and 5.22 show the CDF of delay and normalized throughput. In fact comparing these to Figures 5.8 and 5.9, we can confirm the advantages of CDP over other candidate protocols.

5.7 Summary

This chapter presents study of a set of congestion-aware routing protocols Backpressure (BP), Enhanced Backpressure (E-BP) and Congestion Diversity Protocol (CDP) for routing packets across a wireless multi-hop network. We modify the protocol stack at the routing layer to take the congestion in the network into account. In E-BP and CDP, nodes route packets according to a rank ordering of the nodes based on a conges-

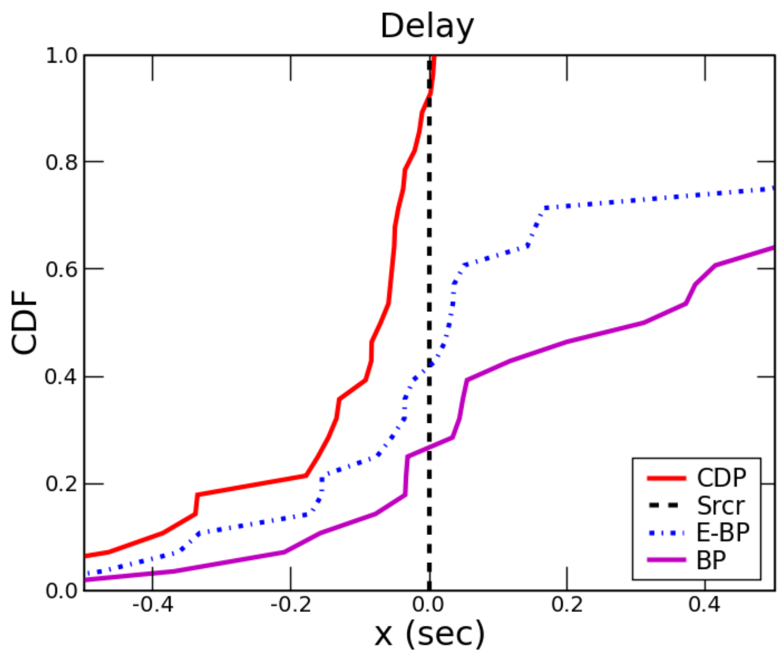


Figure 5.21: CDF of delay differential for high load with external interference

tion measure which combines the important aspects of shortest path routing with those of backpressure routing and are designed to alleviate the delay performance of BP.

We evaluated these routing protocols on a real testbed of 12 nodes with end-end delay and throughput ratio as central metrics for comparison. We compared the performance of BP, E-BP and CDP versus other routing-layer solutions, i.e. SRCR, under both UDP and TCP connections. We showed significant improvements for most arbitrary network set up and traffic conditions for CDP while BP and E-BP showed poorer performance. We also dissected network scenarios to gain insights and understand the reasons for improvements and performance degradations with respect to SRCR. In the process, we shed light on the importance of a cross-layer approach in which scheduling and transport-layer congestion control enable further improvement and complimentary roles in addition to the congestion-aware functionality. This set of observations allow us to suggest and consider a rich set of future directions for our work.

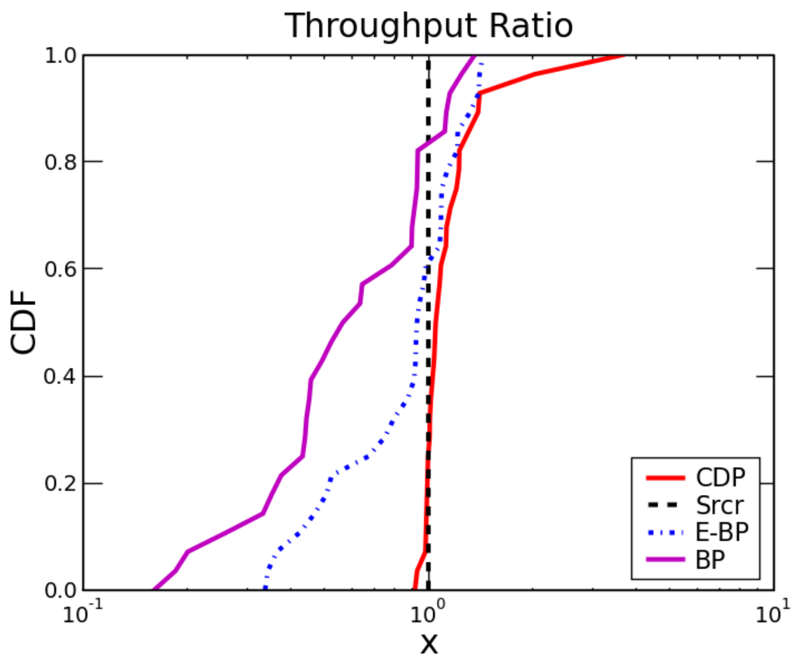


Figure 5.22: CDF of normalized throughput for high load with external interference

Acknowledgement

This chapter, in part, appears in the following publications. The dissertation author was the primary investigator and author of these papers.

- A. Bhorkar, T.Javidi, A. Snoeren, Achieving congestion diversity in wireless Ad-hoc networks,, under review IEEE Transaction on Networking.
- A. Bhorkar, T. Javidi, A. Snoeren, Achieving congestion diversity in wireless Ad-hoc networks, Infocom 11
- A. Bhorkar, T. Javidi, A. Snoeren, On practical implementation for congestion diversity in wireless Ad-hoc networks, Allerton 10
- A. Bhorkar, T. Javidi, A. Snoeren, Empirical Measurement of Draining Time, Expected Transmission (ETX), and Packet Erasure Probability in WiFi-based Mesh Networks, WinMEE 11

I would like to thank my co-authors Prof. Tara Javidi and Prof. Alex Snoeren.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

In this dissertation, we studied various routing algorithms for wireless mesh networks. The main objectives were classified into two categories i) determine a routing algorithm when the topology information is unavailable ii) finding routes to avoid congested paths.

For the first objective, we studies two routing algorithms AdaptOR and NRR in Chapter 2 and 3 respectively. The long term average reward criterion investigated in Chapter 2 inherently ignores the short-term performance. To capture the performance, we designed NRR in Chapter 3 via measuring the incurred "regret" over a finite horizon with an optimal rate of regret.

In Chapter 4, we provided a distributed opportunistic routing policy with congestion diversity (D-ORCD) combining the important aspects of shortest path routing with those of backpressure routing. Furthermore, we proposed a practical distributed and asynchronous 802.11 compatible implementation for D-ORCD, whose performance was investigated via a detailed set of QualNet simulations under practical and realistic networks.

In Chapter 5, we studied the implementation of various congestion aware routing algorithms on a testbed including Congestion Diversity Protocol (CDP), Backpressure (BP) and Enhanced-Backpressure (EBP). We studied the performance of these algorithms with respect to UDP and TCP traffic against state of the art shortest path routing algorithm SRCR. We showed that CDP performed significantly better in real network scenarios while backpressure based algorithms performed poorly. We also studied interesting interaction between the increased interference and exploiting the congestion diversity in the network.

6.2 Future Work

A number of assumptions are imposed on the optimality of AdaptOR and NRR. AdaptOR and NRR are designed for a static network. An important area for future research is to extend these algorithms are mobiles networks. Furthermore, we would like to relax the centralized controller assumption in proving the optimality of NRR.

In Chapter 4, we do not model the interference at the nodes in the network, but instead leave that issue to a classical MAC operation. However, the generalization to the networks with inter-channel interference follow directly from [13]. The price of this generalization is shown to be the centralization of the routing/scheduling globally across the network or a constant factor performance loss of the distributed variants [13, 16, 17]. In the future, we are interested in generalising D-ORCD for joint routing and scheduling optimizations as well as considering system level implications. Incorporating throughput optimal CSMA-based MAC scheduler (proposed in [53]) with congestion aware routing is also a promising area of research.

Our work, however, does not consider this closely related issue. Incorporating congestion control in opportunistic routing algorithms to minimize expected delay without the topology and the channel statistics knowledge is an area for future research.

An important subsequent endeavour is the implementation and performance analysis of AdaptOR, NRR and D-ORCD in our lab. The implementation of the three way handshake procedure at the driver to enable opportunistic routing is a significant challenge.

In Chapter 5, we have taken a modular approach in which the MAC layer and the transport layer are kept intact but the routing layer is modified. Our results indicate the need for the development of practical yet joint MAC, routing, and transport layer protocols that tackle the issues of contentions, congestion, and delay simultaneously as advocated by [53], [56]. This includes the design of a congestion aware MAC and TCP rate control algorithm based on congestion measure V_t. Given the legacy of the 802.11 MAC protocol, it might be useful to further study the modular approach when self-interference dominates. In particular, as seen in Section 5.6, sticking with a single route is beneficial in a self-interfering environment to avoid increased self-interference. Instead of changing the next hop as soon as any congestion is detected, "hold route CDP" which is a slightly modified version of CDP, might be useful to hold the route for a long enough duration until we detect sufficient congestion. Most wireless nodes are equipped with rate selection mechanisms that attempt to optimize the transmission rate. However, an appropriate rate selection algorithm remains as an important area of future work. As a side note, we would like to include a provably loop free mechanisms in CDP. Provably loop free methods using destination sequence numbers [69] are slow to propagate and are unsuitable in a very dynamic system. Future work includes extending a provably loop free technique [70] for CDP.

Appendix A

A.1 Appendix for Chapter 2

We start this section with a note on the notation. On the probability space $(\Omega, \mathcal{F}, P)$, we use notation $I : \Omega \to \{0, 1\}$ to denote the indicator random variable (with respect to $\mathcal{F}$), such that for all $\omega \in \Omega$, $A \in \mathcal{F}$, $I(A) = 1$ for all $\omega \in A$, and $I(A) = 0$ for all $\omega \notin A$. For a vector $x \in \mathbb{R}^D$, $D \geq 1$, we use $x(l)$ to denote the l^{th} element of the vector. Let $||x||_v$ denote the weighted max-norm with positive weight vector v, i.e. $||x||_v = \max_l \frac{|x(l)|}{v(l)}$. We denote the vector in $\mathbb{R}^D$ with all its components equal to 1 by $\mathbf{1}$. We also use the notation X^n to represent the first n random elements of the random sequence $\{X_k\}_{k=1}^{\infty}$.

A.1.1 Proof of Lemma 1

Lemma 1. Let

(J1) $\Lambda_0(\cdot, \cdot, \cdot) = 0$, $\Lambda^T_{max} = -R$, $\Lambda^i_{max} = 0$ for all $i \in \Theta$,

(J2) $\sum_{n=0}^{\infty} \alpha_n = \infty$, $\sum_{n=0}^{\infty} \alpha_n^2 < \infty$.

Then the sequence Λ_n obtained by the stochastic recursion (2.2)

$$\begin{aligned}\Lambda_{n+1}(i, S, a) = \Lambda_n(i, S, a) + \alpha_{\nu_n(i,S,a)}\\ \bigg(- \Lambda_n(i, S, a) + g(S, a) + \Lambda^a_{max} \bigg),\end{aligned}$$

converges to Λ^* almost surely.

To prove Lemma 1, we note that the adaptive computation given by (2.2) utilizes a stochastic approximation algorithm to solve the MDP associated with Problem **(AP)**. To study the convergence properties of this stochastic approximation, we appeal to known results at the intersection of learning and stochastic approximation given below.

In particular, consider a set of stochastic sequences on $\mathbb{R}^D$, as $\{x_n, \bar{\alpha}_n, \mathcal{M}_{n+1}\}$, and the corresponding filtration $\mathcal{G}_n$, i.e. the increasing σ-field, satisfying the following recursive equation

$$x_{n+1} = x_n + \bar{\alpha}_n[U(x'_n) - x_n + \mathcal{M}_{n+1}],$$

where U is a mapping from $\mathbb{R}^D$ into $\mathbb{R}^D$ and $x'_n = \left(x_{n_1}(1), x_{n_2}(2), ..., x_{n_D}(D)\right), 0 \leq n_j \leq n$, $j \in \{1, 2, ..., D\}$, is a vector of possibly delayed components of x_n. If no information is outdated, then $n_j = n$ for all j and $x'_n = x_n$. The following important result on the convergence of x_n is provided in [22].

Fact 4. [22, Theorem 2] Assume $\{x_n, \bar{\alpha}_n, \mathcal{M}_{n+1}\}$ and U satisfy the following conditions:

(G1) For all $n \geq 0$ and $1 \leq l \leq D$, $0 \leq \bar{\alpha}_n(l) \leq 1$ a.s.;
for $1 \leq l \leq D$, $\sum_{n=0}^{\infty} \bar{\alpha}_n(l) = \infty$ a.s.;
for $1 \leq l \leq D$, $\sum_{n=0}^{\infty} \bar{\alpha}_n^2(l) < \infty$ a.s.

(G2) $\mathcal{M}_n$ is a martingale difference with finite second moment, i.e. $\mathbf{E}\{\mathcal{M}_{n+1}|\mathcal{G}_n\} = 0$, and there exist constants A and B such that

$$\mathbf{E}\{\mathcal{M}_{n+1}^2|\mathcal{G}_n\} \leq A + B(\max_{n' \leq n} ||x_{n'}||)^2.$$

(G3) There exists a positive vector v, scalars $\beta \in [0, 1)$ and $C \in \mathbb{R}^+$, such that

$$||U(x)||_v \leq \beta \, ||x||_v + C.$$

(G4) Mapping $U : \mathbb{R}^D \to \mathbb{R}^D$ satisfies the following properties:

1. U is component-wise monotonically increasing;
2. U is continuous;

3. U has a unique fixed point $x^* \in \mathbb{R}^D$;

4. $U(x) - r\mathbf{1} \leq U(x - r\mathbf{1}) \leq U(x + r\mathbf{1}) \leq U(x) + r\mathbf{1}$, for any $r \in \mathbb{R}^+$.

(G5) For any j, $n_j \to \infty$ as $n \to \infty$.

Then the sequence of random vectors x_n converges to the fixed point x^* almost surely.

Let $\mathcal{G}_n$ be the increasing σ-field generated by random vectors $(\Lambda_n, S_n^i, a_n^i, \nu_n)$. Let $x_n = \Lambda_n$ be the random vector of dimension $D = \sum_{i\in\Theta}\sum_{S\in\mathfrak{S}^i} |A(S)|$, generated via recursive equation (2.2). Furthermore,

$$(U\Lambda_n)(i, S, a) = g(S, a) + \sum_{S'\in\mathfrak{S}^a} P(S'|a) \max_j \Lambda_n(a, S', j),$$

$$\bar{\alpha}_n(i, S, a) = \alpha_{\nu_n(i,S,a)} I(S_n^i = S, a_n^i = a).$$

Let $\{\mathcal{M}_{n+1}\}$ be a random vector whose $(i, S, a)^{th}$ element is constructed as follows:

$$\begin{aligned}\mathcal{M}_{n+1}(i, S, a) &= \max_j \Lambda_{n_a}(a, S_{n_a}^a, j) \\ &\quad - \sum_{S'\in\mathfrak{S}^a} P(S'|a) \max_j \Lambda_{n_a}(a, S', j),\end{aligned}$$

where $0 \leq n_a \leq n$, and $S_{n_a}^a$ is the most recent state visited by node a.

Now we can rewrite (2.2) and (2.3) as in the form investigated in Fact 4, i.e.

$$\begin{aligned}\Lambda_{n+1}(i, S, a) &= \Lambda_n(i, S, a) + \bar{\alpha}_n(i, S, a)\Big((U\Lambda_{n_a})(i, S, a) \\ &\quad -\Lambda_n(i, S, a) + \mathcal{M}_{n+1}(i, S, a)\Big).\end{aligned}$$

The remaining steps of the proof reduce to verifying statements (G1)-(G5). This is verified in Lemma 5 below.

Lemma 5. $(\Lambda_n, \bar{\alpha}_n, \mathcal{M}_{n+1})$ *satisfy conditions (G1)-(G5).*

Proof. • (G1): It is shown in Lemma 7 that algorithm d-AdaptOR guarantees that every state-action is attempted infinitely often (i.o.). Hence,

$$\sum_{n=0}^{\infty} \bar{\alpha}_n(i,S,a) = \sum_{n=0}^{\infty} \alpha_{\nu_n(i,S,a)} I(S_n^i = S, a_n^i = a)$$
$$\geq I((i,S,a) \text{ visited i.o. })(\sum_{n=0}^{\infty} \alpha_n) = \infty.$$

However,

$$\begin{aligned}\sum_{n=0}^{\infty} \bar{\alpha}_n^2(i,S,a) &\leq \sum_{i,S,a}\sum_{n=0}^{\infty} \alpha_{\nu_n(i,S,a)}^2 I(S_n^i = S, a_n^i = a) \\ &\leq \sum_{i\in\Theta} |\mathfrak{S}^i||d+1| \sum_{n=0}^{\infty} \alpha_n^2 < \infty.\end{aligned}$$

• (G2):

$$\begin{aligned}\mathbf{E}[\mathcal{M}_{n+1}|\mathcal{G}_n, n_a] &= \mathbf{E}_{S^a}[\max_j \Lambda_{n_a}(a, S^a, j)] \\ &\quad - \sum_{S'} P(S'|a) \max_j \Lambda_{n_a}(a, S', j) \\ &= 0.\end{aligned}$$

$$\mathbf{E}[\mathcal{M}_{n+1}|\mathcal{G}_n] = \mathbf{E}_{n_a}\left[\mathbf{E}[\mathcal{M}_{n+1}|\mathcal{G}_n, n_a]\right] = 0.$$

$$\begin{aligned}\mathbf{E}[\mathcal{M}_{n+1}^2|\mathcal{G}_n, n_a] &\leq \mathbf{E}_{S^a}[(\max_j \Lambda_{n_a}(a, S^a, j))^2] \\ &\leq \max_{S^a}\max_j \left(\Lambda_{n_a}(a, S^a, j)\right)^2 \\ &\leq ||\Lambda_{n_a}||^2 .\end{aligned}$$

$$\begin{aligned}\mathbf{E}[\mathcal{M}_{n+1}^2|\mathcal{G}_n] &= \mathbf{E}_{n_a}\left[\mathbf{E}[\mathcal{M}_{n+1}^2|\mathcal{G}_n, n_a]\right] \\ &\leq \mathbf{E}_{n_a}[||\Lambda_{n_a}||^2] \\ &\leq \max_{n'\leq n} ||\Lambda_{n'}||^2 .\end{aligned}$$

Thus Assumption (G2) of Fact 4 is satisfied.

- (G3): Let $Z_d = \{S : d \in S, S \in \{\mathfrak{S}^i\}_{i\in\Theta}\}$ denote the set of states which contain the destination node d. Moreover, let $Z_d^i = \{S : d \in S, i \in \Theta, S \in \mathfrak{S}^i\}$. Let $\tau_{Z_d}^{\pi}$ be the hitting time associated with set Z_d and policy $\pi \in \Pi$, i.e. $\tau_{Z_d}^{\pi} = \min\{n > 0 : \exists S \in Z_d, S \in \{S_n^i\}_{i\in\Theta}\}$. Policy π is said to be proper if $Prob(\tau_{Z_d}^{\pi} < \infty|\mathcal{F}_0) = 1$. Let us now fix a proper deterministic stationary policy $\pi \in \Pi$. Existence of such a policy is guaranteed from the connectivity between 0 and d. Let F be the termination state which is reached after taking the termination action T. Let us define a policy dependent operator $\mathcal{L}^{\pi}$,

$$(\mathcal{L}^{\pi}\Lambda)(i, S, a) = g(S, a) + \sum_{S' \notin Z_d^a \cup F} P(S'|a)\Lambda(a, S', \pi(S')). \tag{A.1}$$

  We then consider a Markov chain with states (i, S, a) and with the following dynamics: from any state (i, S, a), we move to state $(a, S', \pi(S'))$, with probability $P(S'|a)$. Thus, subsequent to the first transition, we are always at a state of the form $(i, S, \pi(S))$ and the first two components of the state evolve according to policy π. As π is assumed proper, it follows that the system with states (i, S, a) also evolves according to a proper policy. We construct a matrix Q with each entry corresponding to the transition from state (i, S) to $(\pi(S), S')$ with value equal to $P(S'|\pi(S))$ for all $S \notin Z_d^i \cup F, S' \notin Z_d^{\pi(S)} \cup F$ for all i.

  Since policy π is proper, the maximum eigenvalue of matrix Q is strictly less than 1. As Q is a non-negative matrix, the Perron Frobenius theorem guarantees the existence of a positive vector w with components $w_{(i,S,a)}$ and some $\beta \in [0, 1)$ such that

$$\sum_{S' \notin Z_d^a \cup F} P(S'|a)w_{(\pi(S),S',\pi(S'))} \leq \beta w_{(i,S,a)}. \tag{A.2}$$

  From (A.2), we have a positive vector v such that $||(\mathcal{L}^{\pi}\Lambda) - \Lambda^{\pi}||_v \leq \beta\, ||\Lambda - \Lambda^{\pi}||_v$, where Λ^{π} is the fixed point of equation $\Lambda = \mathcal{L}^{\pi}\Lambda$.

  From the definition of U (2.4) and $\mathcal{L}^{\pi}$ (A.1) we have $|(U\Lambda)(\cdot, \cdot, \cdot)| \leq |(\mathcal{L}^{\pi}\Lambda)(\cdot, \cdot, \cdot)|$.

Using this and the triangle inequality, we obtain

$$\begin{aligned} ||U\Lambda||_v &\leq ||\mathcal{L}^\pi \Lambda||_v \\ &\leq ||\mathcal{L}^\pi \Lambda - \mathcal{L}^\pi \Lambda^\pi||_v + ||\mathcal{L}^\pi \Lambda^\pi||_v \\ &\leq \beta\, ||\Lambda - \Lambda^\pi||_v + ||\Lambda^\pi||_v \\ &\leq \beta\, ||\Lambda||_v + 2\, ||\Lambda^\pi||_v\,, \end{aligned}$$

establishing the validity of (G3).

- (G4): Assumption (G4) is satisfied by operator U using following fact:

Fact 5. [32, Proposition 4.3.1] U is monotonically increasing, continuous, and satisfies $U(\Lambda) - r\mathbf{1} \leq U(\Lambda - r\mathbf{1}) \leq U(\Lambda + r\mathbf{1}) \leq U(\Lambda) + r\mathbf{1}$, $r > 0$.

Λ^* is a fixed point of U. From (2.5) and (2.6) we obtain

$$\max_{j \in A(S)} V^*(j) \;=\; \max_{j \in A(S)} \Lambda^*(i, S, j) + R. \tag{A.3}$$

Furthermore, using (2.5) and (A.3), for all $i \in \Theta$

$$\Lambda^*(i, S, a) = g(S, a) + \sum_{S'} P(S'|a) \max_{j \in A(S')} V^*(j) - R. \tag{A.4}$$

The existence of fixed point Λ^* follows from (A.4), while the uniqueness of Λ^* follows from uniqueness of V^* (Fact 1).

- (G5): Suppose $n_a \nrightarrow \infty$ as $n \to \infty$. Therefore, there exists N such that $n_a < N$ for all n. This means that the number of times that node a has transmitted a packet is bounded by N. But this contradicts Lemma 7 which says that each state-action pair (S, a) is visited i.o. Therefore $n_a \to \infty$ as $n \to \infty$ for all a, and condition (G5) holds.

Thus Assumptions (G1)-(G5) are satisfied. Hence, from Fact 4 our iterate (2.2) converges almost surely to Λ^*, the unique fixed point of U. □

Lemma 6. *If policy ϕ^* is followed, then action $a \in A(S)$ is selected i.o. if state $S \in \mathfrak{S}$ is visited i.o.*

Proof. Define the random variable $K_n = I(S_n^i = S \text{ for any } i \in \Theta|\phi^*)$. Let $\mathcal{K}_n$ be the σ-field generated by $(K_1, K_2, \ldots, K_n)$. Let $A_n = \{\omega : a_n^i = a, S_n^i = S \text{ for any } i \in \Theta|\phi^*\}$.

From the construction of the algorithm it is clear that A_n is $\mathcal{K}_n$ measurable. Now it is clear that under policy ϕ^*, A_{n+1} is independent of K^{n-1} given K_n and $N_n(i, S), i \in \Theta$. Define,

$$\begin{aligned} & P(A_{n+1}|K_n, N_n(i, S) \text{ for all } i \in \Theta) \\ & \geq \begin{cases} 0 & \text{if } K_n = 0 \\ \frac{\min_{i\in\Theta} \epsilon_n(i,S)}{|A(S)|} & \text{if } K_n = 1 \end{cases} \end{aligned} \tag{A.5}$$

$$\begin{aligned} & \sum_{n=0}^{\infty} Prob(A_{n+1}|\mathcal{K}_n) \\ & \geq \sum_{n=0}^{\infty} Prob(A_{n+1}|K_n, N_n(i, S) \text{ for all } i \in \Theta) \\ & \geq I(\text{ S is visited i.o.}) \sum_{n=0}^{\infty} \min_{i\in\Theta} \frac{\epsilon_n(i, S)}{|A(S)|} \\ & \geq \frac{I(\text{ S is visited i.o.})}{|A(S)|} \sum_{n=0}^{\infty} \frac{1}{\sum_{i\in\Theta} N_n(i, S) + 1} \\ & \geq \frac{I(\text{ S visited i.o. })}{|A(S)|} \sum_{n=0}^{\infty} \frac{1}{n(d + 1) + 1} = \infty. \end{aligned} \tag{A.6}$$

The next step of the proof is based on the following fact:

Fact 6. [71, Corollary 5.29, (Extended Borel-Cantelli Lemma)] Let $\mathcal{K}_k$ be an increasing sequence of σ-fields and let A_k be $\mathcal{K}_k$-measurable. If $\sum_{k=1}^{\infty} Prob(A_k|\mathcal{K}_{k-1}) = \infty$ then $P(A_k \; i.o.) = 1$.

Thus, from Fact 6, $a \in A(S)$ is visited i.o, if S is visited i.o. □

Lemma 7. *If policy ϕ^* is followed, then each state-action (S, a) is visited infinitely often.*

Proof. We say states $S, S' \in \mathfrak{S}$ communicate if there exists a sequence of actions $\{a_1, \ldots a_k, k < \infty\}$ such that probability of reaching state S' from state S following the sequence of actions $\{a_1, \ldots a_k\}$ is greater than zero. Using Lemma 6, if state $S \in \mathfrak{S}$

is visited i.o., then every action $a \in A(S)$ is chosen i.o. as the set $A(S)$ is finite. Hence, states S' such that $P(S'|a) > 0, S' \in \mathfrak{S}$, are visited i.o. if S is visited i.o. By Lemma 6 every action $a' \in A(S')$ is also visited i.o. Following a similar argument and with repeated application of Lemma 6, every state $S'' \in \mathfrak{S}$ which communicates with state S and actions $a \in A(S'')$ are visited i.o.

Under the assumption of the packet generation process in Section 2.2, a packet is generated i.o. at the source node 0. Thus state $\{0\}$ is reached i.o. The construction of set $\mathfrak{S}$ is such that every state $S \in \mathfrak{S}$ communicates with state $\{0\}$. Thus each (S, a) is visited i.o since $|\mathfrak{S}|$ is finite. $\square$

A.1.2 Proof of Lemma 2

Lemma 2. For any (P)-admissible policy $\phi \in \Phi$ for Problem **(P)** and for all $N = 1, 2, \ldots$,

$$\mathbf{E}^{\phi}\left[\frac{1}{M_N}\sum_{m=1}^{M_N}\left\{r_m - \sum_{n=\tau_s^m}^{\tau_T^m - 1} c_{i_{n,m}}\right\}\right] \leq V^*(0).$$

Proof. To prove the lemma, we refer to the Auxiliary Problem **(AP)**. In this problem we have assumed the existence of a centralized controller with full knowledge of the local broadcast model. Mathematically speaking, let $\mathbb{P}$ be the sample space of the random probability measures for the local broadcast model. Specifically, $\mathbb{P} := \{p \in \mathbb{R}^{2^d} \times \mathbb{R}^d :$ p is a non-square left stochastic matrix$\}$. Moreover, let $\mathcal{P}_P$ be the trivial σ-field generated by the local broadcast model $P \in \mathbb{P}$ (sample point in $\mathbb{P}$), i.e. $\mathcal{P}_P = \{P, \mathbb{P}\backslash P, \emptyset, \mathbb{P}\}$.[1] Recall that S_n^i denotes the set of nodes that have received the packet due to a transmission from node i at time n, while a_n^i denotes the corresponding routing decision node i takes at time n.[2] For Auxiliary Problem **(AP)**, a routing policy is a collection $\pi = \{\pi^i\}_{i\in\Theta}$ of routing decisions taken for nodes $i \in \Theta$ at the centralized controller, where π^i denotes a sequence of random actions $\pi^i = \{a_0^i, a_1^i, \ldots\}$ for node i. The routing policy π is said to be (AP)-admissible for Auxiliary Problem **(AP)** if the event $\{a_n^i = a\}$ belongs to the product σ-field $\mathcal{P}_P \times \prod_i \mathcal{H}_n^i$ [72].

[1] The σ-field captures the knowledge of the realization of local broadcast model and assumes a well-defined prior on these models.

[2] $S_n^i = \emptyset, a_n^i = T$ if node i does not transmit at time n.

From Fact 1, since π^* is the optimal policy for one packet, for each packet m and for any feasible policy $\phi \in \Phi$,

$$\begin{aligned} V^*(0) &= \mathbf{E}^{\pi^*}\left[r_m - \sum_{n=\tau_s^m}^{\tau_T^m - 1} c_{i_{n,m}} | \mathcal{F}_0 \right] \\ &\geq \mathbf{E}^{\phi}\left[r_m - \sum_{n=\tau_s^m}^{\tau_T^m - 1} c_{i_{n,m}} \right], \end{aligned}$$

where the inequality follows from the fact that $\Phi \subseteq \Phi$. The remaining steps are straightforward.

$$\mathbf{E}^{\phi}\left[\frac{1}{M_N} \sum_{m=1}^{M_N} \left\{ r_m - \sum_{n=\tau_s^m}^{\tau_T^m - 1} c_{i_{n,m}} \right\} \right] \leq \mathbf{E}^{\phi}\left(\frac{1}{M_N} \sum_{m=1}^{M_N} V^*(0) \right)$$

$$= V^*(0).$$

$\square$

A.1.3 Proof of Lemma 3

Lemma 3. For any $\delta > 0$,

$$\liminf_{N \to \infty} \mathbf{E}^{\phi^*}\left[\frac{1}{M_N} \sum_{m=1}^{M_N} \left\{ r_m - \sum_{n=\tau_s^m}^{\tau_T^m - 1} c_{i_{n,m}} \right\} \right] \geq V^*(0) - \delta.$$

Proof. From (2.5), (2.6), and (A.3) we obtain the following equality for all $i \in \Theta,\ S \in \mathfrak{S}^i$,

$$\arg\max_{j \in A(S)} V^*(j) = \arg\max_{j \in A(S)} \Lambda^*(i, S, j)\ . \tag{A.7}$$

Let

$$b = \min_{i \in \Theta} \min_{S \in \mathfrak{S}^i} \min_{\substack{j,k \in A(S) \\ \Lambda^*(i,S,j) \neq \Lambda^*(i,S,k)}} |\Lambda^*(i, S, j) - \Lambda^*(i, S, k)|.$$

Lemma 1 implies that, in an almost sure sense, there exists packet index $m_1 < \infty$ such that for all $n > \tau_s^{m_1}$, $i \in \Theta$, $S \in \mathfrak{S}^i$, $a \in A(S)$,

$$|\Lambda_n(i,S,a) - \Lambda^*(i,S,a)| \leq b/2.$$

In other words, from time $\tau_s^{m_1}$ onwards, given any node $i \in \Theta$ and set $S \in \mathfrak{S}^i$, the probability that d-AdaptOR chooses an action $a \in A(S)$ such that $\Lambda^*(i,S,a) \neq \max_{j \in A(S)} \Lambda^*(i,S,j)$ is upper bounded by $\epsilon_n(i,S)$. Furthermore, since $N_n(i,S) \to \infty$ (Lemma 7), for a given $\gamma > 0$, with probability 1, there exists a packet index $m_2 < \infty$ such that for all $n > \tau_s^{m_2}$, $\max_{i,S} \epsilon_n(i,S) < \gamma$.

Let $m_0 = \max\{m_1, m_2\}$. For all packets with index $m \leq m_0$, the overall expected reward is upper-bounded by $m_0 R < \infty$ and lower-bounded by $-\frac{m_0}{\lambda} d \max_i c_i > -\infty$, hence, their presence does not impact the expected average per packet reward. Consequently, we need to only consider the routing decisions of policy ϕ^* for packets $m > m_0$.

Consider the m^{th} packet generated at the source. Let B_k^m be an event for which there exist k instances when d-AdaptOR routes packet m differently from the possible set of optimal actions. Mathematically speaking, event B_k^m occurs iff there exists instances $\tau_s^m \leq n_1^m \leq n_2^m \ldots n_k^m \leq \tau_T^m$ such that for all $l = 1, 2, \ldots, k$

$$\Lambda^*(i_{n_l^m,m}, S_{n_l^m}, a_{n_l^m}) \neq \max_{j \in A(S_{n_l^m})} \Lambda^*(i_{n_l^m,m}, S_{n_l^m}, j),$$

where $S_{n_l^m}$ is the set of nodes that have successfully received packet m at time n_l^m due to a transmission from node $i_{n_l^m,m}$. We call event B_k^m a mis-routing of order k. For $m > m_0$,

$$Prob(B_k^m) \leq (\max_{i,S} \epsilon_n(i,S))^k \leq \gamma^k.$$

Now for packets $m > m_0$, let us consider the expected differential reward under policies π^* and ϕ^*:

$$\mathbf{E}^{\pi^*}\left[\left\{r_m - \sum_{n=\tau_s^m}^{\tau_T^m - 1} c_{i_{n,m}} | \mathcal{F}_0\right\}\right] - \mathbf{E}^{\phi^*}\left[\left\{r_m - \sum_{n=\tau_s^m}^{\tau_T^m - 1} c_{i_{n,m}}\right\}\right]$$

$$
\begin{aligned}
&= V^*(0) - \mathbf{E}^{\phi^*}\left[\left\{r_m - \sum_{n=\tau_s^m}^{\tau_T^m-1} c_{i_{n,m}}\right\}\right] \\
&= \sum_{k=0}^{\infty} \mathbf{E}^{\phi^*}\left[V^*(0) - \left\{r_m - \sum_{n=\tau_s^m}^{\tau_T^m-1} c_{i_{n,m}}\right\}|B_k^m\right] \\
&\quad \times Prob(B_k^m) \\
&\leq \sum_{k=0}^{\infty} k\ R\ Prob(B_k^m) && \text{(A.8)} \\
&\leq R\sum_{k=1}^{\infty} k\gamma^k && \text{(A.9)} \\
&= \delta, && \text{(A.10)}
\end{aligned}
$$

where $\delta = \frac{\gamma R}{(1-\gamma)^2}$. Inequality (A.8) is obtained by noticing that the maximum loss of the reward occurs if algorithm d-AdaptOR decides to drop packet m (no reward) while there exists a node j in the set of potential forwarders such that $V^*(j) \approx R$.

Thus, for all $\delta > 0$, the expected average per packet reward under policy ϕ^* is bounded as

$$
\begin{aligned}
&\liminf_{N\to\infty} \mathbf{E}^{\phi^*}\left[\frac{1}{M_N}\sum_{m=1}^{M_N}\left\{r_m - \sum_{n=\tau_s^m}^{\tau_T^m-1} c_{i_{n,m}}\right\}\right] \\
&\quad \geq \liminf_{N\to\infty} \mathbf{E}^{\phi^*}\left[\frac{1}{M_N}\sum_{m=1}^{M_N}(V^*(0)-\delta)\right] \\
&\quad = V^*(0) - \delta.
\end{aligned}
$$

□

A.2 Appendix for Chapter 3

A.2.1 Proof of Lemma 4

$$
\text{Let } \lambda^* = \sup_{\pi} \liminf_{N\to\infty} \frac{1}{N}\sum_{n=0}^{N} \mathbf{E}^{\pi}\left[g(S_n, a_n)\right]. \tag{A.11}
$$

Furthermore, let $\rho^*(S,a) = h^*(S) - h^*(a), a \in A(S)$.

Fact 7. (Chap 9, [31]) We can associate a bias function $h^*(S)$ and a constant average reward λ^* independent of state, with (A.11) as:[3]

$$h^*(S) = \max_{a \in A(S)} \left\{ g(S,a) + \sum_{S' \subseteq \mathcal{N}(a)} P(S'|a) h^*(S') - \lambda^* \right\}. \tag{A.12}$$

Using 7 and following the steps of Proposition 1 in [39], we can prove the result in Lemma 1.

Next we will bound the term $\mathfrak{N}_N(S,a)$. For each $\epsilon > 0$ let us define events A_n, as $A_n := \{||V_n - V^*||_\infty \leq 2\epsilon\}$.

$$\begin{aligned}
\mathfrak{N}^1_N(S,a,\epsilon) &= \sum_{n=0}^{N-1} \mathbf{1}[\{(S_n,a_n) = (S,a)\} \cap \{A_n\} \\
& \quad \cap \{\Lambda_n(a) \geq V^*(S) - \epsilon\}] \\
\mathfrak{N}^2_N(S,a,\epsilon) &= \sum_{n=0}^{N-1} \mathbf{1}[\{(S_n,a_n) = (S,a)\} \cap \{A_n\} \\
\cap & \quad \{\Lambda_n(a) < V^*(S) - \epsilon\}] \\
\mathfrak{N}^3_N(\epsilon) &= \sum_{n=0}^{N-1} \mathbf{1}[\bar{A}_n],
\end{aligned}$$

where $\bar{A}_n$ denotes the complement of A_n and $V^*(S)$ is an extended set valued function such that $V(S) = \max_{j \in A(S)} V(j)$. It follows that:

$$\mathfrak{N}_N(S,a) \leq \mathfrak{N}^1_N(S,a,\epsilon) + \mathfrak{N}^2_N(S,a,\epsilon) + \mathfrak{N}^3_N(\epsilon).$$

Lemma 8. $\mathbf{E}[\mathfrak{N}^1_N(S,a,\epsilon)] = O(\log N)$

[3]The Bellman equation holds for weakly communicating class of MDP. It is easy to see that the induced MDP for the routing problem is weakly communicating.

Proof:

$$
\begin{aligned}
& E[\mathfrak{N}_N(S,a,\epsilon)] \\
\leq & \sum_{n=0}^{N-1} \mathbf{1}[\{(S_n,a_n)=(S,a)\} \cap \{|V_n(a)-V^*(a)|<2\epsilon\} \\
& \cap\{\Lambda_n(a) > V^*(S)-\epsilon\}] \\
\overset{(a)}{\leq} & \sum_{n=0}^{N-1} \mathbf{1}[(S_n,a_n)=(S,a) \cap |V_n(a)-V^*(a)|<2\epsilon \\
& \cap V_n(a)-V^*(a)+\sqrt{\frac{2\log n}{\mathfrak{N}_n(a)}} > V^*(S)-V^*(a)-\epsilon] \\
\overset{(b)}{\leq} & \sum_{n=0}^{N-1} \mathbf{1}[(S_n,a_n)=(S,a) \cap (2\epsilon + \\
& \sqrt{\frac{2\log n}{\mathfrak{N}_n(a)}} > V^*(S)-V^*(a)-\epsilon)]
\end{aligned}
$$

$$
\begin{aligned}
\leq & \sum_{n=0}^{N-1} \mathbf{1}[(S_n,a_n)=(S,a) \cap \\
& \sqrt{\frac{2\log n}{\mathfrak{N}_n(a)}} > V^*(S)-V^*(a)+\epsilon] \\
\leq & \sum_{n=0}^{N-1} \mathbf{1}[(S_n,a_n)=(S,a) \cap \\
& \mathfrak{N}_n(a) \leq \frac{2\log n}{(V^*(S)-V^*(a)-3\epsilon)^2}] \\
\overset{(c)}{\leq} & \frac{2\log N}{(V^*(S)-V^*(a)-3\epsilon)^2}
\end{aligned}
$$

- (a): Follows from the definition of Λ_n in (3.7).
- (b): Follows from the condition $|V_n(a)-V^*(a)|<2\epsilon$.
- (c): Follows from Lemma 3 [39].

Lemma 9. $\mathbf{E}[\mathfrak{N}_N^2(S,a,\epsilon)] = o(\log N)$

Proof: Let

$$
E_n(S,a,\epsilon) := \{(S_n,a_n)=(S,a) \cap A_n \cap \Lambda_n(a) \leq V^*(S)-\epsilon\}
$$

Thus $\mathfrak{N}_N^2(S,a,\epsilon) = \sum_{n=1}^{N} \mathbf{1}[E_n(S,a,\epsilon)]$. Let $a^* = \arg\max_{j\in A(S)} V(j) = V^*(S)$. When event $E_n(S,a,\epsilon)$ occurs,

$$\Lambda_n(a^*) \le \Lambda_n(a) < V^*(S) - \epsilon. \tag{A.13}$$

(A.13) follows from the fact that a^* is not selected but a is selected. Thus,

$$\begin{aligned}
& \sqrt{\frac{2\log n}{\mathfrak{N}_n(a^*)}} + V_n^*(S) < \sqrt{\frac{2\log n}{\mathfrak{N}_n(a)}} + V_n(a) < V^*(S) - \epsilon \\
\Rightarrow \quad & \sqrt{\frac{2\log n}{\mathfrak{N}_n(a^*)}} + V_n^*(S) < V^*(S) - \epsilon, \forall S \\
\Rightarrow \quad & V^*(S) - V_n^*(S) > \sqrt{\frac{2\log n}{\mathfrak{N}_n(a^*)}} + \epsilon
\end{aligned} \tag{A.14}$$

The difference between the value function $\tilde{V}$ and V^* induced by the optimal policies of the local broadcast models $\tilde{P}(S'|a)$ and $P(S'|a)$ respectively is related as,

Fact 8 ([21])**.** There exists $K_1 > 0$, such that,

$$\max_S \left|\tilde{V}(S) - V^*(S)\right| \le K_1 \max_a \sigma(\tilde{P}(\cdot|a), P(\cdot|a)),$$

if $P(\cdot|a) = 0 \Rightarrow \tilde{P}(\cdot|a) = 0$, where $\sigma(P(\cdot|a), \tilde{P}(\cdot|a)) = \sup_{S\in\mathfrak{S}} \sum_{S'\in\mathfrak{S}} (P(S'|i) - \tilde{P}(S'|i))$.

Using Fact 8, Assumption 4, and Scheffe's lemma [72], $\exists\, K$, with

$$||V^* - V_n||_\infty \le K \max_i ||P(\cdot|a) - P_n(\cdot|a)||_1 .$$

From (A.14),

$$\begin{aligned}
E_n(S,a,\epsilon) &\subseteq \left\{ ||V^* - V_n^*||_\infty > \sqrt{\frac{2\log n}{\mathfrak{N}_n(a)}} + \epsilon \right\} \\
&\subseteq \bigcup_a \left\{ ||P(.|a) - P_n(.|a)||_1 > \frac{1}{K}\sqrt{\frac{2\log n}{\mathfrak{N}_n(a)}} + \epsilon \right\} \\
&\subseteq \bigcup_{m=0}^{n} \{\mathfrak{N}_n(a) = m \cap ||P(.|i) - P_n(.|i)||_1 > \\
&\qquad \frac{1}{K}\sqrt{\frac{2\log n}{\mathfrak{N}_n(a^*)}} + \epsilon \right\}
\end{aligned}$$

By the chernoff bound,

$$\begin{aligned} & Prob(||P_n(S|a) - P(S|a)||_1 > \delta | \mathfrak{N}_n(a) = m) \\ \leq \;& C \exp(-m\delta^2/2). \end{aligned}$$

$$\begin{aligned} & Prob(E_n(S, a, \epsilon)) \\ & \leq \sum_{m=0}^{n} C \exp(-m \left(\frac{1}{K} \sqrt{\frac{2 \log n}{m}} + \epsilon \right)^2) \\ & \leq \frac{C}{n} \sum_{m=1}^{\infty} \exp(-\epsilon^2 m/2 - \frac{\epsilon}{K^2}(\sqrt{2m \log n})) \\ & = o(\frac{1}{n}) \end{aligned}$$

$$\begin{aligned} \mathbf{E}(\mathfrak{N}_n^2) &= \sum_{n=0}^{N} \mathbf{E}[\mathbf{1}[E_n(S, a, \epsilon)]] \\ &= \sum_{n=0}^{N} Prob(E_n(S, a, \epsilon)) = \sum_{n=1}^{N} o(\frac{1}{n}) = o(\log N) \end{aligned}$$

Lemma 10. $\mathbf{E}[\mathfrak{N}_N^3(\epsilon)] = O(\log N)$

Proof: Note that $A_n \subseteq \{||P_n - P||_1 > 2\epsilon\}$. The lemma is proved using the Chernoff bound, and the fact that there are $\log(n), i \in \Theta$ probe packets in time n.

A.3 Appendix for Chapter 4

A.3.1 Throughput Optimal CDP

We now relate CDP to a closely related throughput optimal algorithm.

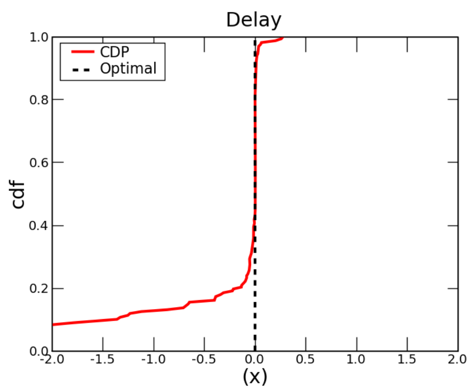
Definition 4. A routing policy is said to be throughput optimal if it stabilizes the network for all arrival rates that belong to the interior of the stability region.

Consider the following variation of CDP algorithm wherein instead of using (5.4) for choosing the next hop K_{CDP}, the next hop is chosen as

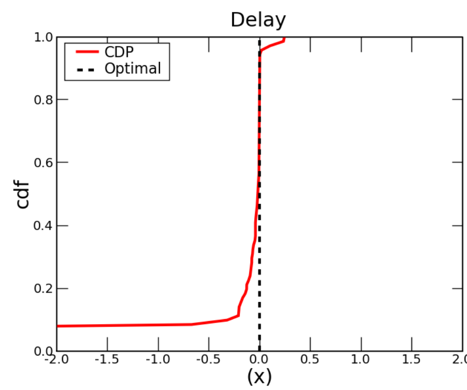
$$K_{OPT}^{(n,d)} = \arg\min_k \left\{ q(n)W(n,k) + V_t^d(k) \right\}. \quad \text{(A.15)}$$

Using the techniques provided in ([50]) it can be proved that such choice of choosing next hop is throughput optimal the assumptions that transmission times are instantaneously and perfectly available at any instant of time.

Thus, optimal algorithm is optimistic in exploration non-congested paths without realizing the causal impacts of interference and performance degradation. In practice, we have modified the throughput optimal routing strategy. This modification reduces the exploratory behaviour and the back-pressure ability of the optimal algorithm (Optimal algorithm tries to forward along high quality links if the a node experiences high congestion at the node itself. However, it can degrade the performance due to increased spreading of packets and self-interference.) as shown in Fig. A.1.



(a) CDF of delay (scenario 1)



(b) CDF of delay (scenario 2)

Figure A.1: Optimal vs. CDP performance

A.3.2 Optimization Perspective of CDP

We will now interpret the fixed point equation of CDP as a maximization problem of a utility maximization problem. Consider a single flow with s as the source node and d as the source node. Let y be the rate with which packets are injected at s, while x_{ij} denote the packet flow rate from node i to node j. Let $U : \mathbb{R} \to \mathbb{R}$ be a concave utility function. Consider the optimization problem **P1**:

$$\text{maximize } U(y) \tag{A.16}$$

$$\sum_j x_{ij} - \sum_j x_{ji} - y1_{\{i=s\}} = 0, \tag{A.17}$$

$$x_{ij} \geq 0, \tag{A.18}$$

$$\sum_j x_{ij} W_{ij} \leq T_i, \tag{A.19}$$

where $1_{\{i=s\}}$ = 1 if $i = s$ otherwise 0, while $T_i > 0$ represents the medium access (per unit time) for node i and Wij denote the average rate of transmission attempts for link ij. Constraint (A.19) imposes average constraint on channel access, which (though not exactly) achieved by the CSMA/CA protocol.[4] We now convert the problem into its dual and let λ_i denote the Lagrange multiplier for constraint (A.18) and μ_i denote the constraint associated with constraint (A.18) for node i.

$$\Lambda(y, \lambda, \mu) \;=\; U(y) - \sum_i \lambda_i (\sum_j x_{ij} - \sum_j x_{ji} \tag{A.20}$$

$$-y1_{\{i=s\}}) - \sum_i \mu_i \sum_{j \in \mathcal{N}(i)} x_{ij} W_{ij} \tag{A.21}$$

By KKT conditions, if $x_{ij} > 0$

$$\frac{(\lambda_i - \lambda_j)}{W_{ij}} = \mu_i \tag{A.22}$$

Under the stable and stationary state of the network, constraints (A.18) and (A.19) are satisfied with equality. Thus,

$$\lambda_i = \mu_i W_{ij} + \lambda_j \tag{A.23}$$

$\lambda_i = V(i)$ and $\mu_i = q_i$ satisfy (A.23) and optimaization problem **P1** is convex. Thus CDP maximizes the utility under flow constraints (A.18) (A.19).

A.4 Appendix for Chapter 5

We provide a sketch of the proof for the throughput optimality of D-ORCD for a connected network.[5]

[4]Assume that T_i is fixed and known.

[5]In a connected network each node has a positive probability path to a destination. If a node has no path to a destination, it cannot sustain any traffic and can be ignored without loss of generality.

A.4.1 Relationship to the Centralized ORCD

We prove the throughput optimality by relating the D-ORCD update equation (4.2) to the convergence of the closely related fixed point equation. In particular, we relate the routing decisions for D-ORCD with the decisions taken according to the congestion measures $\{V_i^*(t)\}$ obtained from the fixed point equation:

$$V_i^*(t) = \frac{Q_i(t)}{P^{(i,D)}(t)} + \sum_{k:H^{(i,D)}(t)} \frac{P_{succ-k}^{(i,D)}(t)}{P^{(i,D)}(t)} V_k^*(t). \tag{A.24}$$

We refer to the centralized routing algorithm which makes decisions at each instant according to $V_i^*(t)$ as C-ORCD. The following lemma states a relationship between D-ORCD and C-ORCD.

Lemma 11. *Assume T_c is sufficiently large ($T_c \sim O(D)$). Then during $T(t) \le t < T(t+T_c)$, (4.2) converges to the fixed point equation (A.24), i.e. $\{V_i^D(T(t+T_c)\}_{i\in\Omega}$ solves (A.24) and $V_i^D(T(t+T_c)) = V_i^*(T(t))$.*

Proof. The convergence $V_i^D(t) \to V_i^*(T(t))$ during $T(t) \le t < T(t+T_c)$ follows by relating (4.2) to the Bellman-Ford algorithm with fixed link cost. It is known from [73, Theorem 2.4] that an asynchronous distributed Bellman-Ford algorithm converges in finite time when the control packets are instantaneously received (or control packets are timestamped and older packets are discarded). Furthermore, with high probability, the time until the termination of the asynchronous Bellman-Ford algorithm is $\mathbf{O}(D)$, where D is the number of nodes in the network.

To implement the D-ORCD algorithm, we broadcast control packets using high priority and the control packets do not undergo backoff. This ensures with high probability that the packets are instantaneously received. Thus the convergence in Lemma 11 is justified. □

We now provide the proof for the throughput optimality of D-ORCD for $T_c \sim O(D)$. To simplify the notation, let policies in C-ORCD and D-ORCD be represented by π^* and $\hat{\pi}$ respectively. In [41], the authors constructed an appropriate Lyapunov function L^* to show that C-ORCD is throughput optimal. In particular, it is shown that:

Fact 9. There exists a Lyapunov function. $L^* : \mathbb{R}_+^D \to \mathbb{R}_+$ such that for all time slots t and $B > 0$, $\epsilon > 0$,

$$\mathbb{E}\left[L^*(\boldsymbol{Q}^{\pi^*}(t+1)) - L^*(\boldsymbol{Q}(t))|\boldsymbol{Q}(t)\right] \leq B - \epsilon \sum_{k=1}^{N} Q_k(t), \tag{A.25}$$

where superscript π^* in $\boldsymbol{Q}^{\pi^*}$ implies that routing decisions are made according to policy π^*.

To prove throughput optimality of D-ORCD, it suffices to show that the Lyapunov drifts under π^* and $\hat{\pi}$ have a bounded difference. More precisely,

Lemma 12. *Let L^* be the Lyapunov function as proposed in [41]. Then for $B'' > 0$,*

$$\mathbb{E}\left[L^*(\boldsymbol{Q}^{\hat{\pi}}(t+1))|\boldsymbol{Q}(t)\right] - \mathbb{E}\left[L^*(\boldsymbol{Q}^{\pi^*}(t+1))|\boldsymbol{Q}(t)\right] < B''.$$

Lemma 12 together with (A.25) implies the existence of a Lyapunov function with a negative difference i.e. for Lyapunov function L^*, there exists $B' > 0$ and $\epsilon' > 0$ such that,

$$\mathbb{E}\left[L^*(\boldsymbol{Q}^{\hat{\pi}}(t+1)) - L^*(\boldsymbol{Q}^{\hat{\pi}}(t))|\boldsymbol{Q}(t)\right] \leq B' - \epsilon' \sum_{k=1}^{N} Q_k(t).$$

The details of the construction of L^* and the proof of Lemma 12 are provided in Appendix A.4.3.

A.4.2 Review of C-ORCD results

Before proceeding, we introduce some notations. Let $[x]^+ = \max\{x, 0\}$. The indicator function $\mathbf{1}_{\{X\}}$ takes the value 1 whenever event X occurs, and 0 otherwise. For any set S, $|S|$ denotes the cardinality of S, while for any vector $\boldsymbol{v}$, $\|\boldsymbol{v}\|$ denotes the Euclidean norm of $\boldsymbol{v}$. When dealing with a sequence of sets $C_1, C_2, \ldots,$ we define $C^i = \cup_{j=1}^{i} C_j$.

The following definitions are required in order to identify the Lyapunov functions for C-ORCD and D-ORCD.

Definition 5. A *rank ordering* $R = (C_1, C_2, \ldots, C_M)$ is an ordered list of non-empty sets $C_1, C_2, \ldots, C_M$ $(1 \leq M \leq D)$, referred to as *ranking classes*, that make up a partition of the set of nodes $\{1, 2, \ldots, D\}$, i.e., $\cup_{i=1}^{M} C_i = \{1, 2, \ldots, D\}$ and $C_i \cap C_j = \emptyset$, $i \neq j$.

Definition 6. A rank ordering $R = (C_1, C_2, \ldots, C_M)$ is referred to as *path-connected* if for each node $i \in C_k$, $1 \leq k \leq M$, there exist distinct nodes $j_1, j_2, \ldots, j_l \in C^{k-1}$ such that $p_{ij_1} > 0, p_{j_1 j_2} > 0, \ldots, p_{j_l D} > 0$. The set of all path-connected rank orderings is denoted by $\mathcal{R}^c$.

Let f be a bivariate function of the following form:

$$f(m, n) = \frac{1}{K^m (K^n - 1)} \text{ for all } m \geq 0, n > 0,$$

where $K = 1 + \frac{1}{p_{\text{min}}}$ for $p_{\text{min}} = \min\{p_{ij} : i, j \in \Omega, p_{ij} > 0\}$.

In [41], the authors proposed a method that utilizes the bivariate function f and partitions the space of queue backlogs, $\mathbb{R}_+^D$, into $|\mathcal{R}^c|$ cones denoted by $\{D_f^c(R)\}_{R \in \mathcal{R}_c}$. The piece-wise Lyapunov function, $L_f^* : \mathbb{R}_+^D \to \mathbb{R}_+$, is then constructed by assigning to each cone $D_f^c(R)$, $R = (C_1, C_2, \ldots, C_M) \in \mathcal{R}_c$, a weighted quadratic function of the form:

$$L_f(\boldsymbol{Q}, R) = \sum_{i=1}^{M} f(|C^{i-1}|, |C_i|) Q_{C_i}^2, \tag{A.26}$$

where $Q_C(t) = \sum\limits_{i \in C} Q_i(t)$. More precisely,

$$L_f^*(\boldsymbol{Q}) = \sum_{R \in \mathcal{R}_c} L_f(\boldsymbol{Q}, R) \mathbf{1}_{\{\boldsymbol{Q} \in D_f^c(R)\}}. \tag{A.27}$$

Let us consider the Lyapunov drift when $\boldsymbol{Q}(t) \in D_f^c(R)$ for some

$$R = (C_1, C_2, \ldots, C_M) \in \mathcal{R}_c.$$

Since the collection of cones $\{D_f^c(R)\}_{R \in \mathcal{R}_c}$ partitions $\mathbb{R}_+^D$, we define function U_f : $\Omega \times \mathbb{R}_+^D \to \mathbb{R}_+$ such that

$$U_f(k, \boldsymbol{Q}) = f(|C^{i-1}|, |C_i|) Q_{C_i}(t), \tag{A.28}$$

where $\boldsymbol{Q} \in D_f^c(R)$, $R = (C_1, C_2, \ldots, C_M)$, and $k \in C_i$.

Let $\{\mu^*_{ij}(t)\}_{i,j\in\Omega}$ represent the routing decisions under C-ORCD, $\{\hat{\mu}_{ij}(t)\}_{i,j\in\Omega}$ represent the routing decisions under D-ORCD. For ease of notation and exposition define $A_C(t) = \sum\limits_{i\in C} A_i(t)$, $\mu_{C,in}(t) = \sum_{k=1}^{D}\sum_{j\in C}\mu_{kj}(t)\mathbf{1}_{\{Q_k(t)\geq 1\}}$, and $\mu_{C,out}(t) = \sum\limits_{k\in C}\sum_{j=1}^{D}\mu_{kj}(t)\mathbf{1}_{\{Q_k(t)\geq 1\}}$. We have,

$$
\begin{aligned}
&L^*_f(\boldsymbol{Q}(t+1)) - L^*_f(\boldsymbol{Q}(t)) \\
&\overset{(a)}{=} \sum_{i=1}^{M} f(|C^{i-1}|,|C_i|)\left[Q^2_{C_i}(t+1) - Q^2_{C_i}(t)\right] \\
&\qquad + O(\|\boldsymbol{Q}(t+1) - \boldsymbol{Q}(t)\|^2) \\
&\overset{(b)}{=} -2\sum_{i=1}^{M} f(|C^{i-1}|,|C_i|)Q_{C_i}(t)\Big(\mu_{C_i,out}(t) - \mu_{C_i,in}(t) \\
&\qquad - A_{C_i}(t)\Big) + O(1) \\
&= -2\sum_{i=1}^{M} f(|C^{i-1}|,|C_i|)Q_{C_i}(t)\mathbf{1}_{\{Q_k(t)\geq 1\}} \\
&\qquad \Big(\sum_{j=1}^{D}\sum_{k\in C_i}\mu_{kj}(t) - \sum_{k=1}^{D}\sum_{j\in C_i}\mu_{kj}(t) - A_{C_i}(t)\Big) + O(1),
\end{aligned} \tag{A.29}
$$

where (a) follows from the continuity and the differentiability of L^*_f [41, Lemma 3] and writing $L^*_f(\boldsymbol{Q}(t+1))$ in terms of its first-order Taylor expansion around $L^*_f(\boldsymbol{Q}(t))$, and (b) follows from Fact 10 below.

Fact 10 ([41])**.** Let $R = (C_1, C_2, \ldots, C_M) \in \mathcal{R}$ and $\boldsymbol{Q}(t) \in D_f(R)$. We have

$$
\begin{aligned}
&Q^2_{C_i}(t+1) - Q^2_{C_i}(t) = \\
&\qquad \beta_f - 2Q_{C_i}(t)(\mu_{C_i,out}(t) - \mu_{C_i,in}(t) - A_{C_i}(t)),
\end{aligned}
$$

where β_f is a constant bounded real number.

Finally from (A.28) and (A.29),

$$\begin{aligned}
&L_f^*(\boldsymbol{Q}(t+1)) - L_f^*(\boldsymbol{Q}(t)) \\
&= -2\sum_{i=1}^{M}\sum_{k\in C_i}\sum_{l=1}^{D} U_f(k,Q)\mu_{kl}(t)\mathbf{1}_{\{Q_k(t)\geq 1\}} \\
&- \sum_{i=1}^{M}\sum_{l\in C_i}\sum_{k=1}^{D} U_f(l,Q)\mu_{kl}(t)\mathbf{1}_{\{Q_k(t)\geq 1\}} - \sum_{i=1}^{M} A_{C_i}(t) + O(1) \\
&= -2\sum_{k=1}^{D}\sum_{l=1}^{D} U_f(k,Q)\mu_{kl}(t)\mathbf{1}_{\{Q_k(t)\geq 1\}} \\
&\quad - \sum_{k=1}^{D}\sum_{l=1}^{D} U_f(l,Q)\mu_{kl}(t)\mathbf{1}_{\{Q_k(t)\geq 1\}} - \sum_{i=1}^{M} A_{C_i}(t) + O(1). \qquad \text{(A.30)}
\end{aligned}$$

Fact 11 ([41]). Routing decisions under C-ORCD are such that $\mu^*_{ij} = 1$, only when $j \in S_i(t)$ and $U_f(j, \boldsymbol{Q}(t)) \leq U_f(k, \boldsymbol{Q}(t))$ for all $k \in S_i(t)$.

This fact together with (A.30) provides the proof of Fact 9.

A.4.3 Proof of Lemma 12

We begin the proof of Lemma 12 by stating the following Claim.

Claim 1. *Routing decisions under D-ORCD are such that $\hat{\mu}_{ij} = 1$, only when $j \in S_i(t)$ and $U_f(j, \hat{\boldsymbol{Q}}(t)) \leq U_f(k, \hat{\boldsymbol{Q}}(t))$ for all $k \in S_i(t)$, where $\hat{\boldsymbol{Q}}(t) = \bar{\boldsymbol{Q}}(T(t))$.*

With this, we are ready to proceed with the proof of Lemma 12.

$$\begin{aligned}
&\mathbb{E}[L_f^*(\boldsymbol{Q}^{\hat{\pi}}(t+1))|\boldsymbol{Q}(t)] - \mathbb{E}[L_f^*(\boldsymbol{Q}^{\pi^*}(t+1))|\boldsymbol{Q}(t)] \\
&= \mathbb{E}\Big[2\sum_{k=1}^{D}\sum_{l=1}^{D}(\hat{\mu}_{kl}(t) - \mu^*_{kl}(t))\Big(U_f(k,\boldsymbol{Q}(t)) \\
&\qquad - U_f(l,\boldsymbol{Q}(t)\Big)\mathbf{1}_{\{Q_k(t)\geq 1\}}|\boldsymbol{Q}(t)\Big] + O(1), \qquad \text{(A.31)}
\end{aligned}$$

where the equality follows from (A.30).

Suppose node i's transmission at time t is received by potential forwarders $S_i(t)$. Furthermore, suppose that nodes $a, b \in S_i(t)$ are the nodes with the highest rank under

C-ORCD and D-ORCD respectively, i.e. $\mu^*_{ia}(t) = \hat{\mu}(t)_{ib} = 1$. From Fact 11 and Claim 1, we have

$$\begin{aligned} U_f(a, \boldsymbol{Q}(t)) &\leq U_f(b, \boldsymbol{Q}(t)), & \text{(A.32)} \\ U_f(a, \hat{\boldsymbol{Q}}(t)) &\geq U_f(b, \hat{\boldsymbol{Q}}(t)). & \text{(A.33)} \end{aligned}$$

In order to prove Lemma 12 it suffices to show that

$$U_f(b, \boldsymbol{Q}(t)) - U_f(a, \boldsymbol{Q}(t)) = O(\|\boldsymbol{Q}(t) - \hat{\boldsymbol{Q}}(t)\|). \tag{A.34}$$

Consider the line that connects $\boldsymbol{Q}(t)$ and $\hat{\boldsymbol{Q}}(t)$ in $\mathbb{R}^D_+$. Suppose this line goes through $M-1$ cones in $\mathbb{R}^D_+$. Let $Z_1, Z_2, \ldots, Z_M$ be respectively the intersection of the line connecting $\boldsymbol{Q}(t)$ to $\hat{\boldsymbol{Q}}(t)$ with the M separating hyperplanes of the $M-1$ cones between them, i.e.

$$\begin{aligned} &\|\boldsymbol{Q}(t) - \hat{\boldsymbol{Q}}(t)\| \\ &= \|\boldsymbol{Q}(t) - \boldsymbol{Z_1}\| + \|\boldsymbol{Z_1} - \boldsymbol{Z_2}\| + \cdots + \|\boldsymbol{Z_M} - \hat{\boldsymbol{Q}}(t)\|. \end{aligned}$$

Note that since $Z_1, Z_2, \ldots, Z_M$ are on the hyperplanes, every two consecutive points in set $\{\boldsymbol{Q}(t), Z_1, Z_2, \ldots, Z_M, \hat{\boldsymbol{Q}}(t)\}$ can be considered to belong to the same cone, and hence, have same rank ordering of the nodes. From the definition of the function U_f, we obtain:

$$\begin{aligned} |U_f(a, \boldsymbol{Q}(t)) - U_f(a, \boldsymbol{Z_1})| &= O(\|\boldsymbol{Q}(t) - \boldsymbol{Z_1}\|), \\ |U_f(a, \boldsymbol{Z_m}) - U_f(a, \boldsymbol{Z_{m+1}})| &= O(\|\boldsymbol{Z_m} - \boldsymbol{Z_{m+1}}\|),\ 1 \leq m \leq M, \\ |U_f(a, \boldsymbol{Z_M}) - U_f(a, \hat{\boldsymbol{Q}}(t))| &= O(\|\boldsymbol{Z_M} - \hat{\boldsymbol{Q}}(t)\|). \end{aligned}$$

Therefore,

$$\begin{aligned} &U_f(a, \boldsymbol{Q}(t)) - U_f(a, \hat{\boldsymbol{Q}}(t)) \\ &= [U_f(a, \boldsymbol{Q}(t)) - U_f(a, \boldsymbol{Z_1})] + [U_f(a, \boldsymbol{Z_1}) - U_f(a, \boldsymbol{Z_2})] \\ &\quad + \cdots + [U_f(a, \boldsymbol{Z_M}) - U_f(a, \hat{\boldsymbol{Q}}(t))] = O(\|\boldsymbol{Q}(t) - \hat{\boldsymbol{Q}}(t)\|). \end{aligned}$$

We can derive the same result for all other nodes in the network. In other words, there exist constants η_a, η_b such that

$$\begin{aligned} U_f(a, \boldsymbol{Q}(t)) &= U_f(a, \hat{\boldsymbol{Q}}(t)) + \eta_a \|\boldsymbol{Q}(t) - \hat{\boldsymbol{Q}}(t)\|, & \text{(A.35)} \\ U_f(b, \boldsymbol{Q}(t)) &= U_f(b, \hat{\boldsymbol{Q}}(t)) + \eta_b \|\boldsymbol{Q}(t) - \hat{\boldsymbol{Q}}(t)\|. & \text{(A.36)} \end{aligned}$$

However, (A.32), (A.33), (A.35), and (A.36) imply that $\eta_a \leq \eta_b$ and

$$U_f(b, \boldsymbol{Q}(t)) - U_f(a, \boldsymbol{Q}(t)) \leq (\eta_b - \eta_a)\|\boldsymbol{Q}(t) - \hat{\boldsymbol{Q}}(t)\|. \tag{A.37}$$

With this, the proof is now complete.

Appendix B

B.1 CDP Implementation description

B.1.1 Testbed Setup

The testbed consists of 12 wireless Alix nodes with 512 MB RAM, a 500 MHz processor and Linux version 2.6.22. The nodes placed as shown in Figure 5.2, are distributed in Atkinson Hall, UCSD, in about 215,000 sq. ft. of space. Each node is equipped with an Atheros based 802.11 a/b/g wireless interface (AR 5213). The nodes are connected to the internet through a Power Over Ethernet router: 137.110.119.80 located on calit 4-th floor. The nodes are assigned IP address in the range 172.19.119.201: 172.19.119.222 for the Ethernet cards. Out of these nodes only 12 nodes are selected for the purpose of the experiment.

The nodes use a low end operating system based on Linux for x86 processors http://linux.voyage.hk/. Each node has a configuration file in /etc/CDP which it uses at the start-up. Based on the nodeID, a wireless IP is selected as 192.168.3.NodeID. /CDP/parameters is the parameter file used to store the nodeID and other constant parameters used during the experiments. The nodes are set to the 802.11g mode at the startup.

The nodes are controlled from server ansel.ucsd.edu. For the following text, we assume that the controlling scripts are stored at /home/bhorkar. The mapping between node ID, wireless IP, wired IP, MAC addresses is stored in /home/bhorkar/Listnodes.

B.1.2 Configuration files

For the purpose of testing, we generate UDP and TCP traffic using a modified version of iperf. The modified version is stored in

```
~/thrulay_dir/src
```

The application traffic can be generated using a Configuration file referred to as Appfile. In this file we store the information about the source Wireless IP, Destination Wireless IP, source Wire IP, destination IP, duration of experiment and type of traffic.

Example 1

Let us write the script to generate UDP at a rate of 100KBps from source node 10 to destination node 17. The wireless ip and wired ip for node 10 is 192.168.3.10 and 172.19.119.209. The wireless ip and wired ip for node 10 is 192.168.3.17 and 172.19.119.221.

The configuration line in the Appfile is:

Src IP	Dst IP	Duration(sec)	Traffic	Type
192.168.3.10	192.168.3.17	60	100K	UDP

If we add a TCP flow from 10-17, then the configuration file will look like

Src IP	Dst IP	Duration(sec)	Traffic	Type
192.168.3.10	192.168.3.17	60	100K	UDP
192.168.3.10	192.168.3.17	60	0	TCP

We can further specify any particular TCP protocol of interest as

192.168.3.10	192.168.3.17	60	0	TCP	Veno

Here TCP Veno can be replaced with any known TCP protocol.

This file is then fed as input to script which sets up the various parameters for the experiments.

```
~/setup_CDP_mix.py -a Appfile -p protocol
```

Many other options such as power level and data rate of the transmitter can also be set as parameter.

B.1.3 Routing Layer

The main routing layer is implemented in the following file.

```
madwifi-CDP/tools/CDP.c
```

This file contains the implementation of CDP and various other protocols. Important functions and their objectives are listed below:

server	handles different received packets
client	generates various control and probe packets
CDPInit	initializes various parameters
CDPCalculateandUpdateVn	updates V_n for different protocols
CDPAlixIoctlPrioList	generates a next hop and feeds it to kernel
CDPCreatePriorityListUpdatePacket	generates control packet

B.1.4 MAC layer

The MAC driver is based on the modifications of the following file

```
~/madwifi-ng_CDP_over_802.11/ath/if_ath.c
```

The current file reflects changes in the version 4307 of the madwifi-driver. Important functions and their objectives are listed below:

$calc - usecs - unicast - packet - wait$	Calculates waiting time for different neighbours
$ath - get - nexthop$	This function obtains the next hop
$ath - get - nexthop$	for a given destination

Experimental setup

Before we setup any experiment, the following steps need to be performed:

```
make ~/madwifi-ng_CDP_over_802.11/
make  madwifi-CDP/tools/
 ~/setup.py
```

setup.py is a script which copies all the driver files and the routing files to each of the nodes.

Bibliography

[1] Parul Gupta and Tara Javidi, " Towards Throughput and Delay Optimal Routing for Wireless Ad-Hoc Networks," in *Asilomar Conference*, November 2007, pp. 249–254.

[2] R. Bruno, M. Conti, and E. Gregori, "Mesh networks: commodity multihop ad hoc networks," *IEEE Commmunication Magazine*, vol. 43, no. 3, pp. 123–131, March 2005.

[3] D. S. J. De Couto, D. Aguayo, J. Bicket, and R. Morris, "A High Throughput Path Metric for Multi-hop Wireless Routing," in *ACM Mobicom*, 2003.

[4] G. Pei, M. Gerla, and T.W. Chen, "Fisheye state routing in mobile ad hoc networks," in *ICDCS Workshop on Wireless Networks and Mobile Computing*, Apr 2000, p. D71D78.

[5] R. Draves, J. Padhye, and B. Zill, "Routing in multi-radio, multi-hop wireless mesh networks," in *ACM MOBICOM*, Sept 2004, pp. 114–128.

[6] John Bicket, Daniel Aguayo, Sanjit Biswas, and Robert Morris, "Architecture and evaluation of an unplanned 802.11b mesh network," in *Proceedings of the 11th annual international conference on Mobile computing and networking , Cologne, Germany*, August 28-September 02, 2005.

[7] C. Lott and D. Teneketzis, "Stochastic routing in ad hoc wireless networks," *Decision and Control, 2000. Proceedings of the 39th IEEE Conference on*, vol. 3, pp. 2302–2307, 2000.

[8] P. Larsson, "Selection Diversity Forwarding in a Multihop Packet Radio Network with Fading channel and Capture," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 2, no. 4, pp. 4754, October 2001.

[9] M. Zorzi and R. R. Rao, "Geographic Random Forwarding (GeRaF) for Ad Hoc and Sensor Networks: Multihop Performance," *IEEE Transactions on Mobile Computing*, vol. 2, no. 4, 2003.

[10] S. Biswas and R. Morris, "ExOR: Opportunistic Multi-hop Routing for Wireless Networks," *ACM SIGCOMM Computer Communication Review*, vol. 35, pp. 33–44, October 2005.

[11] S.R. Das S. Jain, "Exploiting path diversity in the link layer in wireless ad hoc networks," *World of Wireless Mobile and Multimedia Networks, 2005. WoWMoM 2005. Sixth IEEE International Symposium on a*, pp. 22–30, June 2005.

[12] C. Lott and D. Teneketzis, "Stochastic routing in ad-hoc networks," *IEEE Transactions on Automatic Control*, vol. 51, pp. 52–72, January 2006.

[13] M. J. Neely and R. Urgaonkar, "Optimal backpressure Routing for Wireless Networks with Multi-Receiver Diversity," *IEEE Transactions on on Information Theory*, pp. 862–881, July 2009.

[14] 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. 19361949, August 1992.

[15] S. Sarkar and S. Ray, "Arbitrary throughput versus complexity tradeoffs in wireless net- works using graph partitioning," *IEEE Transactions on Automatic Control*, 2008.

[16] L. Ying and S. Shakkottai, "on throughput-optimal scheduling with delayed channel state feedback," in *Information Theory and Applications Workshop*, 2008.

[17] Y. Xi and E. M. Yeh, "Throughput optimal distributed control of stochastic wireless networks," in *WiOpt*, 2006.

[18] B. Smith and B. Hassibi, ", wireless erasure networks with feedback," arXiv:0804.4298v1, 2008.

[19] F. P. Kelly, A. Maulloo, and D. Tan, "Rate Control in Communication Networks: Shadow Prices, Proportional Fairness and Stability," *Journal of the Operational Research Society*, 1998.

[20] Y. Yi and S. Shakkottai, "Hop-by-hop congestion control over a wireless multi-hop network," *IEEE/ACM Transactions on Networking*, 2007.

[21] T. Javidi and D. Teneketzis, " Sensitivity Analysis for Optimal Routing in Wireless Ad Hoc Networks in Presence of Error in Channel Quality Estimation," *IEEE Transactions on Automatic Control*, pp. '1303–1316, August 2004.

[22] J.N. Tsitsiklis, "Asynchronous Stochastic Approximation and Q-learning," *Proceedings of the 32nd IEEE Conference on Decision and Control*, vol. 1, pp. '395–400, Dec 1993.

[23] J. Boyan and M. Littman, "Packet routing in dynamically changing networks: A reinforcement learning approach," in *NIPS*, 1994, pp. 671–678.

[24] J. W. Bates, "Packet Routing and Reinforcement Learning: Estimating Shortest Paths in Dynamic Graphs," unpublished manuscript, 1995.

[25] S. Choi and D. Yeung, "Predictive Q-routing: A memory-based Reinforcement Learning Approach to Adaptive Traffic Control," in *Advances in Neural Information Processing Systems*, 1996, pp. 945–951.

[26] S. Kumar and R. Miikkulainen, "Dual Reinforcement Q-Routing: An On-line Adaptive Routing Algorithm," in *Smart Engineering Systems: Neural Networks, Fuzzy Logic, Data Mining, and Evolutionary Programming*, 2000, pp. 2104– 2125.

[27] S. S. Dhillon and P. Van Mieghem, "Performance Analysis of the AntNet Algorithm," in *Computer Networks*, 2007, vol. 51, pp. 2104– 2125.

[28] P. Purkayastha and J. S. Baras, "Convergence of Ant Routing Algorithm via Stochastic Approximation and Optimization," in *IEEE Conf. on Decision and Control*, 2007, pp. 340–354.

[29] D.P. Bertsekas and J.N. Tsitsiklis, *Neuro-Dynamic Programming*, Athena Scientific, Belmont, Massachusetts, 1996.

[30] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, "Trading Structure for Randomness in Wireless Opportunistic Routing," in *In Proc. of ACM SIGCOMM*, 2007.

[31] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, New York: John Wiley & Sons, 1994.

[32] Dimitri P. Bertsekas and John N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, Athena Scientific, 1997.

[33] W. Stallings, *Wireless Communications and Networks*, Prentice Hall, second edition, 2004.

[34] M. Kurth, A. Zubow, and J. P. Redlich, "Cooperative Opportunistic Routing Using Transmit Diversity in Wireless Mesh Networks ," in *INFOCOM*, April 2008, pp. 1310–1318.

[35] J. Doble, *Introduction to Radio Propagation for Fixed and Mobile Communications*, Artech House, Boston, 1996.

[36] S. Russel and P. Norvig, *Artificial Intelligence: A Modern Approach*, Upper Saddle River, NJ: Prentice Hall, 2nd edition, 2003.

[37] R. Parr and S. Russell, “Reinforcement Learning with Hierarchies of Machines,” in *Proceedings of the 1997 conference on Advances in neural information processing systems 10 (NIPS)*, 1998, pp. 1043–1049.

[38] T.L. Lai and H. Robbins, “Asymptotically efficient adaptive allocation rules,” *Advances in Applied Mathematics*, vol. 6, pp. 4–22, 1985.

[39] Apostolos N. Burnetas and Michael N. Katehakis, “ Optimal adaptive policies for Markov deci- sion processes,” *Mathematics of Operation Research*, 1997.

[40] Ambuj Tewari and Peter L. Bartlett, “Optimistic linear programming gives logarithmic regret for irreducible mdps,” in *In Proceedings of Neural Information Processing Systems Conference (NIPS*, 2007.

[41] M. Naghshvar, H. Zhuang, and T. Javidi, “A general class of throughput optimal routing policies in multi-hop wireless networks,” arXiv:0908.1273v1.

[42] E. Leonardi, M. Mellia, M. A. Marsan, and F. Neri, “optimal scheduling and routing for maximum network throughput”,” *IEEE/ACM Transactions on Networking*, 2007.

[43] S.Shakkottai L.Ying and A. Reddy, “On Combining Shortest-Path and Back-Pressure Routing Over Multihop Wireless Networks,” *IEEE/ACM Transactions on Networking*, vol. 19, June 2011.

[44] L. Huang, S. Moeller, M.J. Neely, and B. Krishnamachari, “LIFO-Backpressure Achieves Near Optimal Utility-Delay Tradeoff,” in *Proc. of 9th Intl. Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, 2011.

[45] Aman Shaikh, Anujan Varma, Lampros Kalampoukas, and Rohit Dube, “Routing stability in congested networks: Experimentation and analysis,” in *In Proc. ACM SIGCOMM*, 2000, pp. 163–174.

[46] J.M. Jaffe and F. Moss, “A Responsive Distributed Routing Alorithm for Computer Networks,” *IEEE Transactions on Communications*, vol. 30, July 1982.

[47] Uyless Black, *IP routing protocols: RIP, OSPF, BGP, PNNI and Cisco routing protocols*, Prentice Hall PTR, Upper Saddle River, NJ, USA, 2000.

[48] M. J. Neely, “Optimal Backpressure Routing for Wireless Networks with Multi-Receiver Diversity,” in *Conference on Information Sciences and Systems (CISS)*, March 2006.

[49] Hairuo Zhuang, *Throughput optimal routing in wireless Ad-hoc networks*, Ph.D. thesis, University of California, San Diego, 2010.

[50] M. Naghshvar and T. Javidi, " Opportunistic Routing with Congestion Diversity in Wireless Multi-hop Networks," *IEEE INFOCOM*, march 2010.

[51] A. Bhorkar and M. Naghshvar and T. Javidi, " Opportunistic Routing with Congestion Diversity in Wireless Multi-hop Networks," *Submitted to Transactions on Networking*.

[52] "Alix, `http://www.pcengines.ch/alix.htm`,".

[53] Libin Jiang and J. Walrand, "A Distributed CSMA Algorithm for Throughput and Utility Maximization in Wireless Networks," in *Communication, Control, and Computing, 2008 46th Annual Allerton Conference on*, Sept. 2008, pp. 1511–1519.

[54] Bret Hull, Kyle Jamieson, and Hari Balakrishnan, "Mitigating congestion in wireless sensor networks," in *in SenSys*. 2004, pp. 134–147, ACM Press.

[55] Rafael Laufer, Theodoros Salonidis, Henrik Lundgren, and Pascal Le Guyadec, "XPRESS: A Cross-Layer Backpressure Architecture for Wireless Multi-Hop Networks," in *ACM MobiCom 2011*, pp. 49–60.

[56] A. Warrier, S. Janakiraman, Ha Sangtae, and I. Rhee, "Diffq: Practical differential backlog congestion control for wireless networks," in *INFOCOM 2009, IEEE*, April 2009, pp. 262–270.

[57] Scott Moeller, Avinash Sridharan, Bhaskar Krishnamachari, and Omprakash Gnawali, "Routing Without Routes: The Backpressure Collection Protocol," in *submitted ACM/IEEE International Conference on Information Processing in Sensor Networks*, 2010.

[58] L. Wang, L. Zhang, Y. Shu, and M. Dong, "Multipath Source Routing in Wireless Ad Hoc Networks," in *Canadian Conf. Electrical and Computer Engineering*, 2000.

[59] S. J. Lee and M. Gerla, "Split multipath routing with maximally disjoint paths in ad hoc networks," in *ICC*, 2001, vol. 10, pp. 3201–3205.

[60] P. Papadimitratos, Z. J. Haas, and E. G. Sirer, "Path set selection in mobile ad hoc networks," in *ACM MobiHOC 2002*, 2002, pp. 160–170.

[61] Nagesh S. Nandiraju, Deepti S. Nandiraju, and Dharma P. Agrawal, "Multipath routing in wireless mesh networks," in *Mobile Adhoc and Sensor Systems (MASS), 2006 IEEE International Conference on*, Oct. 2006, pp. 741–746.

[62] M.K. Marina and S.R. Das, "On-demand multipath distance vector routing in ad hoc networks," in *Network Protocols, 2001. Ninth International Conference on*, Nov. 2001, pp. 14–23.

[63] H. Han and S. Shakkottai and C. V. Hollot and R. Srikant and D. Towsley, “Multi-path TCP: A joint congestion control and routing scheme to exploit path diversity on the internet,” *IEEE/ACM Transactions on Networking*, vol. 14, pp. 1260–1271, 2006.

[64] Radunovic C. Gkantsidis and D. Gunawardena, “Horizon: Balancing tcp over multiple paths in wireless mesh network,” in *ACM MOBICOM*.

[65] Perahia and Stacey, *Next Generation Wireless LANs*, Cambridge University Press, 2008.

[66] John Bicket, “Bit-rate selection in wireless networks,” M.S. thesis, Massachusetts Institute of Technology, 2005.

[67] Kyu han Kim and Kang G. Shin, “On accurate measurement of link quality in multi-hop wireless mesh networks,” in *In ACM MobiCom 06*, 2006, pp. 38–49.

[68] C. P. Fu and S. C. Liew, “TCP Veno: TCP Enhancement for Transmission over Wireless Access Networks,” *IEEE JSAC*, vol. 21, no. 2, pp. 216–228, Feb 2004.

[69] “Ad hoc On-Demand Distance Vector (AODV) Routing,” RFC 3561, 2003.

[70] C. Steigner, H. Dickel, and T. Keupen, “RIP-MTI: A New Way to Cope with Routing Loops,” in *Seventh International Conference on Networking*, A pril 2008, pp. 626 –632.

[71] L. Breiman, *Probability*, Philadelphia, Pennsylvania: Society for Industrial and Applied Mathematics, 1992.

[72] S. Resnick, *A Probability Path*, Birkhuser, Boston, 1998.

[73] J. N. Tsitsiklis and G. D. Stamoulis, “On the Average Communication Complexity of Asynchronous Distributed Algorithms,” *Journal of the ACM*, pp. 382 – 400, 1995.