

# Achieving Congestion Diversity in Wireless Ad-hoc Networks

A. A. Bhorkar\*, T. Javidi\*

\*Department of Electrical Engineering,  
University of California San Diego, La Jolla, CA 92093  
abhorkar@ucsd.edu, tjavidi@ucsd.edu,

A. C. Snoeren†

†Department of Computer Science & Engineering,  
University of California San Diego, La Jolla, CA 92093  
snoeren@cs.ucsd.edu

**Abstract**—This work presents the Congestion Diversity Protocol (CDP), a routing protocol for multi-hop wireless networks that combines important aspects of shortest-path and backpressure routing to achieve improved end-end delay performance. In particular, CDP delivers lower end-to-end delay and fewer packet drops than existing routing protocols while maintaining equivalent throughput. This paper reports on a practical (hardware and software) implementation of CDP in an indoor WiFi network consisting of 12 802.11g nodes. This small test-bed enables an empirical comparison of CDP’s performance against a set of state of the art protocols which include both congestion unaware and congestion aware routing protocols. In most topologies and scenarios we consider, CDP provides improvements for UDP traffic with respect to both end-end delay and throughput over the existing protocols.

## I. INTRODUCTION

Wireless traffic demands continue to increase and show no signs of abating. Higher loads will lead to congestion—and, therefore, delay—when a network’s ingress traffic approaches its carrying capacity. This problem is particularly acute in multi-hop networks, where nodes must relay each others packets. We present a congestion aware routing protocol, the Congestion Diversity Protocol (CDP), which attempts to decrease delay by routing packets along paths with low queuing delay. Our main contribution is the introduction of a time-varying distance metric that approximates the queue draining times along each path. Our proposed “congestion measure” quantifies the congestion and path diversity associated with relaying via various nodes and enables a significantly lower end-end delay than existing protocols. In this work we restrict our attention strictly to the routing layer as it operates over the standard 802.11 MAC layer.

We build upon two classes of existing multi-hop routing protocols. The first class of protocols attempts to decrease the number of transmissions required to relay a packet to its destination. For example, SRCR [1] extends shortest path routing by selecting paths based upon the expected of number of transmission attempts (ETX [2]). Similar solutions are proposed in [3]–[6] where packets are routed along the shortest path to the destination under all congestion levels. This approach does not utilize multiple paths and can potentially result in increased delay, congestion, or even queue instability and buffer overflow. In contrast, the second class of algorithms, exemplified by the well known Backpressure (BP) routing algorithm [7] uses differential backlogs at nodes to make rout-

ing decisions and has been proven to be throughput optimal (i.e., ensure bounded expected delay for all satisfiable traffic). In contrast to ETX, BP ignores the distance of the potential forwarders to the destination and instead balances the queue backlogs in every locations. Combining these two approaches, Enhanced Backpressure (E-BP) [8] relies on the sum of ETX and backlog differentials at forwarding nodes. E-BP remains throughput optimal while taking topology information into account. As shown later, however, these existing throughput optimal routing policies distribute the traffic locally in every neighborhood, potentially resulting in large expected delay. Moreover, when run over an 802.11 MAC, the increased contention leads to drops due to excessive retransmission attempts and overflowing queues.

Instead of considering transmission and queuing separately, we propose to combine them by focusing on a path’s estimated delivery time. CDP is a distance vector protocol that routes packets through neighbors with the least estimated remaining delivery time at each hop. We evaluate our implementation of CDP in a 12-node indoor wireless test-bed consisting of embedded Alix nodes [9]. Our results reveal that for constant bitrate UDP traffic loads that are less than the capacity of the network, congestion diversity routing reduces delay, decreases packet drop rate, and increases throughput in comparison with BP, E-BP and SRCR.

The remainder of this paper is organized as follows. In Section II, we demonstrate the drawbacks of the existing routing protocols (SRCR, BP, E-BP) with an illustrative example. Section II-B introduces our proposed routing algorithm, CDP. In Section III, we discuss various implementation issues of CDP. Section IV provides the performance result for UDP on our test-bed. Finally, we conclude the paper in Section V.

## II. ROUTING WITH CONGESTION DIVERSITY

We first motivate our work by highlighting the shortcomings of both existing routing paradigms, shortest path and backpressure routing. Figure 1 shows an example network that illustrates the drawbacks of existing protocols such as SRCR, BP and E-BP.

### A. Motivating Example

Consider the operation of the network in Figure 1 under constant bitrate UDP traffic (i.e., packet generation at the application layer is unregulated) generated at nodes A and

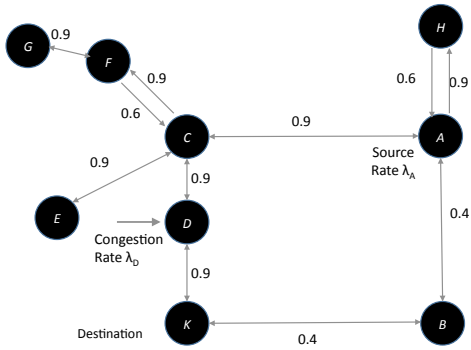


Fig. 1. Example Network

D at rates  $\lambda_A$  and  $\lambda_D$  respectively. Consider the traffic from source node A destined for node K. We first note that if  $\lambda_D$  is large, it might not be wise (or even feasible) to rely on the path A-C-D-K (ETX = 3.3) over A-B-K (ETX = 5). This clearly underlines the shortcomings of SRCR which routes along the paths with minimum ETX independent of the congestion state. It is also intuitive that any packet from the source node A to the destination K, which is relayed through nodes E, F, G or H will incur an unnecessary delay (note that all such packets must traverse back to node C eventually). BP is susceptible to unnecessary delay at nodes E, F, and G when trying to avoid congestion at node D. Here we also see the shortcoming of E-BP: potentially the addition of ETX to the backlog differential may further favor the choice of node C over B (which in case of large  $\lambda_D$  and small queue build-up at node C further increases the number of packets which end up in the nodes E, F and G). In Section IV, we quantitatively document the shortcomings of SRCR, BP and E-BP on our test-bed.

### B. CDP Design

CDP's design is based on the observation that when a packet arrives at a node,  $n$ , it spends a waiting time equal to the duration of the time spent draining the packets that have arrived earlier plus its own transmission time. Assuming a persistent ALOHA MAC one can approximate the expected waiting time at node  $n$  by  $\frac{q_t(n)}{p(n,k)}$ , where  $q_t(n)$  is the total number of packets at time  $t$  awaiting transmission at node  $n$ 's FIFO buffer, and  $p(n,k)$  is the probability that a packet transmitted by node  $n$  is successfully received by node  $k$ . CDP uses this approximate draining time to build a global distance-vector associated with various destinations. While similar in spirit to E-BP, our approach differs in which the backlog information of BP [10] is integrated with the shortest path computation of ETX [1]. More precisely, rather than treating local backlog information and ETX computation separately and then selecting an appropriate linear combination as the distance measure of interest, CDP first modulates the expected number of transmissions by the total number of packets currently waiting at a given node to obtain a local draining time estimate, which is subsequently used to build a global distance metric in a fashion similar to distributed Bellman-Ford.

More formally, CDP is a distributed asynchronous distance vector algorithm under which the routing decisions are made based on a time varying congestion-aware distance vector metric, referred to as the *congestion measure*. We denote the congestion measure associated with node  $k$  at time  $t$  and destination  $d$  as  $V_t^d(k)$ . It provides an estimate of the best possible draining time of a packet arriving at node  $k$  until it reaches destination  $d$ .

Before proceeding, we introduce some notation. We assume a network of  $N+1$  nodes labelled by  $\Omega = \{0, 1, \dots, N\}$ , with a subset  $D \subseteq \Omega$  of destination nodes. Let  $0 \leq \gamma < 1$  be a fixed link-quality threshold and  $\mathcal{N}_\gamma(n) := \{k : p(n,k) > \gamma\}$  be the set of neighbors of node  $n$  with a sufficiently reliable link. Let  $q_t(n)$  is the number of packets at time  $t$  awaiting transmission at node  $n$ .

CDP maintains a routing vector containing the congestion measure  $V_t^d$  routing packets to destination  $d \in D$  for node  $n \in \Omega$ . We calculate the congestion measure  $V_t^d$  in a distributed and iterative manner. At each time  $t$ , node  $n$ ,  $n \neq d$  computes its congestion measure  $V_t^d(n)$  ( $V_t^d(d) = 0$ ) associated with reaching destination  $d$  as

$$V_t^d(n) = \left\{ \frac{q_t(n) + 1}{p(n, K_{CDP}^{(n,d)})} + \tilde{V}_t^d(k_{CDP}^{(n,d)}) \right\}, \quad (1)$$

where  $K_{CDP}^{(n,d)}$  is the best next hop given as

$$K_{CDP}^{(n,d)} = \arg \min_{k \in \mathcal{N}_\gamma(n)} \left\{ \frac{1}{p(n,k)} + V_t^d(k) \right\}, \quad (2)$$

$\tilde{V}_t^d(k) = V_{t-\tau_k}^d(k)$ , and  $\tau_k \leq t$  is the most recent update node  $n$  received from neighbor  $k$ .

### III. IMPLEMENTATION DETAILS

In the following subsections, we discuss the elements of the protocol responsible for the computation of the congestion measure vectors and link probabilities, as well as the control mechanism responsible for the advertisement of the cost vector.

#### A. Protocol Components

1) *CDP routing table*: At each node  $n$  CDP utilizes a list of the neighbors' information at each node  $n$  in the form of routing table. In our implementation, the routing table at each node  $n$  consists of the number of neighbors, a neighbor list, and the best next hop vector  $\{K_{CDP}^{(n,d)}\}_{d \in D}$ . Each neighbor list entry is a structure associated with neighbor  $k \in \mathcal{N}_\gamma(n)$  and consists of that neighbor's IP address, the estimated link probability  $p(n,k)$  and the congestion measure vector  $\{V_t^d(k)\}_{d \in D}$ . Under CDP, node  $n$  periodically uses the entries of the congestion measure vectors embedded in its neighbor lists to compute its own congestion measure vector using (1), which is subsequently advertised to its neighbors. This update, similarly, is used by the neighboring nodes to update their congestion measures and their next hop.

2) *Control Plane*: CDP relies on the periodic exchange and reevaluation of congestion measure among neighbors using control packets. The control packet at node  $n$  consists of of the congestion measure  $\{V_t^d(n)\}$  for each destination  $d \in D$ . The periodic computation and communication of cost vectors propagates routing information, in a manner that is similar to the working of any distance vector protocol.

In our implementation, we take advantage of the priority queues afforded by 802.11e to prioritize control packets over data packets. Moreover, the CDP scheduler assigns a fixed, low PHY rate (11 Mbps in our test-bed) for control packets. These design choices ensure high reliability and speedy delivery of control packets.

3) *Link Probability Estimation Protocol*: CDP computations given by (1) utilize link success probabilities  $p(i, j)$  for each pair of nodes  $i, j$ . We calculate these probabilities using an estimation protocol that consists of two phases: active probing and passive probing. In active probing, each node periodically broadcasts link PROBE packets. The link probability  $p(i, j)_{active}$  can be determined by counting the number of received probes in sequence. In passive probing we utilize the overhearing capability of the wireless medium. The receiver determines the link probability  $p(i, j)_{passive}$  based on missing sequence numbers in the data stream. The routing layer then uses a weighted average of the active and passive probe probabilities to estimate the link quality.

### B. Heuristic Improvements

Implementing CDP at the routing layer, while preserving the functionality of the remaining layers of the protocol stack, poses several challenges to the implementation of the algorithm. Most notable are the problems of looping and accounting for the 802.11's distributed coordination function (DCF) at the MAC layer. Below we describe these challenges and present our current heuristic-based approaches.

1) *Loop Avoidance*: CDP approximates the fixed-point solution to Equation 1 via a distributed distance vector approach. The classical problem of counting to infinity [11] in distance vector routing can affect CDP performance due to the time varying nature of the congestion metric and bursty pattern of arrivals. The problem is most acute when there is a sudden burst of traffic or the number of hops exceeds the time-to-live value due to looping. To address this issue, in our experiments we utilize Split-horizon with Poison Reverse [12] to avoid loops.

2) *Accounting for 802.11 DCF*: Equation 1 assumes a straightforward relationship between transmission delay of the head of the line packet (media access delay) and link probabilities ( $\frac{q(i)}{p(i, j)}$  approximates the total queuing time for  $q(i)$  packets). However, in 802.11 networks, the relationship between link probabilities, packet drops, and media-access delay is far more complicated. Our assumption often underestimates the per packet transmission delay when the link quality drops too low. Furthermore, due to the maximum retry count limit, in the face of persistent packet loss when low-quality links are selected, a large percentage of packets are dropped.

This all implies that the practical performance of CDP depends critically on the neighbor set criterion  $\gamma$ . In our experiments, we select  $\gamma$  empirically. For low values of  $\gamma$ , CDP allows for relaying even via those nodes likely to experience too large number of packet drops. This might result in an unexpected low packet delivery ratio. As  $\gamma$  increases, the average delay experienced by flows in the network decreases. At some point, however,  $\gamma$  becomes too high, in effect, partitioning the network and in some cases limiting the routes only to extremely reliable links.

## IV. THE EXPERIMENTAL SETUP AND MAIN RESULTS

We compare the performance of CDP against three state-of-the-art routing algorithms. Before we proceed, we describe these algorithms as well as our implementation of them.

- **Backpressure (BP)**: BP [10] selects the next hop based on a weighted differential backlog. Specifically, let  $q^d(i)$  denote the number of packets buffered in queue  $i$  destined for destination  $d$ . For any destination  $d$ , BP chooses the next hop  $K_{BP}^{(n, d)}$  at node  $n$ , such that

$$K_{BP}^{(n, d)} = \arg \min_k p(n, k)(q^d(k) - q^d(n)).$$

In our implementation, we serve packets in a FIFO manner and the transmission schedule is fully determined by 802.11.

- **SRCR [1]**: SRCR uses the ETX metric to route packets and does not take into account the queuing information at the nodes. Specifically, For a packet destined for  $d$ , the next hop  $K_{SRCR}^{(n, d)}$  is chosen such that

$$K_{SRCR}^{(n, d)} = \arg \min_k ETX^{(k, d)} + \frac{1}{p(n, k)},$$

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

- **E-Backpressure (E-BP) [8]**: E-BP is a variant of BP, where along with the queue information, the ETX metric is also used for path selection. In particular, for a packet destined for  $d$ , the next hop  $K_{EBP}(n, d)$  is chosen such that

$$K_{EBP}^{(n, d)} = \arg \min_k p(n, k)(q^f(k) - q^f(n)) + ETX^{(k, d)}.$$

To facilitate a direct comparison across approaches, we implement the BP algorithm using a structure similar to CDP in which  $\bar{V}_t^d(n)$  is replaced with  $q^d(i)$ . Also note that the calculation of ETX metric utilized by SRCR can be readily implemented by CDP when  $q_t(i)$  is replaced by constant 0 in (1). Naturally, the implementation of E-BP requires a careful combination of BP and SRCR.

Our test-bed consists of 12 wireless nodes with 512 MB of RAM with 500-MHz processors and Linux version 2.6.22. The nodes, whose placement is given in Figure 2, are distributed in about 215,000 sq. ft. of space. Each node is equipped with an Atheros-based 802.11a/b/g wireless interfaces (AR 5213) connected with omni-directional antennas. All nodes are configured in 802.11g ad-hoc mode with RTS/CTS disabled

and transmission power fixed to 13dBm. We use a low transmission rate of 11 Mbps for the control packets while the data packets are sent at 48 Mbps. The packet size for data packets used is 512 bytes. Based upon extensive experimentation, we have chosen the probability threshold  $\gamma$  to be 0.6. PROBE packets are transmitted at an interval of 1 second, while the control packets are broadcasted at an interval of 0.2 seconds. Link statistics are recalculated every 1 minute. Then, for each algorithm, each iteration of traffic generation is executed for another 1 minute.

We perform a comparative study of various routing protocols for the canonical example in Figure 1. We also present results from a more general scenario with two randomly selected flows, wherein the source and destination of each flow are chosen randomly among the nodes in our setup. In our comparative analysis we investigate the following performance measures:

- 1) End-to-end delay: For  $M$  packets, we define the average per packet delay as  $D = \frac{1}{M} \sum_{j=1}^M (\tau_A^j - \tau_D^j)$ , where  $\tau_A^j$  is the arrival time and  $\tau_D^j$  is the departure time for packet  $j$ . We also are interested in the distribution of the per packet delay, e.g. 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  $D_{cdp} - D_{candidate}$ , where  $D_{cdp}$  is the mean delay for CDP and  $D_{candidate}$  is the mean delay for the comparative protocol.
- 2) Delivery ratio: We plot the mean delivery ratio averaged over all flows. The delivery ratio is the ratio of the number of packets delivered at the destination to the packets sent from the source node, for the duration the experiment (undelivered packets are treated as lost packets). We use the normalized delivery ratio as a measure of performance, where the normalized delivery ratio is defined as the ratio between the delivery ratio of the candidate protocol versus that of CDP.

#### A. CDP Performance: Canonical Example

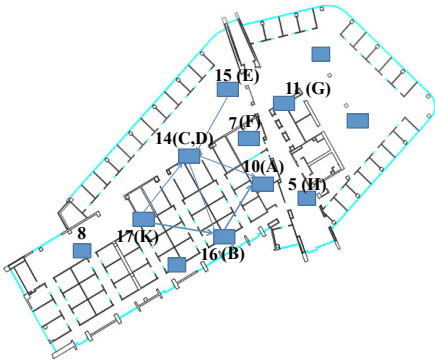


Fig. 2. Canonical Experiment (Nodes inside the bracket are mapped to corresponding nodes in Figure 1.)

Consider the canonical example of Figure 1 as marked in

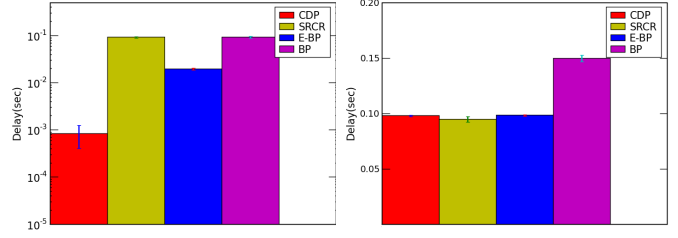


Fig. 3. Mean delay for 10-17 flow Fig. 4. Mean delay for 14-16 flow

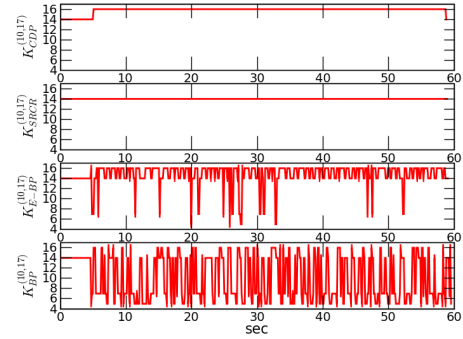


Fig. 5. Routing paths taken by node 10 for the flow 10-17

our test-bed. Poisson UDP traffic is generated at nodes 10 and 14 destined for nodes 17 and 16 respectively. Node 10 generates UDP packets at the rate of 1 Mbps, while node 14 is congested with UDP packets at a higher rate of 8 Mbps. Note that as it can be seen in Figure 2 nodes 10, 16, 14, 15, 7, 11, 5, 17 play roles similar to nodes A, B, D, E, F, G, H, and K respectively in Figure 1.

Figure 3 plots the average end-end delay incurred for the 10-17 UDP session under our four candidate protocols. As conjectured in Section II, the delay performance under CDP shows a significant improvement over the other candidate protocols (up to 50-fold improvement). Additionally, the packet drops due to buffer overflow and/or retry limit are fewer (not shown). Effectively, for 10-17 flow the number packets drops for SRCR, E-BP and BP are 60%, 20% and 40% respectively, while CDP packet loss is less than 1%, mostly due to buffer overflow.

Figure 4 plots the average end-end delay incurred for the 14-16 UDP session. We observe that CDP and E-BP have slightly higher delay compared to SRCR while BP has significantly higher delay for flow 10-17.

Figure 5 illustrates the next hop selections by nodes 10 i.e.  $K^{(10,17)}$  under the candidate protocols throughout the duration of the experiments. 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 forward significant number of packets into 5, 7 and 15 (shown in Figure 5) increasing the interference and packets drops as well as delay. The increased

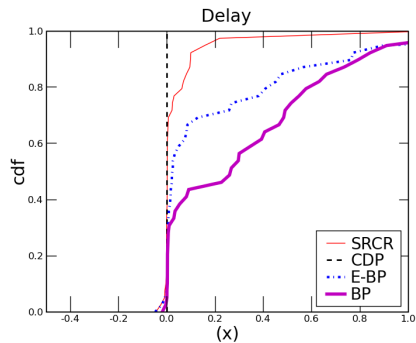


Fig. 6. CDF of delay differential for high load

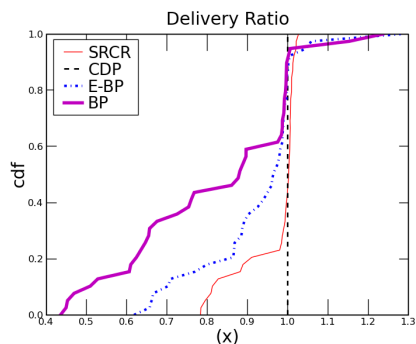


Fig. 7. CDF of delivery ratio for high load

delay for BP is attributed to the increased interference due to spreading of traffic for the flow 10-17.

We note that this is a pathological example carefully constructed to verify our initial intuition about the appropriate congestion measure. Next, we turn to a more general setup and evaluate the performance of CDP over many different topologies.

### B. Experiments with UDP Traffic

To give a flavor of the common case behavior, we also perform a comparative study of the routing protocols under another UDP setting. In particular, we study the gain achieved by CDP under high UDP load by selecting a configuration of two flows with randomly selected source and destination pairs. The traffic injected at each source node is Poisson and flow rate are randomly selected in a ratio 1:5 such that the total input rate at the sources is kept (high) at 10 Mbps. We repeat the experiment for 80 such configurations. We report the cumulative distribution function (CDF) of performance metrics for all protocols. Figure 6 compares the CDF of the delay differential. Figure 7 compares the CDF of the normalized delivery ratio of CDP, SRCR, BP and E-BP.

Figures 6 and 7 confirm the improved performance of CDF over other existing solutions under high load scenarios. In Figure 6, about 30% of the configurations for CDP perform better with respect to delay compared to the other protocols. In low load scenario, the performance of SRCR becomes compa-

parable to CDP (in networks with little congestion shortest path performs sufficiently well), while the performance loss for BP becomes even more significant.

## V. CONCLUSIONS AND DISCUSSIONS

This paper presents the Congestion Diversity Protocol (CDP) for routing packets across a wireless multi-hop network. CDP modifies the protocol stack at the routing layer to take into account the congestion in the network. In CDP, nodes route packets according to a rank ordering of the nodes based on a congestion measure which combines the important aspects of shortest path routing with those of backpressure routing.

We presented an initial evaluation of CDP on a real test-bed of 12 nodes with end-end delay and delivery ratio as central metrics of comparison. We compared the performance of CDP versus other routing-layer solutions, i.e. BP and SRCR under UDP connections. In special cases, we showed that CDP performed about 50 times better compared to other candidates. Furthermore, we showed significant improvements for broader set of UDP-based experiments, although insufficiency of space prevented us to discuss more evaluations necessary to draw general conclusions. In future, we would perform exhaustive experiments to determine precise guidelines where deploying CDP would be beneficial.

## REFERENCES

- [1] 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.
- [2] 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.
- [3] C. Lott and D. Teneketzis, "Stochastic routing in ad-hoc networks," *IEEE Transactions on Automatic Control*, vol. 51, pp. 52-72, January 2006.
- [4] 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.
- [5] 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.
- [6] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, "Trading structure for randomness in wireless opportunistic routing," in *ACM SIGCOMM*, Kyoto, Japan, August 2007.
- [7] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE Transactions on Automatic Control*, vol. 37, no. 12, pp. 1936-1949, August 1992.
- [8] M. J. Neely and R. Urgaonkar, "Optimal backpressure Routing for Wireless Networks with Multi-Receiver Diversity," *IEEE Transactions on Information Theory*, pp. 862-881, July 2009.
- [9] "Alix, <http://www.pcengines.ch/alix.htm>."
- [10] M. J. Neely, "Optimal Backpressure Routing for Wireless Networks with Multi-Receiver Diversity," in *Conference on Information Sciences and Systems (CISS)*, March 2006.
- [11] J.M. Jaffe and F. Moss, "A Responsive Distributed Routing Algorithm for Computer Networks," *IEEE Transactions on Communications*, vol. 30, July 1982.
- [12] Uyless Black, *IP routing protocols: RIP, OSPF, BGP, PNNI and Cisco routing protocols*, Prentice Hall PTR, Upper Saddle River, NJ, USA, 2000.