An Empirical Study of Performance Benefits of Network Coding in Multihop Wireless Networks

Dimitrios Koutsonikolas, Y. Charlie Hu, Chih-Chun Wang
School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907

***Abstract*—Recently, network coding has gained much popularity and several practical routing schemes have been proposed for wireless mesh networks that exploit interflow network coding for improved throughput. However, the evaluation of these protocols either assumed simple topologies and traffic patterns such as opposite flows along a single chain, or small, dense networks which have ample overhearing of each other's transmissions in addition to many overlapping flows. In this paper, we seek to answer the fundamental question: how much performance benefit from network coding can be expected for general traffic patterns in a moderate-sized wireless mesh network? We approach this question via an empirical study of both coordinated and opportunistic coding based protocols subject to general traffic patterns. Our study shows the performance benefits under both types of coding for general traffic patterns are extremely limited. We then analyze and uncover fundamental reasons for the limited performance benefits.**

I. INTRODUCTION

Recently, network coding has gained much popularity as a promising technique for improving throughput of routing protocols in wireless mesh networks. With network coding, a router can combine multiple packets within a single transmission, thus making more efficient usage of the network bandwidth. The basic principle of network coding can be easily explained through the 3-node "Alice and Bob" scenario from [1].

The work in [1] proposed COPE, a practical protocol that extends this basic principle to allow nodes to combine more than two packets together by exploiting the broadcast nature of the wireless medium through opportunistic listening. [1] showed a several-fold gain of COPE over a non-coding scheme on a wireless testbed and claimed that the maximum gain is unbounded. However, subsequent works [2], [3], [4], [5] identified practical limitations of COPE, and showed that its gain highly depends on the topology, traffic pattern or offered load.

More generally, these works tried to quantify the performance gain of network coding in multihop wireless networks, and can be grouped into two approaches. The first one (e.g., [5], [4]) studies theoretical upper bounds of coding gain, by constructing best-case coding scenarios. On one hand, the estimated upper bounds can be far from achievable for general topologies and traffic patterns. On the other hand, in deriving the upper bounds, it is often necessary to make assumptions about a particular coding structure, such as coding opportunities at a hotspot (e.g. [4]) that is crossed by many flows, ignoring the fact that nodes adjacent to the hotspot may also perform network coding which can lead to higher gain.

The second approach (e.g. [2], [3]) identifies that opportunistic coding such as in COPE may, in practice, miss several coding opportunities, depending on the order in which nodes in a neighborhood transmit packets. These works then propose the use of *coordinated* network coding, in which transmissions of neighboring nodes are scheduled with the goal of maximizing the gain from network coding. However, these works either provide only theoretical results, making unrealistic assumptions of a slotted MAC layer [2] or only consider a superficial traffic pattern, consisting of pairs of perfectly overlapping flows going towards opposite directions [3].

The above discussion suggests that quantifying the *average-case* (rather than providing upper bounds) throughput gain of routing protocols based on network coding under practical scenarios, i.e., in a general topology and under generic traffic patterns, remains an important open question. Quantifying the practical performance gain from network coding is of timely importance, as it not only helps to guide the design of high performance routing protocols, but also helps to justify the significant research efforts being invested by the community on exploring this new technique.

Since theoretically analyzing the average-case coding gain is extremely hard, in this paper, we perform an *empirical* study of practical coding gains by subjecting state-of-the-art coding schemes to a general topology and general traffic patterns. Our study shows the performance benefits of coding for general traffic patterns are extremely limited. We then analyze and uncover fundamental reasons for the limited performance benefits.

II. METHODOLOGY

We used Glomosim for our empirical evaluation study. Glomosim is a popular wireless network simulator with a detailed physical layer. We used the *2-ray* propagation model. We set the transmission range to 250m; the interference range, based on the physical model parameters used in Glomosim, was 460m. To facilitate network coding, we used the 802.11 broadcast MAC with a nominal link data rate of 2Mbps.

A. Topology and Traffic Pattern

Since analyzing the average performance of network coding under arbitrary topologies is hard, in this paper we focus on a generic, grid topology, one of the most natural candidate topologies in a planned deployment of mesh networks. We consider a square area of dimension 1.2Km × 1.2Km, and place 49 nodes in a grid topology. Based on the selected transmission and interference range, each node can directly communicate with its 4 neighbors along the left, right, up, and down directions, and is within the interference range of its 2-hop neighbors.

One particular feature of our study is that we consider a *sparse* node deployment, which provides only one next hop for each direction and no redundant node. This deployment does not provide any chance of opportunistic listening, and hence *at most 2* packets can be coded at any node. This choice of deployment is in contrast with previous studies that have assumed *dense* mesh networks (e.g., [6], [1]), and is chosen for two reasons. First, a recent study [7] showed that even low-rate control overhead in non-forwarding links can have a multiplicative throughput degradation on data-carrying links, and this effect worsens as the node density increases, which creates more non-forwarding links. Second, another recent study [8] showed that a node density slightly smaller than ours (about 30 nodes per Km^2) can provide 90% coverage in a currently operational mesh network. The same study argues that client-side solutions (e.g., higher-gain antennas) are much more cost-effective than denser topologies. Note that in our case, 100% coverage is guaranteed due to the grid placement.

Since in typical deployments of mesh networks all mesh nodes are expected to serve as access routers for some clients, it is expected every mesh node will originate some traffic. In our study, each node randomly selects a destination among the rest 48 nodes and initiates a constant rate flow with a packet size of 1500 bytes for a fixed duration of 1500 sec. We vary the sending rate from 10Kbps up to 2Mbps while keeping it the same for all 49 flows.[1]

B. Routing

We did not use any of the coding-aware routing schemes from the literature for several reasons: either they are centralized and difficult to realize in practice (e.g., [9]) or they are not deterministic and their control traffic may add variance to the performance of the protocols we examine, depending on the topology and traffic used (e.g., [10], [11]). In addition, we did not use any link quality metric; on a grid structure like the one we used, links are expected to have similar long term average link qualities, however short-term oscillations may cause frequent route changes, which can reduce the amount of existing coding opportunities and also makes the results of our study non-deterministic. Instead, we used fixed, shortest path routing in our study.

There are many possible shortest paths from a source to a destination in a grid (unless the source and the destination have one common coordinate). As we show in [14], dimension-ordered routing (DOR) [12], a well-known deterministic routing scheme for mesh and torus interconnection networks for parallel computers, gives more coding opportunities than a randomized shortest-path routing, which always randomly picks either x-axis or y-axis to make forward progress at each hop. In a two-dimensional grid, there are two ways of performing DOR for a given flow (source-destination pair). To maximize the coding opportunities of different flows, we propose a simple heuristic scheme, where each node decides which dimension to traverse first based on the relative orientation of the destination, as shown in Figure 1.

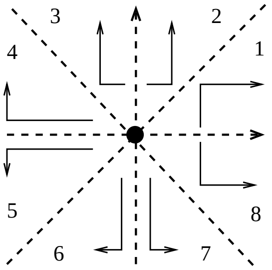


Fig. 1. Proposed path selection in DOR.

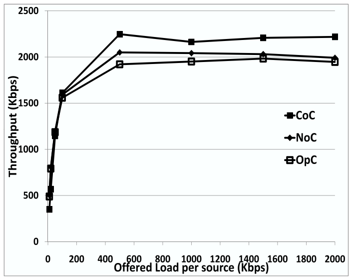


Fig. 2. Total throughput for CoC, OpC, and NoC in a 49-node grid.

C. Rate Control

A subtlety in evaluating coding based protocols is that the gain from network coding increases when the offered load in the network is increased, which creates more coding opportunities. However, letting the sources transmit at high data rates without any control can lead the network to congestion, and hence significantly reduced performance, as discussed in [13]. For this reason, we added rate control when comparing different protocols. We used CXCC [15], a hop-by-hop backpressure-based congestion control protocol for multihop wireless networks.

D. Network Coding

There have been two alternative approaches to developing practical interflow coding-based routing protocols, based on either ***Op****portunistic* ***C****oding* (OpC) or ***Co****ordinated* ***C****oding* (CoC). We study the performance benefits for both approaches in this paper, and compare them against the performance of CXCC, which we use as a representative protocol of ***No*** ***C****oding* (NoC).

1) OpC: COPE [1] is the first practical implementation of network coding. It performs *opportunistic network coding*, i.e., nodes mix (XOR) packets whenever they get an opportunity to do so, but they never delay packets waiting for coding opportunities to arise. To include rate control in OpC, we used a variation of COPE that is a simple integration of CXCC and COPE (also used in [3]).

2) CoC: noCoCo was proposed in [3] to perform *coordinated network coding* to ensure that nodes will not miss any coding opportunities. noCoCo extends the idea of CXCC for network coding by applying backpressure to two overlapping flows flowing in opposite directions. If a node sends out a coded packet, i.e., an XOR of two packets from two different flows, then from that point on, it is only allowed to transmit coded packets from those two flows.

noCoCo was designed to fully exploit coding opportunities in perfectly overlapping, bidirectional flows. In a network with many flows, there can be more than two flows that (partially) overlap with each other, and hence multiple ways to perform

---

[1] We have also experimented with different packet sizes, different sets of 49 flows, and different grid dimensions (network sizes). The results were qualitatively the same as and quantitatively very similar to those reported in the following sections and are omitted due to the space limitation.

pair-wise coding. However, noCoCo by itself does not have any mechanism for determining which flows should be coded together. To capture possible maximal coding opportunities, we use an offline centralized greedy heuristic to identify pairs of flows that should be coded together, by iteratively pairing up flows that have the maximum number of coding opportunities.

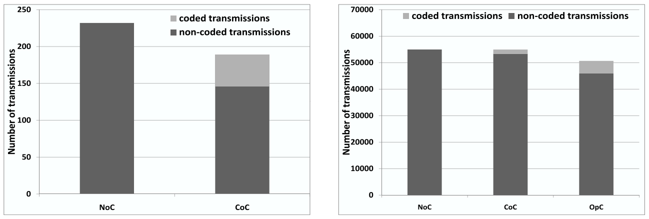
III. OVERALL RESULT

Figure 2 shows the total throughput achieved by the 49 flows in our simulation with NoC, OpC, and CoC, as the source sending rate varies. We make the following observations. First, we observe that the performance of all three schemes are similar. The gain of CoC over NoC after the convergence of throughput, i.e., after the source sending rate is higher than 500Kbps, is only 6%–11%. These numbers are much lower than the ones reported in [3], which considered the most ideal environment for network coding where only bidirectional flows were used. One may conjecture that, in a network with random flows, OpC can potentially outperform CoC, since it may identify more coding opportunities, without tying together specific pairs of flows. However, Figure 2 shows that the throughput with OpC is worse than with CoC. In fact, it is even lower than that with NoC (by about 2%–7%).

We contrast our results to those in previous studies. First, our result is different from the one in [3], in which the performance of OpC lies between the performance of NoC and that of CoC. Again, the reason is that bidirectional flows were used in [3] which offered more coding opportunities compared to in our settings. Second, our result is very different from the one reported in [1], where in a random topology and with a random traffic pattern, the original COPE protocol (OpC) offered a 3-4x throughput increase over NoC. This significant gain is due to the small (with a total of 20 nodes) and dense network used in [1], where flow lengths vary between 1 and 6 hops. Such a setting offered high overhearing probabilities, giving nodes the chance to XOR more than 2 packets together. In contrast, our evaluation was done in a larger and sparser network, with dimensions and settings closer to those we believe will be featured by future WMNs. In this network, overhearing does not add any benefit, and the flow lengths are much larger, varying from 1 to 12 hops.

IV. ANALYSIS OF CODING OPPORTUNITIES

The first step towards understanding the reasons for the low benefit of network coding for general traffic patterns in a sparse network deployment is to analyze the existence of coding opportunities. To measure the maximum achievable coding gain under general traffic patterns, i.e., each source initiates a flow to a random destination, we resort to a simple simulator that assumes an ideal MAC and physical layer without contention or packet losses. We used the simulator to estimate the normalized (with respect to the total packets transmitted) total number of transmissions required in order to deliver one packet from each source to its corresponding destination in the steady state with and without coding. We will call the result obtained with this simplified simulator *ideal*, in contrast to the *practical* result obtained with Glomosim.



(a) Result from the simplified simulator.

(b) Result from Glomosim – sources send packets at 2Mbps.

Fig. 3. Total number of non-coded and coded transmissions with NoC, CoC, and OpC for the 49-node scenario of Figure 2 with the simplified simulator and Glomosim.

Figure 3(a) shows the total number of transmissions without (NoC) and with (CoC) coding for the set of 49 flows used in Section III, obtained with this simplified simulator. We observe that the total number of transmissions required to deliver one packet from each source to its corresponding destination is reduced from 232 with NoC to 189 with CoC, a reduction of 22%, i.e., there are $232 - 189 = 43$ coded transmissions.

However, the reduction in the total number of transmissions from the realistic simulations is much lower. Figure 3(b) shows, compared to NoC, CoC and OpC reduce the total number of transmissions by only 0.5% and 7.8%, respectively. These small reductions in transmissions resulted in 11% higher throughput for CoC but 2% lower throughput for OpC, compared to NoC, as shown in Figure 2. Note that we cannot make any direct correlation between the reduction in the total transmissions and the total throughput improvement, since the 49 flows under consideration have different lengths; reduction in the number of transmissions only gives an indication of possible throughput improvement.

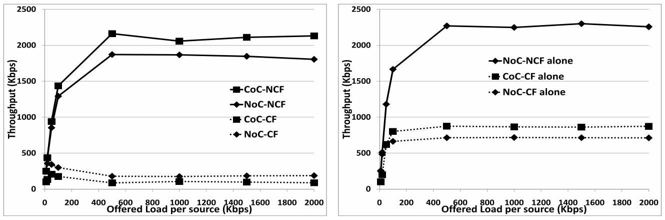
To understand why the number of coding opportunities captured in practice is even lower than the ideal number, we next analyze separately CoC and OpC in detail.

V. COORDINATED CODING

We begin with coordinated coding (CoC).

Separating codable and non-codable flows. We take a closer look at the 49 flows by breaking them into two classes. Flows that had their packets coded with packets from other flows at one node at least belong to class CF (codable flows). All the remaining flows, which had all their packets transmitted non-coded belong to class NCF (non-codable flows). Flows of class CF are the ones that contribute to the reduction of the total number of transmissions due to coding. With CoC, once a node decides to mix together two packets from two CF flows, it will mix together *all* the subsequent packets belonging to those two flows. We found that out of the 49 flows we initiated, only 24 belong to class CF, and the remaining 25 belong to class NCF. These numbers show again the low number of coding opportunities, since half of the flows could not be mixed with any other flow.

We repeated the experiment of Section III, however, this time we measured separately the total throughput of the 24 CF flows and the 25 NCF flows. The results are shown in



(a) Total throughput for the 25 NCF flows and the 24 CF flows with CoC and NoC when they coexist in a 49-node grid.

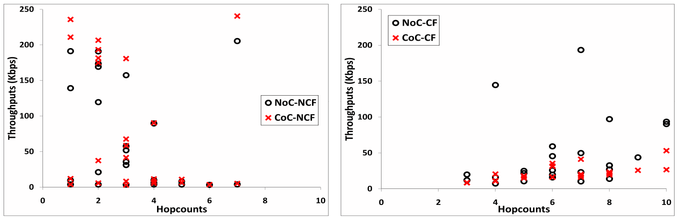
(b) Total throughput for the 25 NCF flows alone and the 24 CF flows alone with CoC and NoC in a 49-node grid.

Fig. 4. Throughput comparison of NCF and CF flows with CoC and NoC, when they coexist and when they are alone, in a 49-node grid.

Figure 4(a). We observe that the 25 NCF flows achieve much higher throughput than the 24 CF flows, both with NoC and with CoC. More surprisingly, the throughput of the 24 CF flows is less with CoC than with NoC. In other words, the use of network coding has a "squeezing effect" on the flows that actually have their packets mixed together, reducing the total throughput of those flows by 40% - 53% in the steady state (i.e., for offered load per source higher than 500Kbps), while the throughput of the remaining flows, whose packets are never mixed together, increases by 10% - 18%. In fact, these 25 NCF flows utilize almost the whole network capacity, since their total throughput is one order of magnitude larger than the throughput of the 24 CF flows. This explains why the total throughput with CoC is higher than with NoC in spite of the small number of coding opportunities, and in spite of the fact that the *relative* throughput decrease for the 24 CF flows is much larger than the relative throughput increase of the 25 NCF flows.

To verify that this counter-intuitive result is not due to some artifact in the protocol implementation, we repeated the simulation for CoC with only the 24 CF flows present in network. We also repeated the simulation for NoC twice, first only with the 24 CF flows, and then only with the 25 NCF flows. The results are shown in Figure 4(b). We observe that the results for the 24 CF flows are reversed, compared to Figure 4(a); when only the 24 CF flows are present in the network, the throughput with CoC is 21% higher than with NoC. This confirms that coding indeed helps when all flows are codable. It is the interaction of CF vs. NCF flows that causes the counter-intuitive result in Figure 4(a)! Note that all the results in [3] were obtained from settings similar to the one in Figure 4(b), with only CF flows present in the network.

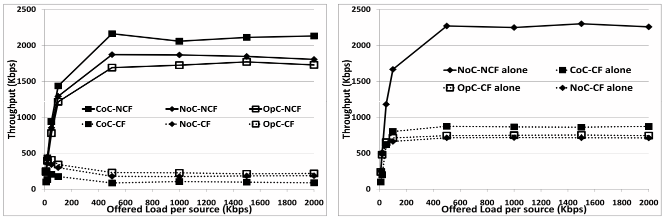
Correlating with flow lengths. In Figure 4(b), we also observe that the throughput of the 25 NCF flows, without the presence of the 24 CF flows, is more than 2.5 and 3.2 times higher than the throughput of the 24 CF flows with network coding and without network coding, respectively. To understand this, we correlated the individual throughputs of the 49 flows with their lengths (in terms of hopcounts).



(a) Hopcounts and throughputs for each of the 25 NCF flows with CoC and NoC.

(b) Hopcounts and throughputs for each of the 24 CF flows with CoC and NoC.

Fig. 5. Hopcounts and throughputs for non-coded and coded flows with CoC and NoC.



(a) Total throughput for the 25 NCF flows and the 24 CF flows with CoC, NoC, and OpC when they coexist in a 49-node grid.

(b) Total throughput for the 25 NCF flows alone and the 24 CF flows alone with CoC, NoC, and OpC in a 49-node grid.

Fig. 6. Throughput comparison of CF and NCF flows with CoC, NoC, and OpC when they coexist and when they are alone, in a 49-node grid.

Figures 5(a), 5(b) show the individual flow lengths and throughputs of the 25 NCF flows and the 24 CF flows, respectively, with CoC and NoC. We make three observations. (1) The 25 NCF flows are on average much shorter than the 24 CF flows. This is intuitive, since the longer the two flows, the higher the probability that they will overlap (at least partially), creating some coding opportunities; (2) This in turn explains why the throughputs for the NCF flows are in general much higher than for the CF flows; and (3) These individual flow throughputs are lower with CoC than with NoC for most of the CF flows, but higher for most of the NCF flows, which explains the throughput results in Figures 2 and 4(a).

VI. NON-COORDINATED CODING

We now return to OpC. In Figure 3(b), we saw that OpC had a larger fraction of coded over non-coded transmissions, compared to CoC. Yet it has the lowest total number of transmissions, and in Figure 2 the lowest total throughput, among the three schemes.

Separating CF and NCF flows. To explain this result, we first plot separately the throughput with OpC for the 25 NCF flows and the 24 CF flows, in Figure 6(a). We also keep the same breakdown for NoC and CoC from Figure 4(a) for easier comparison. Figure 6(a) shows that OpC exhibits exactly the opposite behavior compared to CoC. With OpC, the throughput for the 24 CF flows increases compared to NoC, while it decreases for the 25 NCF flows. Actually, the throughput with OpC for the 24 CF flows is the highest among the three schemes, and on the other hand, the throughput with OpC for the 25 NCF flows is the lowest among the three schemes. The most likely reason for this is that under OpC CF flows manage to capture more coding opportunities than

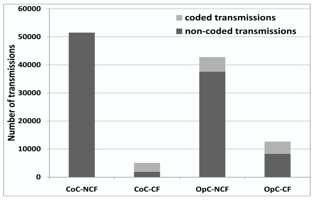


Fig. 7. Total number of non-coded and coded transmissions for the 25 NCF flows and the 24 CF flows with CoC and OpC with a source sending rate of 2Mbps.

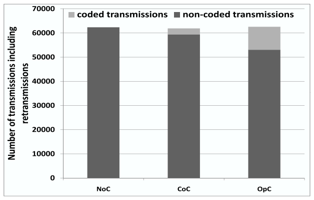


Fig. 8. Total number of non-coded and coded transmissions, including retransmissions, with NoC, CoC, and OpC, for the 49-node scenario of Figure 2.

under CoC, although the later guarantees none of the existing opportunities in CF flows are missed.

Does OpC indeed find more coding opportunities than CoC? To understand coding opportunities captured by OpC, we repeated the simulation with OpC for the 24 CF flows defined in Section V *alone*. The result is shown in Figure 6(b), in which we have also added the curves from Figure 4(b). for easier comparison. We observe the throughput of the 24 CF flows with OpC is only slightly better than with NoC (less than 5% difference), and lower than with CoC (10-15% difference). This suggests that in a network with only CF flows, which exhibit many coding opportunities, OpC actually misses many of them due to lack of coordination, resulting in lower performance than CoC.

Actually, since OpC applies only one-way backpressure to individual flows and it never ties two flows together, the classification of CF and NCF flows, which we carried over from Section V, has no meaning for OpC. In other words, OpC can potentially mix packets from CF flows with NCF flows. In light of this, in Figure 7 we break down the total number of transmissions with CoC and OpC, separately for the 24 CF and 25 NCF flows (according to the classification based on CoC), into coded and non-coded ones, with the source sending rate equal to 2Mbps. Each coded packet is counted twice, once for each flow.

We make two observations. (1) When we look at the 2nd and 4th bar, we observe that the percentage of coded over total transmissions for CF is larger with CoC than with OpC. This verifies again that OpC misses many coding opportunities for packets belonging to the CF flows. (2) The third bar shows that, with OpC, 12% of the data packets belonging to the 25 NCF flows were also sent out encoded. On the contrary, all the packets of these 25 flows were sent out as single packets with CoC, as the first bar shows. Hence we observe a tradeoff with OpC compared to CoC: OpC misses many coding opportunities among the CF flows due to lack of coordination, but, on the other hand, it captures more coding opportunities between random pairs of flows. This explains the initial observation, i.e., when all 49 flows are present, OpC captures in total more coding opportunities than CoC.

Why is OpC's throughput then even lower than NoC?

To understand the reason for this fundamental question, we plot again in Figure 8 the total number of transmissions with each scheme, and their breakdown into coded and non-coded ones, as in Figure 3(b), but this time we also count retransmissions. We observe that when retransmissions are also counted, OpC makes the largest number of transmissions, followed by NoC, and then by CoC. This implies that OpC is the most susceptible to packet failures among the three schemes, which forces it to make many retransmissions. This explains why OpC achieves the lowest throughput among the three schemes. We refer the reader to [14] for details.

VII. CONCLUSION

In this paper, we presented an empirical study of the *average-case* performance gain from adding network coding to routing protocols in wireless mesh networks. Our study shows the performance benefits under both coordinated and opportunistic coding for general traffic patterns are extremely limited. More importantly, we investigated and uncovered the fundamental reasons for the limited performance benefits: (1) There is an inherent limitation to the coding opportunities for general traffic patterns; (2) Codable flows are typically much longer than non-codable flows and hence suffer low throughput to begin with; (3) Under coordinated coding, the codable flows are squeezed by non-codable flows and hence suffer reduced flow rate, i.e., reduced performance benefit from coding; (4) While opportunistic coding appears to capture more coding opportunities, it is also more susceptible to packet loss than coordinated coding and no coding, and this susceptibility can result in lower throughput even compared to no coding.

ACKNOWLEDGMENT

This work was supported in part by NSF grant CNS-0626703.

REFERENCES

[1] S. Katti, et al., "XORs in the air: Practical wireless network coding", in *Proc. of ACM SIGCOMM*, August 2006.
[2] P. Chaporkar and A. Proutiere, "Adaptive network coding and scheduling for maximizing throughput in wireless networks", in *Proc. of ACM MOBICOM*, 2007.
[3] B. Scheuermann, et al., "Near-optimal coordinated coding in wireless multihop networks", in *Proc. of ACM CoNEXT*, 2007.
[4] J. Le, et al., "How many packets can we encode? – an analysis of practical wireless network coding", in *Proc. of IEEE INFOCOM*, 2008.
[5] J. Liu, et al., "Bounds of the gain of network coding and broadcasting in wireless networks" in *Proc. of IEEE INFOCOM*, 2007.
[6] J. Bicket, et al., "Architecture and evaluation of an unplanned 802.11b mesh network", in *Proc. of ACM MOBICOM*, 2005.
[7] J. Camp, et al., "A measurement study of multiplicative overhead effects in wireless networks," in *Proc. of IEEE INFOCOM*, 2008.
[8] J. Robinson, et al., "Assessment of urban-scale wireless networks with a small number of measurements", in *Proc. of ACM MOBICOM*, 2008.
[9] S. Sengupta, et al., "An analysis of wireless network coding for unicast sessions: The case for coding-aware routing", in *Proc. of IEEE INFOCOM*, 2007.
[10] J. Le, et al., "DCAR: Distributed coding-aware routing in wireless networks", in *Proc. of IEEE ICDCS*, 2008.
[11] S. M. Das, et al. "Context based routing: Technique, applications and experience", in *Proc. of USENIX NSDI*, April 2008.
[12] W. Dally and C. Seitz, "Deadlock-free message routing in multiprocessor interconnection networks", *IEEE Trans. on Computers*, vol. 36, no. 5, pp. 547–553, May 1987.
[13] D. Koutsonikolas, et al., "How to evaluate exotic wireless routing protocols?", in *Proc. of ACM HotNets-VII*, 2008.
[14] D. Koutsonikolas, et al., "An Empirical Study of Performance Benefits of Network Coding in Multihop Wireless Networks", TR-ECE-08-11, Purdue University, December 2008.
[15] B. Scheuermann, et al., "Implicit hop-by-hop congestion control in wireless multihop networks", *Elsevier Ad Hoc Networks*, vol. 6, no. 2, pp. 260–286, Apr. 2008.