

PERFORMANCE OF NETWORK CODING IN AD HOC NETWORKS

Joon-Sang Park[†], Desmond S. Lun[‡], Fabio Soldo[§] Mario Gerla[†], and Muriel Médard[¶]

[†] Computer Science Department, University of California, Los Angeles

[‡] Coordinated Science Laboratory, University of Illinois at Urbana-Champaign

[§] Dipartimento di Matematica, Politecnico di Torino

[¶] Laboratory for Information and Decision Systems, Massachusetts Institute of Technology

E-mail: {jspark,gerla}@cs.ucla.edu, dslun@uiuc.edu, fabio.soldo@studenti.polito.it, medard@mit.edu

Abstract—Network coding, the notion of performing coding operations on the contents of packets while in transit through the network, was originally developed for wired networks; recently, however, it has been also applied with success also to wireless ad hoc networks. In fact, it has been shown that network coding can yield substantial performance gains, e.g., reduced energy consumption, in ad hoc networks. In this paper, we compare, using linear programming formulations, the maximum throughput that a multicast application can achieve with and without network coding in unreliable ad hoc networks; we show that network coding achieves 65% higher throughput than conventional multicast in a typical ad hoc network scenario. The superiority of network coding, already established by the analytic results, is confirmed by simulation experiments.

I. INTRODUCTION

Recent results on the advantages of network coding in wired networks have stimulated a lot of interest in the subject and in particular, in the application of network coding to wireless ad hoc networks. Network coding refers to the basic notion of performing coding operations on the contents of packets throughout a network, and is generally attributed to Ahlswede et al. [1], who showed the utility of the network coding for multicast in wired networks. The work of Ahlswede et al. was followed by other work by Koetter and Médard [5] who showed that codes with a simple, linear structure were sufficient to achieve the capacity of multicast connections in lossless, wireline networks. This result was augmented by Ho et al. [3], who showed that, in fact, a random construction of the linear codes was

This work was supported in part by the National Science Foundation under Grant No. 0520332 and the US Army under MURI award W911NF-05-1-0246. The research is continuing through participation in the International Technology Alliance sponsored by the U.S. Army Research Laboratory and the U.K. Ministry of Defence under Agreement Number W911NF-06-3-0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the National Science Foundation, the US Army Research Laboratory, the U.S. Government, the UK Ministry of Defense, or the UK Government. The US and UK Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

sufficient. The utility of such random linear codes for reliable communication over lossy packet networks—such as wireless ad hoc networks—was soon realized [7]. In [8], a prescription for the efficient operation of ad hoc networks is given, which proposes using the random linear coding scheme of [7] coupled with optimization methods for selecting the times and locations for injecting coded packets into the network.

In this paper, we consider the problem of identifying the maximum end-to-end throughput that a multicast connection can achieve with network coding given an unreliable ad hoc network. We give two mathematical optimization formulations for maximum throughput multicast: one with network coding and one without; then, we compare the maximum throughput that network coding achieves to the maximum throughput that conventional multicast achieves in an example network topology. We develop formulations based on the convex programming formulation of the minimum cost multicast problem for network coding given in [8]. In contrast to the minimum cost multicast problems considered in [8], modeling the wireless medium contention constraints is crucial in the maximum throughput problem that we consider. The mathematical programming formulations of the maximum throughput multicast problem presented in this paper includes the wireless medium contention constraints. We use a technique similar to those proposed in [4], [11], [13] to model such constraints.

The rest of this paper organized as follows. In Section II, we develop mathematical formulations for maximum throughput multicast in unreliable ad hoc networks with and without network coding; in Section III, we compare the performance of network coding and conventional multicast in ad hoc networks via numerical analysis and simulation; finally, in Section IV, we conclude this paper with a brief summary.

II. MODELING MAXIMUM MULTICAST THROUGHPUT PROBLEM IN AD HOC NETWORKS

In this section, we formulate the maximum multicast throughput problem in ad hoc networks with and without network coding. First, we develop an optimization formulation for the case with network coding based on the model for the minimum cost multicast in ad hoc networks presented in [8]. Specifically, we change the model to maximize throughput instead of minimizing cost and include wireless medium contention/scheduling constraints.

As suggested in [8], we represent the network with a directed hypergraph $\hat{\mathcal{H}} = (\mathcal{N}, \hat{\mathcal{A}})$, where \mathcal{N} is the set of nodes and $\hat{\mathcal{A}}$ is the set of hyperarcs. Each hyperarc (i, J) represents a broadcast link from node i to nodes in the non-empty set $J \subset \mathcal{N}$. The hyperarc (i, J) indicates that a packet transmitted by node i can be received by all nodes in J (and no nodes in $\mathcal{N} \setminus J$). For example, if packets transmitted by node 1 can be received only by node 2, 3, and 4, then $(1, \{2\ 3\ 4\}) \in \hat{\mathcal{A}}$. We assume that there is no rate and/or transmission/reception (tx/rx) range adaptation, i.e., all the broadcast links operate at the same data rate and have the same tx/rx range and thus every node $i \in \mathcal{N}$ has at most one hyperarc $(i, J) \in \hat{\mathcal{A}}$. In fact, most ad hoc multicast routing protocols assume fixed rate broadcast links due to the lack of practical rate adjustment schemes suitable for broadcasting. Now let $\mathcal{H} = (\mathcal{N}, \mathcal{A})$ where $\mathcal{A} = \{(i, K) | (i, J) \in \hat{\mathcal{A}}, K \subseteq J, K \neq \emptyset\}$. For examples, if $(1, \{2\ 3\ 4\}) \in \hat{\mathcal{A}}$ then $(1, \{2\})$, $(1, \{3\})$, $(1, \{4\})$, $(1, \{2\ 3\})$, $(1, \{3\ 4\})$, $(1, \{2\ 4\})$, $(1, \{2\ 3\ 4\}) \in \mathcal{A}$. Each hyperarc in \mathcal{A} represents a specific usage of a broadcast link in $\hat{\mathcal{A}}$. The hyperarc $(1, \{2\ 3\ 4\}) \in \hat{\mathcal{A}}$ can be used as $(1, \{2\ 3\})$ to realize a packet transmission from node 1 to nodes 2 and 3 only. In general, it is not harmful if the packet reaches non-intended destinations, node 4 in this case. Describing the exact usage of hyperarcs is needed to take interference among hyperarcs into account. Hereafter, we use the term hyperarc to indicate an element in \mathcal{A} .

Under unreliable wireless channel, packets can be lost on their way to destinations when they are injected into the links (or hyperarcs). In case of a packet injected into a hyperarc with multiple destinations, the packet may reach only a subset of destinations. Let z_{iJ} ($\leq L$, the link capacity) be the average rate at which packets are injected into hyperarc (i, J) and let z_{iJK} be the average rate at which packets are received by all nodes in K (and no nodes in $\mathcal{N} \setminus K$) given z_{iJ} . Assuming a node $j \in J$ receives a packet injected into a hyperarc (i, J) with probability p and independence of the packet reception events, we have $z_{iJK} = z_{iJ} p^{|K|} (1-p)^{(|J|-|K|)}$. More precisely, let $A_{iJK}(\tau)$ for $\tau \geq 0$ be the total number of packets that are injected on hyperarc (i, J) and received by all nodes in K (and no nodes in $\mathcal{N} \setminus K$) between

time 0 and time τ then we assume almost surely that $\lim_{\tau \rightarrow \infty} \frac{A_{iJK}(\tau)}{\tau} = z_{iJK}$ and $z_{iJ} = \sum_{K \subset J} z_{iJK}$. The rate vector \mathbf{z} consisting of z_{iJ} 's is the coding subgraph (indicating the times and locations for injecting coded packets into the network) for the multicast connection of interest and can be varied within a convex, constrained set $Z \subset [0, L]^{|\mathcal{A}|}$.

We assume that the network is time-slotted. That is, time is divided into slots and packet transmissions are restricted to start at slot boundaries. Some hyperarcs interfere each other such that in a specific time slot only a certain set of hyperarcs can be activated simultaneously. Consider the following set of hyperarcs: $(1, \{2\})$, $(1, \{3\})$, $(1, \{2\ 3\})$, $(4, \{3\})$, $(4, \{5\})$, $(4, \{6\})$, $(4, \{3\ 5\})$, $(4, \{3\ 6\})$, $(4, \{5\ 6\})$, and $(4, \{3\ 5\ 6\})$. Node 1 and node 4 cannot transmit packets simultaneously using hyperarcs $(1, \{2\ 3\})$ and $(4, \{3\ 5\ 6\})$ since a collision occurs at node 3; however, simultaneous packet transmissions can be scheduled if two *non-interfering* hyperarcs $(1, \{2\})$ and $(4, \{5\ 6\})$ are used. Formally, we define two hyperarcs (i, J) and (i', J') to be non-interfering if and only if the three following condition are satisfied:

- $i \neq i'$
- $\nexists (i', K) \in \mathcal{A}$ s.t. $K \cap J \neq \emptyset$
- $\nexists (i, K) \in \mathcal{A}$ s.t. $K \cap J' \neq \emptyset$

The second and third condition state that if there is any hyperarc connecting any $j \in J$ and i' or any hyperarc connecting any node in J' and i , a collision occurs. Note that a hyperarc (i, J) in \mathcal{A} denotes a specific usage of a broadcast link in $\hat{\mathcal{A}}$ and thus a packet transmission using (i, J) may reach other nodes in $\mathcal{N} \setminus J$.

Based on the time slotted network assumption, we model the wireless medium contention as the following *scheduling* constraints:

$$\begin{aligned} \sum_k \lambda_k c_k(i, J) - z_{iJ} &\geq 0, \quad \forall (i, J) \in \mathcal{A}, \\ \sum_k \lambda_k &\leq 1 \end{aligned} \tag{1}$$

where

$$c_k(i, J) = \begin{cases} L & \text{if } (i, J) \in \mathcal{A}_k, \\ 0 & \text{otherwise.} \end{cases}$$

We enumerate all the possible maximal sets of non-interfering hyperarcs as $\mathcal{A}_k (\subset \mathcal{A})$ where $k = 1, \dots, M$ (M varies as the network topology changes and by maximal sets we mean $\mathcal{A}_i \not\subseteq \mathcal{A}_j$ for all $i, j \in \{1 \dots M\}$, $i \neq j$) and $\lambda_k \in [0, 1]$ denotes the fraction of time allocated to \mathcal{A}_k . These constraints are used to impose a feasible scheduling over the network. Any solution satisfying these constraints is always feasible and the

corresponding feasible link schedule is having every hyperarc set \mathcal{A}_k exclusively active for λ_k fraction of time. In fact, it is also easy to see that the converse holds, i.e., if a solution is feasible then it must satisfy inequalities (1). This makes the constraints a sufficient and necessary condition for feasible scheduling. This modeling approach is similar to those discussed in [4], [11], [13] and is basically the same as the one in [4] except that we use the notion of hyperarc. In [4], the network is modeled as a graph (as opposed to a hypergraph) since neither network coding nor multicast is considered.

Now suppose we have a source node $s \in N$ transmitting data to a non-empty set of terminal nodes T . The maximum throughput multicast with network coding can be formulated as follows:

$$\text{maximize } f$$

subject to

$$\begin{aligned} \sum_k \lambda_k c_k(i, J) - z_{iJ} &\geq 0, \quad \forall (i, J) \in \mathcal{A}, \\ \sum_k \lambda_k &\leq 1, \end{aligned}$$

$$\begin{aligned} \sum_{\{L \subset J | L \cap K \neq \emptyset\}} z_{iJL} - \sum_{j \in K} x_{iJj}^{(t)} &\geq 0, \\ \forall (i, J) \in \mathcal{A}, K \subset J, t \in T, \end{aligned}$$

$$\begin{aligned} \sum_{\{J | (i, J) \in \mathcal{A}\}} \sum_{j \in J} x_{iJj}^{(t)} - \sum_{\{j | (j, I) \in \mathcal{A}, i \in I\}} x_{jIi}^{(t)} \\ = \begin{cases} f & \text{if } i = s, \\ -f & \text{if } i = t, \\ 0 & \text{otherwise,} \end{cases} \quad \forall i \in \mathcal{N}, t \in T, \end{aligned}$$

$$\begin{aligned} x_{iJj}^{(t)} &\geq 0, \quad \forall (i, J) \in \mathcal{A}, j \in J, t \in T, \\ z_{iJ} &\geq 0, \quad \forall (i, J) \in \mathcal{A}, \\ \lambda_k &\geq 0, \quad \forall k \end{aligned}$$

where

$$c_k(i, J) = \begin{cases} L & \text{if } (i, J) \in \mathcal{A}_k, \\ 0 & \text{otherwise.} \end{cases}$$

The quantity $x_{iJj}^{(t)}$ represents the amount of data flow transmitted from node i to j with respect to destination t using hyperarc (i, J) . The solution produced by this linear program is always feasible and asymptotically exact. That is, given a network $\hat{\mathcal{H}} = (\mathcal{N}, \hat{\mathcal{A}})$ the multicast connection (s, T) can achieve the maximum throughput f^* arbitrarily closely and this is feasible when all the hyperarcs belong to \mathcal{A}_k simultaneously activated for λ_k

fraction of time, which will result in the optimal rate vector z^* . Given z^* , how to encode packets on each node to achieve f^* is a separate problem and we can use a scheme described in [7], namely random network coding.

The maximum throughput multicast without network coding in wired networks can be achieved using the tree packing strategy (e.g., [12]), i.e., constructing multiple multicast trees (or subgraphs) each of which carries an independent flow such that the aggregated flow is maximized. Thus, we formulate the maximum throughput multicast problem in an ad hoc network not performing network coding as the tree packing problem. Let us first consider the following program ([2], modified for consistency):

$$\text{minimize } \sum x_{ij}$$

subject to

$$\begin{aligned} \sum_{\{j | (i, j) \in \mathcal{A}'\}} x_{ij}^{(t)} - \sum_{\{j | (j, i) \in \mathcal{A}'\}} x_{ji}^{(t)} \\ = \begin{cases} 1 & \text{if } i = s, \\ -1 & \text{if } i = t, \\ 0 & \text{otherwise,} \end{cases} \quad \forall i \in \mathcal{N}, t \in T, \end{aligned} \quad (2)$$

$$\begin{aligned} x_{ij} &\geq x_{ij}^{(t)} \quad \forall (i, j) \in \mathcal{A}', t \in T, \\ x_{ij}^{(t)} &\in \{0, 1\} \quad \forall (i, j) \in \mathcal{A}', t \in T \end{aligned}$$

where $\mathcal{A}' := \{(i, j) | (i, J) \in \mathcal{A}, J \ni j\}$.

The above program solves the single multicast tree construction problem given an ad hoc network $\mathcal{H} = (\mathcal{N}, \mathcal{A})$ and the source and the destination set (s, T) . Owing to the integrality constraint the program finds a multicast tree where only one path from the source to each destination exists. Now we extend this idea to formulate the tree packing problem in ad hoc networks as a mixed integer linear program. Consider the following program:

$$\text{maximize } \sum_c f^{(c)}$$

subject to

$$\sum_k \lambda_k c_k(i, J) - \sum_c z_{iJ}^{(c)} \geq 0, \quad \forall (i, J) \in \mathcal{A}, \quad (3)$$

$$\sum_k \lambda_k \leq 1, \quad (4)$$

$$\begin{aligned} \sum_{\{L \subset J | L \cap K \neq \emptyset\}} z_{iJL}^{(c)} - \sum_{j \in K} x_{iJj}^{(t, c)} &\geq 0, \\ \forall (i, J) \in \mathcal{A}, K \subset J, t \in T, c, \end{aligned} \quad (5)$$

$$\sum_{\{J|(i,J) \in \mathcal{A}\}} \sum_{j \in J} x_{ij}^{(t,c)} - \sum_{\{j|(j,I) \in \mathcal{A}, i \in I\}} x_{ji}^{(t,c)} = \begin{cases} f^{(c)} & \text{if } i = s, \\ -f^{(c)} & \text{if } i = t, \\ 0 & \text{otherwise,} \end{cases} \quad \forall i \in \mathcal{N}, t \in T, c, \quad (6)$$

$$x_{ij}^{(t,c)} = \sum_{\{J \ni j | (i,J) \in \mathcal{A}\}} x_{iJ}^{(t,c)}, \quad \forall (i,j) \in \mathcal{A}', t \in T, c, \quad (7)$$

$$\sum_{\{j|(i,j) \in \mathcal{A}'\}} n_{ij}^{(t,c)} \leq 1, \quad \forall i \in \mathcal{N}, t \in T, c, \quad (8)$$

$$\sum_{\{i|(i,j) \in \mathcal{A}'\}} n_{ij}^{(t,c)} \leq 1, \quad \forall j \in \mathcal{N}, t \in T, c, \quad (9)$$

$$x_{ij}^{(t,c)} \leq n_{ij}^{(t,c)}, \quad \forall (i,j) \in \mathcal{A}', t \in T, c, \quad (10)$$

$$Mx_{ij}^{(t,c)} \geq n_{ij}^{(t,c)}, \quad \forall (i,j) \in \mathcal{A}', t \in T, c, \quad (11)$$

$$n_{ij}^{(t,c)} \in \{0, 1\} \quad \forall (i,j) \in \mathcal{A}', t \in T, c, \quad (12)$$

$$x_{iJ}^{(t,c)} \geq 0, \quad \forall (i,J) \in \mathcal{A}, j \in J, t \in T, c, \quad (13)$$

$$z_{iJ}^{(c)} \geq 0, \quad \forall (i,J) \in \mathcal{A}, c, \quad (14)$$

$$\lambda_k \geq 0, \quad \forall k \quad (15)$$

$$\quad (16)$$

where $\mathcal{A}' := \{(i,j) | (i,J) \in \mathcal{A}, J \ni j\}$,

$$c_k(i,J) = \begin{cases} L & \text{if } (i,J) \in \mathcal{A}_k, \\ 0 & \text{otherwise,} \end{cases}$$

and M is a sufficiently large number.

This program finds the maximum achievable multicast throughput under the tree packing strategy given $(\mathcal{N}, \mathcal{A})$ and (s, T) . We attempt to build C multicast subgraphs, each of which connects the source s to the destination set T and admits an independent data flow, i.e., no mixing is allowed between different flows. The amount of independent data flow through each structure is represented as $f^{(c)}$ where $c = 1, \dots, C$ and $C = \max_{(i,J) \in \mathcal{A}} |J|$ is sufficient. The rate vector $\mathbf{z}^{(c)}$ consisting of $z_{iJ}^{(c)}$'s represents a specific multicast subgraph c (by indicating the times and locations that the multicast subgraph is injecting packets into the network). In detail, inequalities (3) and (4) impose that all distribution subgraphs share the wireless medium, i.e., the fraction of time assigned to hyperarc (i,J) must be now shared among every subgraph using that hyperarc. The variable $x_{ij}^{(t,c)}$ depicts the amount of flow from i to j with respect to destination t on subgraph c and the binary variable $n_{ij}^{(t,c)}$ is linked to variable $x_{ij}^{(t,c)}$ through constraints (10) and (11). The idea here is to construct tree-like structures as we did in the previous program (2). Since $n_{ij}^{(t,c)}$ equals to 1 if and only if there is data flow from i to j using any eligible hyperarc, (8) and (9) allow no split (or allow no gain

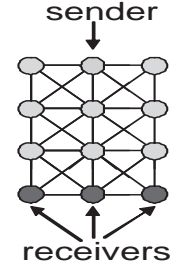


Fig. 1. 4x3 grid network topology graph where each edge represents a unit-capacity link. (The sender is located in the top row, middle column and three receivers are in the bottom row.)

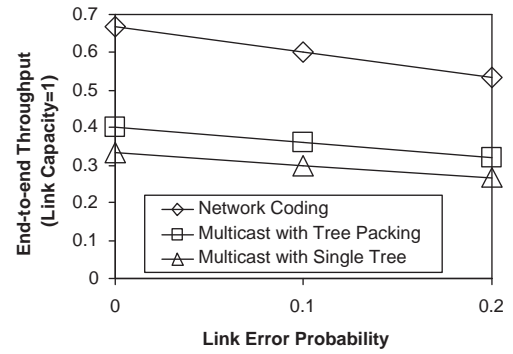


Fig. 2. Maximum Throughput in 4x3 grid network

from split) of data flow from source s to destination t in a multicast subgraph.

In this formulation for multicast without network coding, we assume hop-by-hop error correction (by retransmission or hop-by-hop erasure coding). For each packet, nodes transmit multiple packets to ensure that the packet is communicated despite random packet drops caused by channel error. When the error probability is p , for example, it is assumed that the sender transmits $1/p$ packets to communicate a single packet to the receiver. Retransmissions and erasure coding consume additional channel bandwidth and thus the maximum throughput of a multicast connection with unreliable channels will be less than that with reliable channels. In the case of retransmissions, we assume free feedback, i.e., the sender knows (magically with no cost) whether the receiver has received the transmitted packet correctly or not. In the network coding case, error correction is automatically provided by the network code and no additional mechanism for hop-by-hop error correction is required. We refer readers to [8] for details.

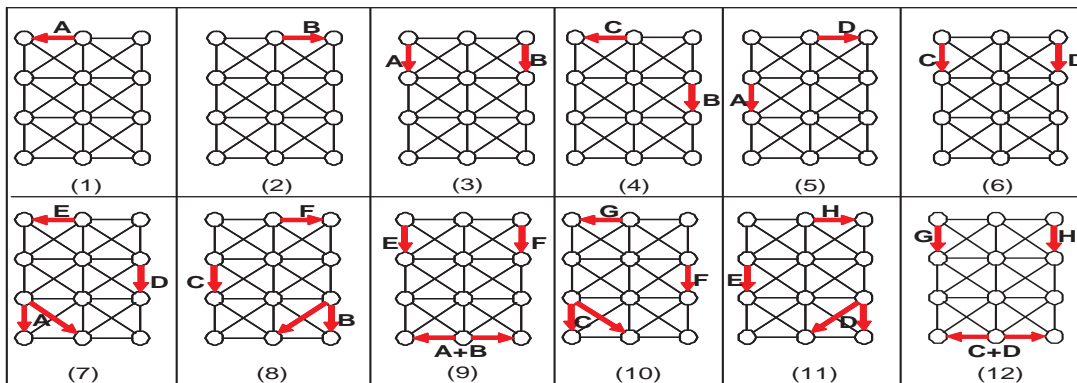


Fig. 3. Coding example achieving multicast throughput of $2/3$ in 4×3 grid network with unit link capacity. (Three receivers in the bottom row receive 2 packets every 3 slots after 6^{th} slot.)

III. PERFORMANCE COMPARISON OF NETWORK CODING AND CONVENTIONAL MULTICAST IN AD HOC NETWORKS

By solving the two optimization formulations using the same input we can compare the maximum multicast throughput with and without network coding for the given input. The results obtained by solving the programs using the 4×3 grid network illustrated in Figure 1 as the input are shown in Figure 2. We assume that in the network every link has unit capacity. We can see in the results that network coding achieves a maximum throughput of $2/3$, which is 65% higher than a maximum throughput of conventional multicast, which is $2/5$ when there is no error. A link schedule and coding example achieving $2/3$ throughput is presented in Figure 3. In conventional multicast as expected, the throughput is higher in the case when tree packing strategy is used (denoted as “Multicast with Tree Packing” in the figure) than the case with a single distribution tree (denoted as “Multicast with Single Tree”). The throughput decreases linearly in the link error probability for both network coding and conventional multicast case. This linear decrease of throughput in the link error probability is achieved under the assumption of a hypothetical error correction scheme. Network coding achieves the same results without the assumption.

The above comparison between network coding and conventional multicast is based on several assumptions which are not easy to replicate in real life systems: time-slotted scheduled multiple access, the existence of “genie” that can tell the sender whether the multicast receivers received the packet correctly or not in conventional multicast case. Moreover, nodes are assumed to be static. To get a more realistic assessment, we compare, via simulation, a realistic implementation of network coding, namely CodeCast [9] with a realistic implemen-

tation of conventional multicast, namely ODMRP [6]. These results were actually already reported in an earlier paper [9]. They are repeated here to contrast them with the linear programming comparison and to confirm the superiority of network coding even in realistic scenarios.

The most important difference between linear programming solutions and simulation is that in the latter there is no time-slotted and scheduled access to wireless channel, nor a “genie” that retransmits packets. Thus, in the multicast implementation packets are lost. The proper measure for comparison is then the delivery ratio (of network coding and multicast respectively) under the same input rate.

Using [10] we conducted a set of experiment with the following settings: 802.11 DCF MAC; two-ray ground path-loss propagation model; 376m of transmission range and 2Mbits/sec of bandwidth; 100 nodes randomly placed on $1500 \times 1500 m^2$ field; single multicast group with single source and 10 receivers; constant bit-rate, 5Kbytes/sec, application using fixed 512bytes packet size; Random Waypoint Mobility model with 0 pause time, 0 minimum speed, and varying maximum speed.

For CodeCast, two different block sizes are used to evaluate the impact of the block size on the performance. In Figure 4, CodeCast- α -dp β denotes CodeCast using α -packet blocks and operating in the artificial lossy channel with packet drop probability $\beta\%$. (Nodes are forced to drop successfully received packets randomly with a certain probability.) CodeCast-8-dp0 indicates 8-packet block and packet drop probability 0 case, CodeCast-8-dp10 does 8-packet block and packet drop probability 10% case, and CodeCast-4-dp0 is CodeCast for 4-packet block and packet drop probability 0. Similarly, ODMRP-dp β denotes ODMRP for the packet drop probability $\beta\%$ case.

In Fig.4(a), CodeCast demonstrates near 100% data delivery regardless of mobility speed, block size, packet

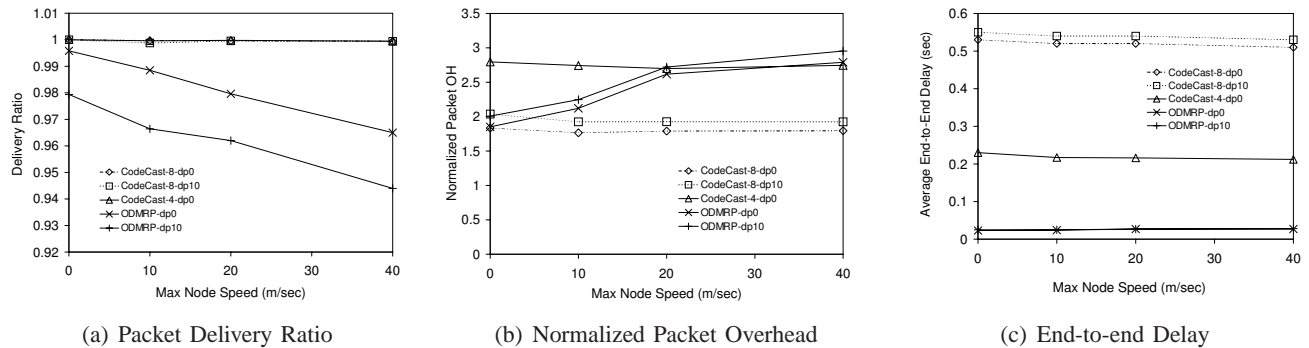


Fig. 4. CodeCast vs. ODMRP

drop probability. On the other hand, the packet delivery ratio of the conventional multicasting represented by ODMRP degrades from 99% to 94% as mobility and packet drop probability increase. The packet delivery ratio is defined as the ratio of data packets received by all receivers over total data packets sent. More importantly, as shown in 4(b), CodeCast incurs less overhead than ODMRP (if the block size is 8 packets). When the maximum node speed is 40 m/sec the reduction in overhead is as much as 40%. To measure protocol overhead, we use a common metric, the normalized packet overhead, defined as the number of any packets transmitted to the broadcast channel divided by the total number of data packets delivered to any receiver. The overhead of CodeCast with 4-packet block is comparable to that of ODMRP when mobility is high. ODMRP is designed to use more and more nodes as forwarding nodes when mobility increases, which is equivalent to trading off overhead for high packet delivery ratio to cope with mobility. 4(c) exposes one of the weak points of CodeCast, namely end-to-end delay. The end-to-end delay is the difference between packet generation time at the source and packet delivery to the receiver. In CodeCast, a certain level of increase in end-to-end delay is inevitable since the source must take time to collect a block of packets such that coding over that block is possible. In our simulations, the application generates packets at a rate of 10 packets/sec so on average each packet spends 0.35 seconds waiting in the buffer at the source if the block size is 8 packets. It spends 0.15 seconds if the block size is 4 packets. This explains in part why CodeCast-4 has lower average end-to-end delay than CodeCast-8s.

IV. CONCLUSIONS

In this paper, we discussed the problem of identifying the maximum throughput that a multicast connection with or without network coding can achieve in an unreliable static ad hoc network. We presented mathematical programming formulations for the problem that

include wireless medium contention constraints, which are crucial to the problem, and, through numerical analysis using the formulations, we show that network coding achieves 65% higher throughput than conventional multicast in a typical setting. In addition, we showed through simulation that network coding allowed very robust communications with significantly less overhead than conventional multicast.

REFERENCES

- [1] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung. Network information flow. *IEEE Trans. Inform. Theory*, 46(4):1204–1216, July 2000.
- [2] M. Goemans and Y. Myung. A catalog of steiner tree formulations. *Networks*, 23:19–28, 1993.
- [3] T. Ho, M. Médard, R. Koetter, D. R. Karger, M. Effros, J. Shi, and B. Leong. A random linear network coding approach to multicast. Submitted to *IEEE Trans. Inform. Theory*.
- [4] K. Jain, J. Padhye, V. Padmanabhan, and L. Qiu. Impact of interference on multi-hop wireless network performance. In *Proc. ACM MOBICOM*, 2003.
- [5] R. Koetter and M. Médard. An algebraic approach to network coding. *IEEE/ACM Trans. Networking*, 11(5):782–795, Oct. 2003.
- [6] S.-J. Lee, M. Gerla, and C.-C. Chiang. On-demand multicast routing protocol. In *Proc. of IEEE WCNC '99*.
- [7] D. S. Lun, M. Médard, R. Koetter, and M. Effros. On coding for reliable communication over packet networks. Submitted to *IEEE Trans. Inform. Theory*.
- [8] D. S. Lun, N. Ratnakar, M. Medard, R. Koetter, D. R. Karger, T. Ho, E. Ahmed, and F. Zhao. Minimum-cost multicast over coded packet networks. *IEEE Trans. Inform. Theory*, 52(6):2608–2623, June 2006.
- [9] J.-S. Park, D. S. Lun, Y. Yi, M. Gerla, and M. Médard. Codecast: A network coding based ad hoc multicast protocol, 2006. Submitted to *IEEE Wireless Communications*.
- [10] Qualnet. <http://www.scalable-networks.com>.
- [11] Y. E. Sagduyu and A. Ephremides. Joint scheduling and wireless network coding. In *Proc. WINMEE, RAWNET and NETCOD 2005 Workshops*, 2005.
- [12] Y. Wu, P. Chou, and K. Jain. A comparison of network coding and tree packing. In *Proc. IEEE Intl Symp. Information Theory (ISIT)*, 2004.
- [13] Y. Wu, P. Chou, Q. Zhang, K. Jain, W. Zhu, and S. Kung. Network planning in wireless ad hoc networks: a cross-layer approach. *IEEE J. Select. Areas Commun.*, 23(1):136–150, Jan. 2005.