Distributed Network Coding-based Opportunistic Routing for Multicast

Abdallah Khreishah
Department of Computer & Information Sciences
Temple University, Philadelphia PA
akhreish@temple.edu

Issa M. Khalil
Faculty of Information Technology
United Arab Emirates University, UAE
ikhalil@uaeu.ac.ae

Jie Wu
Department of Computer & Information Sciences
Temple University, Philadelphia PA
jiewu@temple.edu

ABSTRACT

In this paper, we tackle the network coding-based opportunistic routing problem for multicast. We present the factors that affect the performance of the multicast protocols. Then, we formulate the problem as an optimization problem. Using the duality approach, we show that a distributed solution can be used to achieve the optimal solution. The distributed solution consists of two phases. In the first phase, the most reliable broadcasting tree is formed based on the ETX metric. In the second phase, a credit assignment algorithm is run at each node to determine the number of coded packets that the node has to send. The distributed algorithm adapts to the changes in the channel conditions and does not require explicit knowledge of the properties of the network. To reduce the number of feedback messages, and to resolve the problem of delayed feedback, we also perform network coding on the feedback messages. We evaluate our algorithm using simulations which show that in some realistic cases the throughput achieved by our algorithm can be double or triple that of the state-of-the-art.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Wireless Communications

General Terms

Algorithm, Performance

Keywords

Network coding, opportunistic routing, multicast, distributed algorithms, coded-feedback

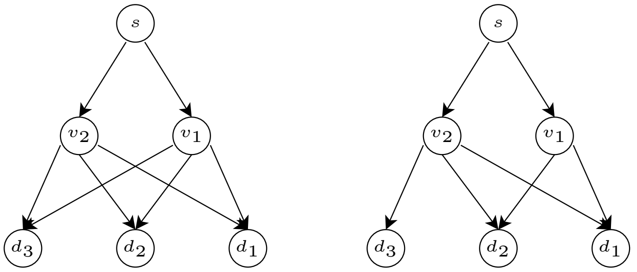
1. INTRODUCTION

Multicasting is an important operation in wireless multihop networks. Its applications range from software updates to video/audio file downloads. Designing an efficient and reliable multicasting protocol for wireless multihop networks is not a straightforward extension from the protocols designed for wireline networks. This is due to the unique features of wireless multihop networks. These features are the lossy behavior, the diversity of the links, the broadcast nature of the links, and the correlations among the links.

Opportunistic routing [1] has been proposed as a way to exploit the unique features of wireless multihop networks. In opportunistic routing, there is no specific next-hop node. Therefore, any node that receives the packets can forward it. To avoid duplication, the receivers of a specific transmission need to coordinate to specify which one of them has to forward the packet that has been received by more than one receiver. This requires the design of a specific MAC protocol. Another shortcoming of opportunistic routing is the difficulty of the extension to the multicast case as stated in [2].

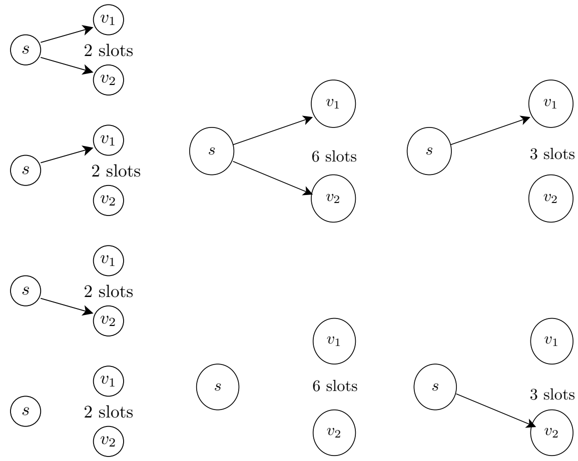
Using *intrasession network coding* [2], the shortcomings of opportunistic routing can be eliminated. In intrasession network coding, the source node divides the message it wants to send into batches, each having K packets of the form $P_1, \ldots, P_K$. The source node keeps sending coded packets of the form $\sum_{i=1}^{K} \gamma_i P_i$, where $\gamma_i, \forall i$ is a random coefficient chosen over a finite field of large enough size, typically 2^8–2^{16}. Upon receiving a coded packet, the intermediate relay node checks to see if the coded packet is linearly independent to what it has received before. If so, it keeps the coded packet, otherwise it drops the packet. When the destination receives K linearly independent packets, this means that it can decode all of the packets of the batch. Therefore, it sends a feedback to the source, using the traditional shortest path, that says: stop sending from this batch and move to the next one. The advantage of using network coding is that the destination node does not need to receive the specific K original packets, but can receive any K linearly independent ones. This resolves the problem of designing a new MAC protocol because we do not insist on receiving a specific packet. Network coding-based opportunistic routing can also be generalized to the multicast case as network coding enhances the achievable throughput for the multicast case with low complexity and in a distributed way, even for wireline networks [3–5].

Despite the attractiveness of using intrasession network coding-based opportunistic routing for the multicast case in wireless networks, most of the works in the literature focus on the unicast case [6–10]. The major challenge that the



(a) (b)

Figure 1: Examples of multicast wireless networks. The second figure is formed by removing the link from v_1 to d_3, which changes the optimal solution.



(a) Independent (b) Correlated (c) Uncorrelated

Figure 2: Illustration of the channel activation scenarios that insure that v_1 and v_2 collectively achieve full rank under different correlation conditions between the channels.

implementation of intrasession network coding-based opportunistic routing for multicast faces is to specify the number of packets that each node has to send. The work in [2, 11] resolves this problem by using an estimation based on the link loss rates. While this approach shows improvement over the non-coded approach, the estimation performed by this approach does not take into account all of the factors that affect the optimal solution. Also, it does not adapt to the changes in the channel conditions.

Obviously, the loss rates of the links are a major factor in deciding the optimal solution. However, there are other factors that affect the optimal solution. To illustrate the other factors, take Fig. 1(a) as an example. In the figure, source s is the source that multicasts its packets to the destination nodes $\{d_1, d_2, d_3\}$. Node s should send enough packets so that the next-hop nodes have, collectively, a full rank matrix. Assume that the batch size is 6, and the loss rate of both of the output links of s is 0.5. If the two output links are correlated, i.e., they are on at the same time and off at the same time, then we need 12 transmissions to ensure that the next-hop nodes collectively achieve full rank. On the other hand, if the two links are independent, we need 8 transmissions; and we only need 6 transmissions, if the links are uncorrelated. Fig. 2 represents the three cases. While most of the work on opportunistic routing and network coding in wireless networks assume independent links, a recent study [12] has shown that the correlation among the links can be arbitrary and dynamically changing over time. Therefore, it is important to formulate and solve the problem under arbitrary correlations among the links.

In addition to the loss rate and the correlations among the links, the reachability of the nodes plays a major role in specifying the number of packets a node has to send. Fig. 1(b) represents the same network in Fig. 1(a) but after removing the link between nodes v_1 and d_3. In this case, node v_2 has to receive vectors from s that can achieve full rank as there is no other path to d_3 without going through this node. Therefore, we need 12 transmission regardless of the correlations among the links.

In this paper, we show that these three factors - the loss rate, the correlation among the links, and the reachability of the node - can be used to design optimal network coding-based opportunistic routing multicast in wireless multihop networks. Our contribution lies in the following:

- We formulate the optimal network coding-based opportunistic routing for multicast as an optimization problem. We use a wireless to wireline mapping mechanism to show that our formulation achieves the maximum possible rate with intrasession network coding.
- We develop a fully distributed algorithm for the problem such that each node only needs local information. The distributed algorithm consists of two phases: reliable multicast tree construction and distributed credit assignment. The distributed algorithm adapts to the changes in the channel conditions and converges to the optimal solution. Also, it does not need an explicit knowledge of the channel conditions or the correlation among the links.
- We integrate our algorithm with the coded feedback approach to reduce the number of feedback messages and eliminate the assumption of immediate feedback.
- Using simulations we compare our results to the state of the art opportunistic routing protocol for multicast ,MORE [2], and show the effectiveness of our protocol in maximizing the throughput.

The rest of the paper is organized as follows. In Section 2 we present our settings followed by the mapping of wireline to wireless networks in Section 3. We present the formulation of the problem in Section 4 and derive a distributed algorithm based on the formulation in Section 5. The integration of the coded-feedback approach with our algorithm is presented in Section 6. We evaluate our algorithm by using simulations in Section 7, and we conclude the paper in Section 8.

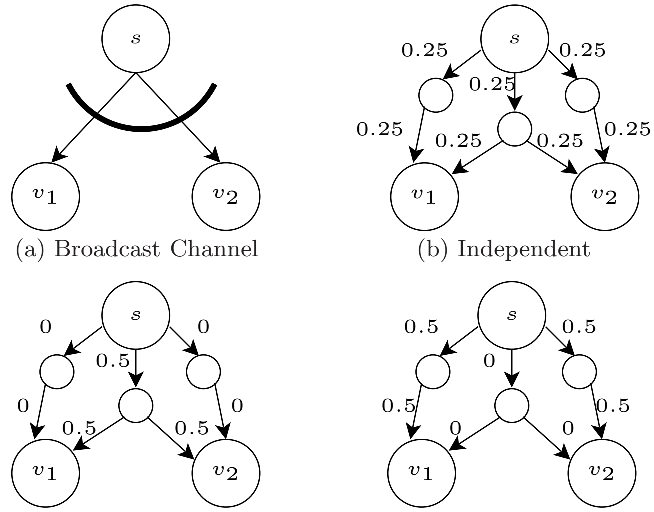
2. SETTINGS

In this paper, we consider a network that is represented by a set of nodes V. The links between the nodes are lossy and time varying. A transmission by a node can be received by any subset of next-hop nodes. We represent this by a hyperedge (u, J), where u is the node that performs the transmission and J is a subset of the set of next-hop nodes. Unlike previous work, the correlation among individual links of a given hyperedge is arbitrary, and we don't need to measure it as has been done in [12]. There are N multicast sessions in the network, each with a source s_i, a set of destination nodes D_i, a rate R_i, and a utility function $U_i(R_i)$, $\forall i \in \{1, \ldots, N\}$. Like most of the opportunistic routing protocols [2, 6, 8, 10], we are interested in the transmission of large files. Therefore, the throughput is the most important factor, and the individual packet delays are of lower importance. Also, the transmission scheme has to be reliable such that every packet sent by the source of the session has to be received by all of the destinations of that session, regardless of the network bandwidth to that receiver. This means that the same rate R_i has to be supported by all of the destinations of session i. Since we are using intrasession network coding, one important factor to determine is the rate of linearly independent packets that a node has to successfully deliver to next-hop nodes. To model this factor, we use the concept of credits similar to [2, 9]. Symbol X^i_{uv} is used to represent the rate of credits for session i that node u gives to node v, which represents the total number of linearly independent packets that node v has to forward to next-hop nodes (out of the packets it has received from node u). Therefore, the total rate of credits for session i at node u would be $\sum_{v \in V} X^i_{vu}$, and these credits will be distributed to next-hop nodes. We also use α^i_u to represent the fraction of time that node u is scheduled to send the packets of session i. Symbol R_{uJ} represents the rate of packets that are sent by node u and are received by any of the nodes in J. Since our solution is based on building multicast trees, we use $T(i)$ to represent the multicast tree for the i-th multicast session. For any node $u \in T(i)$, we use $RC(u,i)$ to represent the descendent destination nodes for node u on T(i). Therefore, if the destination node $d \in DC(u,i)$, then $\exists$ a path from u to d on $T(i)$. We also use $I(u,i)$ $(O(i,u))$ to represent the direct parents (children) of node u on tree $T(i)$. Also, for a set of nodes J, another node $d \in RC(J,i)$, if $d \in RC(v,i), \forall v \in J$.

3. WIRELESS NETWORKS AND THEIR WIRELINE COUNTERPART

For wireline networks, and for a single multicast session, it has been shown in [3] that the maximum multicast rate is the minimum of the min-cut max-flow between the source and each destination. For multiple multicast sessions in wireline networks, intrasession network coding -where coding is performed on the packets of every session separately- is used to share the bandwidth of the network [13, 14]. Therefore, the optimal algorithm in wireline networks has to achieve the minimum of the min-cut max-flow value between the source and each destination for the single source multicast problem. Also, the same optimal algorithm can use intrasession network coding to share the resources of the network in the case of multiple multicast sessions. This is due to the difficulty of using intersession network coding [15, 16], which codes packets of different flows together.

Due to the broadcast nature of wireless links and the correlations among the links, how we can find the min-cut max-flow between two nodes is not clear. In this section, we perform mapping from any wireless network with any channel conditions to its corresponding wireline network, such that the capacity properties of the wireless network are preserved. A similar conversion has been done in [17, 18] for



(a) Broadcast Channel (b) Independent (c) Correlated (d) Uncorrelated

Figure 3: Wireless to wireline mapping with different correlations among the channels. The big circles represent the original nodes and the small ones represent the added auxiliary nodes.

lossless wireless links. The mapping can be done on each broadcast link by introducing an auxiliary node for each set of receivers in the broadcast link and then by connecting the transmitter node of the broadcast link to each auxiliary node with a separate directed link and the auxiliary node to the set of receivers it represents with directed links. The weight assigned to each link that uses a given auxiliary node can be computed as the transmission bandwidth of the source node of the broadcast link times the probability that all of the outgoing nodes of the auxiliary node *exclusively* overhear a given transmission. Fig. 3 represents this mapping with different correlations among the broadcast links.

In Fig. 3(b), we assume that the two links of the broadcast channel in Fig. 3(a) are independent. Therefore, the probability that only v_1 (v_2) overhears a transmission would be $p_{s,v_1} \times (1 - p_{s,v_2})$ ($p_{s,v_2} \times (1 - p_{s,v_1})$), where p_{uv} is the delivery rate from node u to v. Similarly, the probability that both of v_1 and v_2 overhear a transmission would be $p_{s,v_1} \times p_{s,v_2}$. Assuming the fraction of time where node s is scheduled equals one justifies the weights assigned to the links on Fig. 3(b). In Figs. 3(c) and 3(d), the channels are correlated and uncorrelated, respectively, which justifies the weights assigned to the channels in these Figs.

Note that this mapping allows us to make an equivalent wireline network model for any wireless network with arbitrary characteristics. For example, if we assume that the nodes in Fig. 1(a) use orthogonal channels, which means that the nodes can be scheduled for an arbitrary fraction of the time, the min-cut max-flow from s to each of the destinations will be 0.75, 0.5, and 1, respectively, if the links are independent, correlated, and uncorrelated, respectively. The mapping can also handle the case where the nodes are scheduled for limited amount of time due to interference by multiplying the weight of every link by the fraction of time that the sender node of that link is scheduled. Note also that our algorithm does not require us to do the mapping. We use the mapping only to show that our algorithm achieves

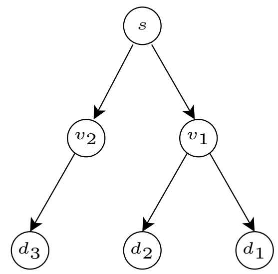


Figure 4: An example of a multicast wireless network formed by removing links from the networks in Fig.s 1(a) and 1(b).

the min-cut max-flow bound of a wireline network that is equivalent to our wireless network.

4. PROBLEM FORMULATION

Our problem can be formulated as follows.

Maximize:

$$\sum_{i=1}^{N}(U(R_i))$$

Subject to:

$$\sum_{v:v\in I(u,i)} X^i_{vu} - \sum_{\substack{v:v\in O(u,i)\\ d\in RC(v,i)}} X^i_{uv} \leq \begin{cases} -R_i & u = s_i \\ 0 & else \end{cases}$$

$$\forall i, \forall u \in T(i)\backslash D_i, d \in D_i, d \in RC(u,i) \qquad (1)$$

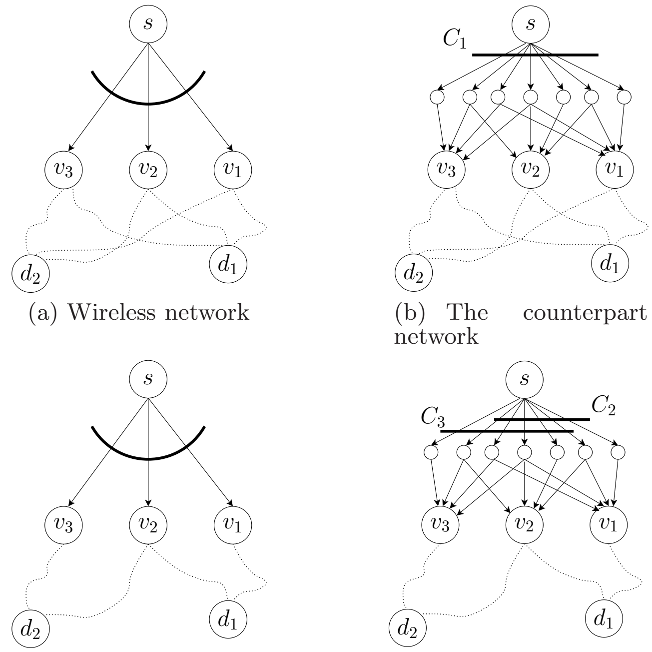
$$\sum_{\substack{v:v\in J\\ d\in RC(J,i)}} X^i_{uv} \leq \alpha^i_u R_{uJ}$$

$$\forall i, \forall (u,J), \forall d \in D_i, d \in RC(u,i) \qquad (2)$$

We assume that the utility function $U_i(R_i)$ is non-decreasing and strictly concave. If the utility function is chosen properly, maximizing the objective function will achieve different kinds of fairness among the sessions [19]. Examples of $U(R_i)$ would be $w_i \log(1+R_i)$ and $w_i \frac{R_i^{1-\gamma}}{1-\gamma}$, where $0 \leq \gamma \leq 1$, and w_i is the weight assigned for session i.

Here, α^i_u depends on the underlying interference model. Typically, it corresponds to the convex hull of all of the achievable rates at all links [20]. Generally, the corresponding optimal scheduling policy is NP-hard, and approximation algorithms are used. While it is easy to extend our formulation and algorithm to include optimal scheduling, we consider that scheduling is of secondary importance, and we use IEEE 802.11 in the simulations. We do this to focus on the network coding part and to have a fair comparison with the other approaches that use IEEE 802.11.

The first set of constraints represents balance equations so that at every node, and for every destination d, the total number of received credits should be no less than the total number of credits assigned to next-hop nodes that can reach destination d. For example, in Fig. 1(a), node s can split the credits it has between nodes v_1 and v_2 because all of $\{d_1, d_2, d_3\}$ are reachable from both v_1 and v_2. On the other hand, in Fig. 1(b), node s can't split the credits it has between v_1 and v_2 because d_3 is reachable by only v_2, and



(a) Wireless network (b) The counterpart network

(c) Wireless network (d) The counterpart network

Figure 5: Two examples of wireless networks and their wireline counterpart. The dotted line between two nodes means there exists a path between them. a) Both d_1 and d_2 are reacheable from v_1, v_2, and v_3. b) The wireline counterpart of part (a) with the cut C_1 that is an upperbound on the number of credits that can be forwarded to v_1, v_2, and v_3. c) Both d_1 is reacheable from v_1, v_2, and d_2 is reacheable from v_3, v_3. d) The wireline counterpart network of part (c). Since not all of the d nodes can be reached by all of the v nodes, we have two different constraints on the maximum credits that can be sent to both v_1 and v_2 and to both v_2 and v_3. These constraints are represented by the two cuts C_2 and C_3 on the mapped wireline netwwork.

v_2 has to receive the same number of credits assigned to s which is confirmed by the first set of constraints.

The second set of constraints represents the fact that if a packet is received by more than one of the next-hop nodes, such that all of these nodes can reach destination d, only one of them can use this packet to increase its credits. For example, in both Figs. 1(a) and 1(b), if the links of node s are independent, and assuming that node s has sent m randomly coded packets, then both v_1 and v_2 will receive $0.5m$ linearly independent packets. Half of the received packets by v_1 are also received by v_2 due to the independence of the links. Therefore, the total credits assigned to both v_1 and v_2 can not exceed $0.75m$ because both v_1 and v_2 can reach both d_1 and d_2. However, in Fig. 3, the total credits assigned for both v_1 and v_2 can be m because there is no destination that is reachable by both v_1 and v_2, which nullifies the effect of the commonly received packets by both v_1 and v_2.

LEMMA 1. *Assuming the use of perfect scheduling for one session, the achievable rate of the formulation in ((1)-(2)) is the minimum of the min-cut max-flow from the source to* *each of the destinations after converting the network to its wireline counterpart.*

PROOF. It is easy to show that the first set of constraints guarantees achieving the min-cut max-flow bound from the source to the destinations for wireline networks. Therefore, what remains to show is that the second set of constraints is equivalent to mapping the wireless network to its wireline counterpart described in the previous section.

For a given broadcast channel, the rank of the matrix, represented by the packets sent by the sender of the broadcast channel and received by a subset of the receivers node of this broadcast channel, is upper bounded by the probability that any one of these nodes receives a given sent packet multiplied by the transmission rate of the sender node of that broadcast channel. This is because, if a packet is received by more than one of the receivers of the broadcast channel, the rank of the matrix that represents all of the packets at these nodes that received the transmitted packet can not be increased by more than one. If all of these nodes can reach a specific destination d_1, then the total number of linearly independent packets that these receiver nodes can push to the destination d_1 can not exceed the rank of the matrix, which limits the number of credits assigned to these nodes, i.e., for a packet that has been received by two nodes, only one credit can be assigned to both of these two nodes. However, if a group of these receiver nodes can reach destination d_1 but not d_2, and another group of these receivers nodes can reach destination d_2 but not d_1, then, if a packet is received by nodes in the two groups, the packet can increase the rank of the matrix in both of these groups because the matrices are destined to two different receivers. This means that the packet that is received by the two groups can increase the number of credits by two instead of one.

Note that the rank of the matrix at a group of receiver nodes of a broadcast channel equals the max-flow from the source of the broadcast channel to these receiver nodes on the mapped wireline counterpart of the wireless broadcast channel, as illustrated in Fig. 5. Therefore, the second constraint set is equivalent to mapping the wireless network to its wireline counterpart.

InFigure, if a destination is reacheable by all of the nodes in the set $\{v_1, v_2, v_3\}$, then the cut in Fig. 5(b) represents an upper bound on the maximum number of credits that can be assigned to the nodes in the set $\{v_1, v_2, v_3\}$. However, when there is no destination node that is reachable by all of the nodes in $\{v_1, v_2, v_3\}$, we need to consider other cuts as in Fig. 5(d), and these cuts are also represented by the second constraints. □

So far we have shown that any wireless network can be mapped to a wireline counterpart such that both of them have the same capacity characteristics. We have also shown that for one source multicast, our formulation achieves the min-cut max-flow bound on the mapped wireline counterpart network, which is the maximum achievable rate.

PROPOSITION 1. *For multiple sessions, the achievable rates of our formulation represent the optimal solution with intrasession network coding.*

PROOF. With intrasession network coding no coding is permitted between different sessions. Therefore, if the maximum rate can be achieved for every session by Lemma 1, then by using the time sharing variables α_u^i, the optimal solution with intrasession network coding can be achieved. □

Note that the optimal multiple unicast sessions case [6–10] can be obtained as a special case of our formulation. This can be done by assigning every destination set to a single node. Also, the tree can be replaced by a single or multiple paths. Alternatively, we can use a back-pressure algorithm [21] to jointly assign the credits and to find the paths to the destination.

5. DISTRIBUTED ALGORITHM

5.1 Constructing Broadcast Trees

In this section, we provide a simple way for constructing the broadcasting tree and setting up $RC(u, i)$ for every node u. Each node in the network computes the ETX metric [22] to the destination nodes of every session. Also, every node u initializes $RC(u, i), \forall i$ to an empty set. For every session i, the source node s_i sets $RC(s_i, i)$ to D_i. For every destination node in its $RC(s_i, i)$, the source node selects each node u that has a lower ETX metric value to that destination than its own as a forwarder node and adds that destination to $RC(u, i)$. Every intermediate node u repeats the same comparison process, but just for the destinations in its own $RC(u, i)$ instead of $RC(s_i, i)$.

5.2 Structure of the Optimal Solution

After building the broadcast tree, the next step is to develop a distributed algorithm that assigns credits to the nodes along the tree, which we discuss in this Section.

Since the constraints are linear, we have a convex optimization problem. Therefore, there is no duality gap, and we can use the duality approach to solve the problem [23, 24].

Ignoring the scheduling constraints, we associate a Lagrange multiplier q_{ud}^i with each constraint in (1) and another one $\lambda_{(u,J)d}^i$ with each constraint in (2). This results in the following Lagrange function $L(\mathbf{R}, \mathbf{X}, \mathbf{q}, \lambda)$ that is equal to

$$\sum_{i=1}^{N} U_i(R_i) - \sum_{i,u}\Big(\sum_{\substack{d:d\in D_i \\ d\in RC(u,i)}} q_{ud}^i \Big(\sum_{v:v\in I(u,i)} X_{vu}^i - \sum_{\substack{v:v\in O(u,i) \\ d\in RC(v,i)}} X_{uv}^i \Big)\Big)$$
$$- \sum_{i,(u,j)} \sum_{\substack{d:d\in D_i \\ d\in RC(u,i)}} \lambda_{(u,j)d}^i \Big[\Big(\sum_{\substack{v:v\in J \\ d\in RC(J,i)}} X_{uv}^i \Big) - \alpha_u^i R_{u,J}\Big]$$

With simple changes of variables, the Lagrange function becomes

$$\sum_{i=1}^{N} [U_i(R_i) - \sum_{d:d\in D_i} q_{s_i d}^i R_i]$$
$$+ \sum_{u,i} \sum_{v} \Big(\sum_{\substack{d:d\in D_i \\ d\in RC(u,i)}} (q_{ud}^i - \sum_{\substack{J:v\in J \\ d:d\in RC(J,i)}} (\lambda_{(u,J)d}^i))$$
$$- \sum_{\substack{d:d\in D_i \\ d\in RC(v,i)}} q_{vd}^i) X_{uv}^i + \sum_{i,(u,J)} \sum_{\substack{d:d\in D_i \\ d\in RC(u,i)}} \lambda_{(u,J)d}^i R_{(u,J)}$$

The Lagrange function is separable [24], which means that the problem can be solved in a distributed way by using the gradient method as follows.

Source Algorithm: Each source s_i selects its rate at each time slot as follows:

$$R_i(t) = \arg\max_{R_i}[U(R_i) - \sum_{d:d\in D_i} q^i_{s_i d}(t) R_i] \quad (3)$$

Intermediate Node Algorithm: Each intermediate node u selects the number of credits for session i to transfer to all of its next-hop nodes at each time slot as follows:

$$\{X^i_{uv}(t)\} = \arg\max_{\mathbf{X}} \sum_{v\in O(u,i)} \Big(\sum_{\substack{d:d\in D_i \\ d\in RC(u,i)}} (q^i_{ud}(t) - \sum_{\substack{J:v\in J \\ d:d\in RC(J,i)}} (\lambda^i_{(u,J)d}(t))) - \sum_{\substack{d:d\in D_i \\ d\in RC(v,i)}} q^i_{vd}(t) \Big) X^i_{uv} \quad (4)$$

Dual Variables Updates: The dual variables can be updated in a distributed way as follows:

$$q^{id}_u(t+1) = [q^i_{ud}(t) + \beta^i_{ud}(\sum_{v:v\in I(u,i)} X^i_{vu}(t) - \sum_{\substack{v:v\in O(u,i) \\ d\in RC(v,i)}} X^i_{uv}(t))]^+, \quad (5)$$

$$\lambda^i_{(u,J)d}(t+1) = [\lambda^i_{(u,J)d}(t) \quad (6)$$

$$+ \beta^i_{(u,J)d}(\sum_{\substack{v:v\in J \\ d\in RC(J,i)}} X^i_{uv}(t) - \alpha^i_u(t) R_{uJ}(t))]^+. \quad (7)$$

Here, $[.]^+$ is a projection on the positive real numbers, and β is the step size.

THEOREM 1. *The algorithm converges to the optimal solution of the problem.*

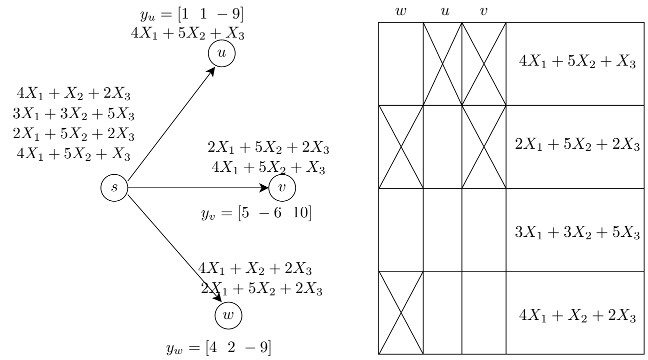
Due to space limitations, we remove the proof.

6. INTEGRATING THE ALGORITHM WITH THE CODED FEEDBACK APPROACH

6.1 Challenges

The algorithm represented by ((3)-(7)) converges to the optimal solution, but it has the following shortcomings. Firstly, the algorithm requires a large amount of feedback messages. For example, if the batch size is 32, and the node which has l next-hop forwarders sends 32 packets from the batch, we need $(32l)$ feedback messages. Secondly, the links are lossy, which increases the number of required transmissions for the feedback messages. Thirdly, the algorithm assumes immediate hop-by-hop feedback which is not realistic due to the scheduling problem in wireless networks. Finally, it takes very long time to converge. Also, to converge, the generation size should be very large. However, for practical reasons, and in order to carry the coding coefficients, the generation size should be small, typically 32. Despite these shortcomings, the structure of the solution above inspires us to design an efficient distributed algorithm for the problem. We introduce our solutions to the above problems through the use of the coded-feedback approach.

6.2 Integrating the Coded Feedback Approach with the Algorithm



(a) The transmission scenario (b) The table created by the algorithm

Figure 6: An example representing our coded feedback approach. An X in a cell means that the coded packets representing the row has been received by the node representing the column.

In this Section, we use the coded-feedback approach, which has been proposed recently [10, 25, 26], to resolve the previously mentioned shortcomings of our basic algorithm. The objective of the coded feedback approach is to perform network coding on the feedback messages, such that the transmitter node can learn about the linear space that each receiver node has received through the channel. The common way of performing coded feedback is through the null space. The null space of the matrix A is the linear space of vectors such that the result of multiplying anyone of these vectors by A equals zero. For example, if y belongs to the null space of A, then $y^T A = 0$, where y^T is the transpose of y.

Take Fig. 6(a) as an example, in which node s sends four coded packets. Node v receives two of them. Node v can compute the null space[1] of the space of the packets it receives, choose a vector from this space, and send it back to node s. As is illustrated in Fig. 6(b), node s can now multiply this vector with each of the packets it has sent. If the result is zero, node s can infer that the packet has been received by node v with high probability. Otherwise, node s knows that the packet has not been received by node v. By using a hash table, the work in [10] makes the false positive probability very low, about 10^{-10}. In Fig. 6(a), y_v is a randomly chosen vector from the null space of the received packets by node v. If node s multiplies the y_v vector with each of the vectors representing it's own packets, the result will be zero for the following two vectors, $4X_1 + X_2 + 2X_3$ and $4X_1 + 5X_2 + X_3$. Therefore, node s will conclude that these packets have been received by node v. On the other hand, the result of multiplying y_v with the vectors representing the other coded packets sent by s will be non-zero, and node s will conclude that these packets have not been received by node v.

In order to integrate the coded-feedback approach with our algorithm, we first relax the Lagrange function by removing the constraints in (2) from the objective function of

[1]Note that the example here is for illustrative purpose. That is why we use the negative sign for the vector in the null space. In reality, the elements of the vectors in the null space will be positive, and their values depend on the size of the finite field.

the dual problem, and we keep them in the constraints. The new Lagrange function becomes:

$$\sum_{i=1}^{N}[U_i(R_i) - \sum_{d:d\in D_i} q^i_{s_i d} R_i] + \sum_{u,i}\sum_{v}(\sum_{\substack{d:d\in D_i\\ d\in RC(u,i)}} q^i_{ud} - \sum_{\substack{d:d\in D_i\\ d\in RC(v,i)}} q^i_{vd})X^i_{uv}$$

subject to (2).

Therefore, we end up with one type of queue (dual variables) q^i_{ud}. Also, the new dual problem is separable, and we have the same source algorithm with the following modified intermediate node algorithm:

Intermediate Node Algorithm: Each intermediate node u selects the packets to send and the number of credits for each session to transfer to each of its next-hop nodes at time t by solving the following optimization problem:

$$\{X^i_{uv}(t)\}_{v\in V, i\in\{1,\ldots,N\}} = \arg\max_{\mathbf{X}} \sum_{i=1}^{N}\sum_{v}(\sum_{\substack{d:d\in D_i\\ d\in RC(u,i)}} q^i_{ud}(t) \tag{8}$$

$$- \sum_{\substack{d:d\in D_i\\ d\in RC(u,i)}} q^i_{vd}(t))X^i_{uv} \tag{9}$$

Subject to:

$$\sum_{\substack{v:v\in J\\ d\in RC(v,i)}} X^i_{uv} \le \alpha^i_u R_{uJ}$$

$$\forall i, \forall (u,J), \forall d \in D_i, d\in RC(u,i) \tag{10}$$

The relay node u has to perform two decisions that lead to maximizing (8), subject to (10).

- It has to decide the session that the packet should be sent from.
- It has to also decide the number of credits to be assigned to each next-hop node.

To perform the first decision optimally, the relay node should choose session i^* that achieves the maximum value for the following among all of the sessions.

$$\{X^i_{uv}(t)\} = \arg\max_{\mathbf{X}} \sum_{v\in O(u,i)} \Big(\sum_{\substack{d:d\in D_i\\ d\in RC(u,i)}} q^i_{ud}(t) - \sum_{\substack{d:d\in D_i\\ d\in RC(v,i)}} q^i_{vd}(t)\Big) X^i_{uv} \tag{11}$$

Subject to:

$$\sum_{\substack{d\in RC(v,i)\\ v:v\in J}} X^i_{uv} \le \alpha^i_u R_{uJ} \tag{12}$$

$$\forall i, \forall (u,J), \forall d \in D_i, d\in RC(J,i)$$

To do so, for every session i, node u ranks the next-hop nodes v according to the backlog difference $(\sum_{\substack{d:d\in D_i\\ d\in RC(u,i)}} q^i_{ud} - \sum_{\substack{d:d\in D_i\\ d\in RC(v,i)}} q^i_{vd})$. Then, it gives as many virtual credits[2] to this next-hop node, subject to (12). For every sent packet, next-hop node v gets a virtual credit if node v has received the packet, and no other node w has received the packet, such that (1) w has a higher backlog difference and (2) w has a common receiver, i.e., $RC(u,i) \cap RC(w,i) \ne \phi$. This can be checked by using the coded feedback approach. Let us denote the virtual credit for session i and node v by Z^i_v; then, node u calculates $w_i = \sum_v ((\sum_{\substack{d:d\in D_i\\ d\in RC(u,i)}} q^i_{ud} - \sum_{\substack{d:d\in D_i\\ d\in RC(v,i)}} q^i_{vd})Z^i_v)$, such that all of the v nodes have positive backlog differences. Then, node u selects the session that achieves the maximum w_i. Algorithm 1 describes the above strategy.

Algorithm 1 Selecting the packet to send

$Z^i_v \leftarrow 0, \forall i, \forall v$
for $i \leftarrow 1$ N **do**
  Sort next-hop nodes according to $(\sum q^i_{ud} - \sum q^i_{vd})$.
  Remove the nodes with negative backlog
  set T to the remaining nodes
  for Each sent packet P **do**
    set S to each node v in T such that $y^{iT}_v * P$ is zero.
    while S is not empty **do**
      Choose node v with the highest non-negative backlog difference from S
      Remove each node w from S such that $RC(u,i) \cap RC(w,i) \ne \phi$
      set $Z^i_v \leftarrow Z^i_v + 1$
    end while
  end for
  set $w_i \leftarrow \sum_v Z^i_v (\sum_{\substack{d:d\in D_i\\ d\in RC(u,i)}} q^i_{ud} - \sum_{\substack{d:d\in D_i\\ d\in RC(v,i)}} q^i_{vd})^+$
end for
select $i^* \leftarrow \arg\max_i w_i$.
send a packet from session i^*

Node u can perform the second decision by assigning the credits to next-hop nodes in a batch-by-batch manner. Therefore, for each session i, node u keeps sending packets from the batch with the smallest index until it makes sure that for each receiver $d \in RC(u,i)$, the total number of linearly independent packets from that batch, received by the next-hop nodes that have paths to d is no less that the total credit it is assigned for that batch. At that time, this node assigns the credits for next-hop nodes and moves to the next batch of that session. Note that this approach might increase the delay of individual packets, but the total throughput is not affected if the size of the file is very large. This is because the source does not wait for the receiver to decode the batch in order to move to the next batch. As long as next-hop nodes are assigned credits for the batch, the node moves to the next batch. This approach for delaying the sending of packets until enough feedback has been received, is used in different works and is shown to achieve the capacity under specific conditions [27, 28].

Every time node u receives a vector in the null space from the next-hop node, it multiplies that vector with all of the

[2]Note that these are different from the actual credits that will be distributed as a strategy for the second decision the node has to perform. These credits are just for knowing the packet of which session should be sent.

Algorithm 2 Credits assignment algorithm for session i

Set $C_v^i \leftarrow 0, \forall v$
for Each sent packet P **do**
  $S \leftarrow$ each nex-hop node v with positive back-log difference and with $y_v^{iT} P = 0$ and $C_v^i \leq C_u^i$
  Sort nodes in S with respect to the backlog difference.
  while $S \neq \phi$ **do**
    Choose node v from S with the highest back-log difference.
    Set $C_v^i \leftarrow C_v^i + 1$
    Remove each node w from S such that $\exists d \in RC(u,i)$ s.t. $RC(v,i) \cap RC(w,i) = d$
  end while
end for

packets it has sent so that it can know the number of linearly independent packets that has been received by next-hop nodes. For each $d \in RC(u,i)$, once that rank for all next-hop nodes that have paths to d becomes equal to or greater than the number of credits assigned for that batch at that node, the node distributes its credits to next-hop nodes in a fashion similar to Algorithm 1. However, this time the node only focuses on one session i, and the credits that are assigned are real not virtual credits. Algorithm 2 represents the credit assignment algorithm. In the algorithm, C_u^i represents the total credits assigned to node u.

6.3 Details of the Practical Protocol

So far, we have identified the structure of the optimal solution and discussed the major challenges that face its implementation. We have then designed a back-pressure algorithm that uses the coded feedback approach to resolve these challenges. In this section, we outline the details of implementing our algorithm under practical settings.

In our protocol, every node maintains the following information: the received and sent coded packets, the available number of credits, and the batch and session number of the received and sent packets. We adopt a packet format similar to [2, 10], such that each packet has 1500 bytes of data. The packet also contains the coefficients of the coding vector along with its session and batch numbers. The packet contains the three most recent batches it has received, each with a vector from the null space of the packets in that batch. The packet contains the number of currently queued credits. The packets also contain the number of credits assigned to each next-hop node and the batch number for these credits. As can be seen, the overhead is about 1-2%, which is very small.

In every time slot, the source node computes the source rate according to (3). This adds more credits to the source node's queue. The source node moves to the next batch when the number of credits assigned to the current batch equals the size of the batch. When an intermediate node transmits a packet, it fills the null space fields with randomly chosen vectors from the null spaces of the batches it is currently receiving coded packets from. Note that the coded packets that the node sends, and the null space vectors that the node generates at any given time, could be for different batches. A node keeps sending packets from a batch until it makes sure using the coded feedback approach, that for each destination the total number of linearly independent packets received by all next-hop nodes that can reach that destination, is no less than the number of credits it has for this batch. At that time, the node transfers the credits to the next-hop nodes using Algorithm 1. An intermediate node sends the credit assignment information for the last three batches it has made assignments for. Note that there is a very small feedback or credit assignment overhead due to the integration with the data packets. Also, assigning the credits to next-hop nodes when enough packets have been sent from a batch serves two purposes. Firstly, it gives enough time for the feedback packets and the credits to reach the intended nodes due to the lossy behavior of the links. Secondly, it allows Algorithm 2 to find the optimal credit assignments.

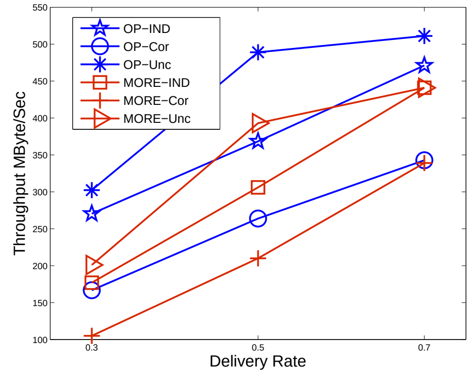


Figure 7: Simulation results for the topology in Fig. 1(a).

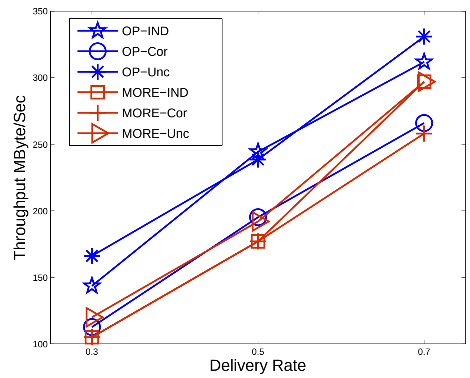


Figure 8: Simulation results for the topology in Fig. 1(b).

7. EVALUATION

In this section, we provide simulation results to illustrate the effectiveness of our protocol over the state-of-the-art opportunistic routing-based multicast protocol, MORE [2]. We start by showing results for the illustrative topologies presented in the introduction section, and then we show results for a 4×4 grid network. We develop our simulations using MATLAB. We also use similar values of parameters to MORE [2] to make a fair comparison.

7.1 Results on Illustrative Topologies

We simulate both MORE and our protocol on the topology in Fig. 1(a). We choose the following values for the delivery rate of the links $0.3, 0.5$, and 0.7. We also vary the correlation among the links according to the κ-factor in [12]: we set this parameter to $-1, 0$, and 1 respectively. When the value of $\kappa = -1$, this represents the uncorrelated links, and we add the symbol, Unc, to the name of the scheme in the plots. Similarly, when $\kappa = 0$ or 1, we add IND or Cor symbols, respectively, to the name of the scheme in the plots. We use a batch size of 32 packets, and we assume that transmission bandwidth of all of the nodes is 1500Mbytes/sec. We also use IEEE 802.11 to perform scheduling. We use the symbol, OP, to represent our protocol and MORE to represent MORE.

Fig. 7 represents the simulation results for the topology in Fig. 1(a). The results show that our protocol always results in more gain compared to MORE; also, the gain of our protocol is maximized when we have low delivery rate links. As illustrated in the figure, when the delivery rate is 0.3, the gain of our protocol is in the range of 50-75%, depending on the correlation among the links, while when the delivery rate is 0.7, the gain is in the range of 2-20%. Also, MORE does not exploit the benefit of having uncorrelated or independent broadcast links, as the gain of our protocol is maximized in these cases. This is alligned with the expectations in [12] that we have many coding opportunities under these cases which are not fully captured by MORE. Under the correlated case, there are not many coding opportunities, which justifies the small gain of our protocol over MORE. Fig. 8 shows the results for the topology in Fig. 1(b). Our protocol still has a gain of 5-45% depending on the delivery rate and the correlations, even with the limited coding opportunity caused by the bottleneck receiver d_1, as explained in the introduction.

7.2 Results on a 4×4 Grid Topology

We perform simulations on a 4×4 grid topology. We set up the link delivery rate to 0.5. We vary the correlation among the links to be the following three cases: independent, correlated, and uncorrelated. We place the source at one of the corners and place the destinations randomly on the two sides of the grid topology opposite to the source. We vary the number of destinations to 2, 4, and 6. We select the following utility function, $U(R_i) = log(R_i)$. We plot the results in Fig. 9.

As can be noted from the figure, our protocol results in a higher gain compared to MORE in all cases. The gain obtained by our protocol varies from 50% to 4-fold depending on the number of receivers and the correlations among the links. The gain increases as we increase the number of receivers. When we have two receivers, the gain varies from 50 to 90% while it increases to about 3 to 4-fold with six receivers. Also, the throughput decreases dramatically as we increase the number of receivers in the MORE case while decreases slowly in our case. The reason is that MORE works in a batch-by-batch manner. Therefore, its throughput is limited by the length of the path to the farthest receiver, while in our protocol, the throughput is limited to the min-cut max-flow between the source and the worst receiver, which has a smaller effect on the throughput than the length of the worst path. Our protocol takes advantage of the coding opportunities created by uncorrelated and independent links, which agrees with the conclusion in [12]. This is justified by a gain of about 70% when uncorrelated links are used compared to correlated links, and a gain of about 50% when independent links are used compared to correlated links. On the other hand, MORE throughput increases by about only 30% when uncorrelated or independent links are used compared to the correlated case.

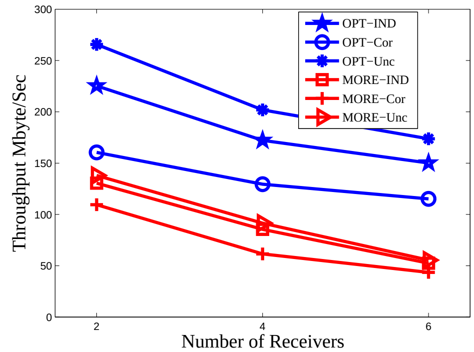


Figure 9: Simulation results for one session and different numbers of receivers on a 4×4 grid topology.

8. CONCLUSION

In this paper we tackle the problem of optimal network coding-based opportunistic routing for multicast, which has received less attention from the community compared to the unicast problem. We identify the factors that affect the optimal solution, which are the delivery rates of the links, the correlations among the links presented recently in [12], and the reachability of the nodes to the different destinations. We formulate the problem as an optimization problem and show that it achieves the maximum possible rate by using mapping from wireless to wireline framework. We then develop a distributed solution based on the duality approach. We integrate our solution with the coded feedback approach so that it can be implemented with a delayed and lossy feedback environment. We evaluate our protocol by using simulations which show the effectiveness of our protocol.

9. ACKNOWLEDGMENT

This research was supported in part by NSF grants ECCS 1128209, CNS 1065444, CCF 1028167, CNS 0948184, and CCF 0830289.

References

[1] S. Biswas and R. Morris, "Opportunistic routing in multi-hop wireless networks," in *Proc. ACM Special Interest Group on Data Commun. (SIGCOMM), Philadelphia, PA, USA*, Sept 2005.

[2] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, "Trading structure for randomness in wireless opportunistic routing," in *ACM Special Interest Group on Data Commun. (SIGCOMM) Kyoto, Japan,*, Aug 2007.

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

[4] S. Jaggi, P. Sanders, P. A. Chou, M. Effros, S. Egner, K. Jain, and L. Tolhuizen, "Polynomial time algorithms for multicast network code construction," *IEEE Trans. on Information Theory*, vol. 51, no. 6, pp. 1973–1982, 2005.

[5] T. Ho, M. Médard, R. Koetter, D. Karger, M. Effros, J. Shi, and B. Leong, "A random linear network coding approach to multicast," *IEEE Trans. Inform. Theory*, vol. 52, no. 10, pp. 4413–4430, 2006.

[6] B. L. Y. Lin and B. Liang, "CodeOR: Opportunistic routing in wireless mesh networks with segmented network coding," in *in the Proceedings of the 16th IEEE International Conference on Network Protocols (ICNP 2008), Orlando, Florida*, October 2008.

[7] C. Gkantsidis, W. Hu, P. Key, B. Radunovic, P. Rodriguez, and S. Gheorghiu, "Multipath code casting for wireless mesh networks," in *in Proc. of ACM CoNEXT*, 2007.

[8] X. Zhang and B. Li, "Optimized multipath network coding in lossy wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 27, no. 5, pp. 622–634, 2009.

[9] B. Radunovic, C. Gkantsidis, P. Key, and P. Rodriguez:, "Toward practical opportunistic routing with intra-session network coding for mesh networks," *IEEE/ACM Trans. Networking*, vol. 18, no. 2, pp. 420–433, 2010.

[10] D. Koutsonikolas, C.-C. Wang, and Y. Hu, "CCACK: Efficient network coding based opportunistic routing through cumulative coded acknowledgments," in *in Proceedings of the 29th Conference on Computer Communications (INFOCOM), San Diego, USA,*, March 2010.

[11] D. Koutsonikolas, Y. Hu, and C. Wang, "Pacifier: High-throughput, reliable multicast without crying babies in wireless mesh networks," in *28th IEEE Conference on Computer Communications (INFOCOM). Rio de Janeiro, Brazil,*, April 2009.

[12] K. Srinivasan, M. Jain, J. Choi, T. Azim, E. Kim, P. Levis, and B. Krishnamachari, "The κ-factor: Inferring protocol performance using inter-link reception correlation,," in *ACM MobiCom. Chicago, IL,*, Sept 2010.

[13] L. Chen, T. Ho, S. H. Low, M. Chiang, and J. C. Doyle, "Optimization based rate control for multicast with network coding," in *Proc. of IEEE Conference on Computer Communications (INFOCOM), Anocharage, AK*, May 2007.

[14] Y. Wu and S.-Y. Kung, "Distributed utility maximization for network coding based multicasting: a shortest path approach," *IEEE J. on Selected Areas in Communications*, vol. 24, no. 8, pp. 1475–1488, Apr 2006.

[15] R. Dougherty, C. Freiling, and K. Zeger, "Insufficiency of linear coding in network information flow," *IEEE Trans. on Information Theory*, vol. 51, no. 8, pp. 2745–2759, 2005.

[16] A. Lehman and E. Lehman, "Complexity classification of network information flow problems," in *Proceedings of ACM-SIAM SODA*, Jan 2004.

[17] A. Khreishah, C.-C. Wang, and N. Shroff, "Cross-layer optimization for wireless multihop networks with pairwise intersession network coding," *IEEE Journal on Selected Areas in Communications*, vol. 27, no. 5, pp. 606–621, 2009.

[18] Y. Wu, P. Chou, Q. Zhang, K. Jain, W. Zhu, and S.-Y. Kung, "Network planning in wireless ad hoc networks: a cross-layer approach," *IEEE JSAC*, vol. 23, 2005.

[19] T. Bonald and L. Massoulie, "Impact of fairness on Internet performance," in *Proc. of ACM Joint International Conference on Measurement and Modeling of Computer Systems (Sigmetrics), Cambridge, MA*, June 2001.

[20] X. Lin and N. Shroff, "The impact of imperfect scheduling on cross-layer congestion control in wireless networks," *IEEE/ACM Trans. Networking*, vol. 14, no. 2, pp. 302–315, 2006.

[21] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE Trans on Automatic Control*, vol. 37, no. 12, Dec 1992.

[22] D. D. Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing,," in *ACM MobiCom. San Diego, CA,*, Sept 2003.

[23] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[24] D. Bertsekas and J. N. Tsitsikalis, *Parallel and Distributed Computation: Numerical Methods*. Athena Scientific, 1997.

[25] J. Park, M. Gerla, D. Lun, Y. Yunjung, and M. Medard, "Codecast: a network-coding-based ad hoc multicast protocol," *IEEE Wireless Communications*, vol. 13, no. 5, 2006.

[26] C.-C. Wang, "Pruning network coding traffic by network coding — a new class of max-flow algorithms," *IEEE Trans. on Info. Theory*, vol. 56, no. 4, pp. 1909–1929, 2010.

[27] ——, "On the capacity region of 1-hop intersession network coding – a broadcast packet erasure channel analysis with general message side information,," in *in the Proceedings of IEEE ISIT*, June 2010.

[28] C.-C. Wang, A. Khreishah, and N. Shroff, "On cross-layer optimizations for intersession network coding on practical 2-hop relay networks,," in *Asilomar Conference*, November 2009.