Energy Efficient Opportunistic Network Coding for Unicast Flows in Wireless Networks

Tao Cui, Lijun Chen, and Tracey Ho

***Abstract*— Due to the broadcast nature of wireless channels, a packet sent by one node may be "overheard" by several nearby nodes. Rather than taking these overheard packets as interference, multiple-reception of packet can be exploited to improve network performance. In this paper, we consider energy efficient network coding design in wireless networks with multiple unicast sessions. Our approach decomposes multiple unicast sessions into a superposition of multicast and unicast sessions, with coding occurring only within each session. We give an optimization-based approach that is more general than the existing poison-remedy optimization formulation. For the case of wireless, we consider XOR coding and give an achievable rate region for a primary interference model. To simplify network operation, we give an oblivious backpressure algorithm which may not utilize overhearing of transmissions optimally, and a practical protocol called COPR based on the oblivious backpressure algorithm. Simulation experiments show that COPR largely reduces network power consumption over existing algorithms.**

***Index Terms*— Network Coding, Energy efficiency, Wireless Networks, Opportunistic Routing, Optimization.**

I. INTRODUCTION

In this paper, we consider energy efficient cross-layer optimization for wireless networks by exploiting network coding and multiple-reception gain. A fundamental trait of wireless channels is that a node's transmissions can be possibly received by any nodes that lie within its communication range. Due to this trait, multiple-reception gain, including multiuser diversity gain [1] as a special case, may result in power saving. We focus on network coding across multiple unicast sessions, or intersession network coding. Optimal intersession network coding design is an open problem; various suboptimal algorithms have been proposed, see, e.g., [2], [3], [4], [5].

Our approach decomposes multiple unicast sessions into a superposition of multicast and unicast sessions, with coding occurring only within each session. For the case of wireless networks, we consider simple one-hop XOR coding as in COPE [2], where each node uses knowledge of what its neighbors have overheard to perform opportunistic network coding such that each encoded packet can be decoded immediately at the next hop. Reference [2] demonstrated substantial throughput gains for network coding that grow with the level of congestion. In this paper we consider the benefit of network coding for energy saving in power-constrained settings with less congestion.

To exploit multiple-reception gain, we model the network as a directed hypergraph. The achievable rate region of one-hop XOR coding is determined under a primary interference model. It is difficult and complicated to design dynamic scheduling and coding algorithms to achieve the entire rate region as it typically requires optimization over overheard flows. To simplify network operation, an *oblivious backpressure* algorithm is proposed which may not optimally utilize overheard flows. The link scheduling problem is found to be a maximum weighted hypergraph matching problem, which can be solved distributedly by using the algorithms in, e.g., [6]. To further reduce the complexity of session scheduling, the algorithm optimizes only over coding opportunities for packets at the head of queues at each node. By using the suboptimal scheduling algorithm, a fully distributed COPR protocol is proposed. Our simulation experiments show that COPR achieves up to 25% power saving over pure routing, showing that exploiting multiple-reception gain and network coding can enhance overall network performance substantially.

Our contributions can be summarized as follows:

- A new optimization-based approach is proposed for intersession network coding, based on decomposition into a superposition of multicast and unicast sessions with intrasession network coding. This formulation includes the poison-remedy approach of [3], [4], [5] as a special case.
- The achievable rate region of one-hop wireless XOR coding is determined under a primary interference model.
- An oblivious backpressure algorithm is proposed for dynamic scheduling and one-hop wireless XOR coding. Note that COPE does not have a specially designed session and link scheduling algorithm. Moreover, we also consider exploiting multiple-reception gain. The oblivious backpressure based scheduling can also be combined with fixed path routing as in COPE.
- A fully distributed protocol, COPR, is proposed. By using specially designed packets' format, the overhead of sending reception reports is reduced.

II. RELATED WORK

There exist extensive works on network coding and on backpressure techniques in networking. We mention here only a few that are most closely related to this work. Without network coding, joint congestion control, routing, and scheduling

This work has been supported in part by DARPA grant N66001-06-C-2020, Caltech's Lee Center for Advanced Networking, the Okawa Foundation Research Grant and a gift from Microsoft Research. This paper has been presented in part at the IEEE Conference on Computer Communications (Infocom), Phoenix, Arizona USA, April 2008.

The authors are with Engineering & Applied Science Division, California Institute of Technology, USA (Email: {taocui@, chen@cds., tho@}caltech.edu).

is studied in. e.g., [7]. The impact of imperfect scheduling on cross-layer design is studied in [8]. In [9], the network capacity region is characterized, and a joint routing and power allocation policy is proposed to stabilize any input rates within the capacity region. This approach is extended in [10] to consider energy efficiency, and in [11] to consider multiuser diversity. Opportunistic routing protocol is proposed in [12] which makes use of multiuser diversity.

With intrasession network coding, in [13], Lun *et. al.* propose a dual subgradient method for the problem of minimum cost multicasting. For rate control, the approach in [7] is extended to wireline networks in [14]. In [15], the rate stability region for a wireless network with and without correlated sources is characterized. Crosslayer design with broadcast advantage is considered in [6], where the scheduling problem is formulated as a hypergraph matching problem.

With intersession network coding, opportunistic XOR coding is proposed in [2]. Constructive XOR coding across pairs of unicasts is considered in [3] using a linear optimization approach. Dynamic backpressure is applied in [4], [5]. The work by Rayanchu et al. [16] considers throughput optimization and formulates a local coding optimization problem that is shown to be ♯p-hard, so a number of heuristic algorithms are given. The works [18], [19] focus on characterizing capacity in a two-hop topology with multiple unicast sessions communicating via a central relay node, whereas we consider energy-efficient optimization for multi-hop networks with general topologies. Different from the work [17], [18], [19] where throughput is considered, we study energy efficient cross layer optimization in this paper, which uses back pressure to make forwarding, coding and scheduling decisions.

III. Preliminaries

A. Network Model

Wireless networks are considered in this paper. As in [15], [20], [6], the network is modeled as a directed hypergraph $\mathcal{H} = (\mathcal{N}, \mathcal{A})$, where $\mathcal{N}$ is the set of nodes and $\mathcal{A}$ is the set of hyperarcs.[1] A hyperarc is a pair (i, J), with $i \in \mathcal{N}$ the start node and $J \subseteq \mathcal{N}$ the set of end nodes. Each hyperarc (i, J) represents a broadcast link from node i to nodes in J, which indicates that a packet transmitted by node i may be received by nodes in J due to the broadcast nature of the wireless channel. When J only contains a single node j, the hypergraph reduces to the conventional graph model used in, e.g., [7], [8]. A set $\mathcal{C}$ of unicast sessions or commodities is transmitted through the network, each indexed by $c \in \mathcal{C}$ with source s_c and receiver t_c and a flow rate of $\tilde{x}^c$. For each unicast session $c \in \mathcal{C}$, denote by f^c_{iJj} the flow rate over hyperarc (i, J) that is intended to node $j \in J$. By the flow conservation condition, we have

$$\sum_{\{J|(i,J)\in\mathcal{A}\}} \sum_{j\in J} f^c_{iJj} - \sum_{j\in\mathcal{N}} \sum_{\{I|(j,I)\in\mathcal{A},\, i\in I\}} f^c_{jIi} = \sigma^c_i,\ \forall i \in \mathcal{N}, \tag{1}$$

where

$$\sigma^c_i = \begin{cases} \tilde{x}^c, & \text{if } i = s_c \\ -\tilde{x}^c, & \text{if } i = t_c \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

In this paper we consider the primary interference model, where each node is equipped with only a single transceiver. Therefore, links that share a common node cannot be active simultaneously. If we further assume that nodes use orthogonal CDMA or FDMA, links that do not share nodes can transmit at the same time. Under this interference model, it is easy to see that any feasible link schedule corresponds to a hypergraph matching [6], where a hypergraph matching is defined as a set of hyperarcs with no pair incident to the same node. Let Λ denote the set of all hypergraph matchings with each hypergraph matching indexed by ξ. We represent a hypergraph matching as a $|\mathcal{A}|$-dimensional 0-1 vector α^ξ, with entry

$$\alpha^\xi_{iJ} = \begin{cases} 1, & \text{if } (i, J) \in \xi, \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

The set of all the feasible scheduling vectors is defined as the convex hull of these 0-1 scheduling vectors

$$\Pi = \left\{ \alpha \,\middle|\, \alpha = \sum_{\xi\in\Lambda} b_\xi \alpha^\xi,\, b_\xi \geq 0,\, \sum_{\xi\in\Lambda} b_\xi \leq 1 \right\}, \tag{4}$$

where $\alpha = \{\alpha_{iJ}\}_{(i,J)\in\mathcal{A}}$ with $\alpha_{iJ} = \sum_{\xi\in\Lambda} b_\xi \alpha^\xi_{iJ}$ denoting the frequency of (i, J) used in a scheduling.

Different from previous work [6], [7], [8], where lossless link or hyperarc is assumed and Shannon channel capacity is used, here we assume that each hyperarc is lossy, which means it experiences packet erasures, a more realistic assumption for practical wireless systems. Let R_{iJ} denote node i's transmission rate on hyperarc (i, J) and P_{iJ} denote the corresponding transmitting power. We assume that R_{iJ} is fixed, and P_{iJ} is fixed and equal to the total power P^{tot}_i at node i.[2] Under the primary interference model, the packet reception probability p_{iJj} at node j for a transmission on (i, J) is only determined by R_{iJ}, P_{iJ}, and the channel state h_{ij} from node i to node j, and is independent of other nodes' transmissions. Denote 2^J as the power set of J and η_{iS} as the probability that $S \subseteq J$ is exactly the set of nodes that successfully receive a packet transmitted by node i. The average capacity provisioned by successful transmissions from i to only and all nodes in S is then $\alpha_{iJ}\eta_{iS}R_{iJ}$. Denote by $f^{S,c}_{iJj}$ the flow rate of commodity c from i to j when $S \subseteq J$ is exactly the set of nodes that receive a packet transmitted by node i successfully; obviously, $f^{S,c}_{iJj} = 0$ when $j \notin S$. We have the following constraints

$$f^c_{iJj} = \sum_{\{S|S\in 2^J,\, j\in S\}} f^{S,c}_{iJj},\ \forall i,\, j \in J, \tag{5}$$

$$\sum_{c\in\mathcal{C}} \sum_{j\in S} f^{S,c}_{iJj} \leq \alpha_{iJ}\eta_{iS}R_{iJ},\ \forall S \in 2^J/\{\emptyset\}, \tag{6}$$

where $\emptyset$ denotes the empty set. The second constraint simply means that the aggregate flow rate for all commodities should

[1] Here we assume that the network topology is given. There is a network setup phase where each node leans of its neighbors and forms hyperarcs with them.

[2] Note that the proposed algorithm can be extended to the case that R_{iJ} is chosen from a set of discrete values corresponding to different modulations and P_{iJ} is chosen from a set of different power levels.

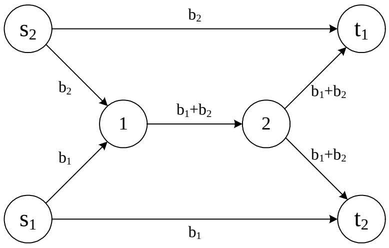

Fig. 1. The wireline butterfly network. Each link is of unit capacity. Two unicast sessions exist. s_i and t_i are source and receiver of session c_i, $i = 1, 2$.

not exceed the allocated capacity. Note that, even though all the nodes in S receive the packet, only one of such received (identical) packets will be forwarded and contribute to the flow to the receiver of commodity c. So, $\sum_{j\in S} f_{iJj}^{S,c}$ is the physical flow for commodity c.

B. Network Coding

In network coding, each node is allowed to perform algebraic operations on received packets. Network coding can be classified into intrasession network coding and intersession network coding. Intrasession network coding performs coding only across packets of the same session, while intersession network coding codes packets across different sessions. Intrasession network coding allows traffic for different sinks of a session to share network capacity. Random network coding can be used for intrasession coding to ensure fully distributed network operating algorithms.

Intersession network coding is still in its infancy. State of art approach usually uses poison-remedy flow scheme, see, e.g., [3], [4], [5]. For example, consider the wireline butterfly network in Fig. 1. If all the links have unit capacity, a flow rate of 1 for both sessions is not feasible with routing, yet feasible with intersession network coding as shown in Fig. 1. However, when the capacity of each link is arbitrary, it is not easy to determine the optimal intersession network coding strategy. In [3], [4], [5], whenever the flows from the two sessions are coded (or poisoned) at node 1, a remedy request is sent to s_1 (s_2), and s_1 (s_2) sends remedy packets along the link (s_1, t_2) ((s_2, t_1)) to facilitate decoding the encoded packets at t_2 (t_1). This approach does not allow coding over remedy packets, and it only allows coding over two sessions each time.

IV. Intersession Network Coding via Session Decomposition

In this section we describe a new optimization-based approach for intersession network coding that is also inspired by Fig. 1 but is more general than the poison-remedy formulation. Our algorithm is based on the observation that in the coding scheme in Fig. 1, source $s_1(s_2)$ actually multicasts $b_1(b_2)$ to both t_1 and t_2. Therefore, the two unicast sessions can be considered as a single multicast session with two sources s_1, s_2 and two receivers t_1, t_2. When the capacity of each link in Fig. 1 is arbitrary, we can consider that three sessions exist, where two unicast sessions u_i, $i = 1, 2$ share the same source and sink as c_i, and one multicast session m has two sources s_1, s_2 and two receivers t_1, t_2. The coding scheme in Fig. 1 then becomes intrasession coding within the multicast session m. Note that there is no intersession coding in the new formulation.

This approach decomposes multiple unicast sessions into a superposition of multicast and unicast sessions, and can be generalized to coding across any number of commodities on a general network. For simplicity of presentation, in the following we focus on pairwise decomposition where coding is explored and exploited across pairs of commodities, as an illustrative example to explain the approach. The source s_c of commodity $c \in \mathcal{C}$ partitions its exogenous packets into a unicast to receiver node t_c and $|\mathcal{C}| - 1$ multicast sessions each involving one other commodity. Each multicast session m involves two commodities c_1^m, c_2^m and two corresponding receiver nodes t_1^m and t_2^m. For $i = 1, 2$, t_i^m becomes an intermediate source for the commodity c_i^m packets it decodes, and discards packets that are not from c_i^m; like the original source it partitions its decoded commodity c_i^m packets into a unicast to receiver node $t_{c_i^m}$ and $|\mathcal{C}|-1$ multicast sessions each involving one other commodity. The commodity c packets that are unicast from the original and/or intermediate sources are treated as a single unicast session u_c. The packets from commodities c_1^m, c_2^m that are multicast from the original and/or intermediate sources to a pair of corresponding receiver nodes t_1^m and t_2^m are treated as a single multicast session m. Intrasession network coding is applied within each session.

The network capacity constraints and flow conservation for each session are the same as for the intrasession network coding problem, see, e.g., [13], [14], [15], and the techniques in these works can be generalized to this case by adding the following additional constraints on flow conservation across sessions, where x_i^{mc} and $x_i^{u_c}$ denote the flow rate of commodity c from (new or intermediate) source node i for multicast session m and unicast session u_c respectively:

$$\begin{aligned}&\sum_{\substack{\{m \mid t_1^m = i \text{ and } c = c_1^m \\ \text{or } t_2^m = i \text{ and } c = c_2^m\}}} \sum_{j\in\mathcal{N}} x_j^{mc} + I_{(t_c=i)} \sum_{j\in\mathcal{N}} x_j^{u_c} + I_{(s_c=i)} \tilde{x}^c \\ &= \sum_m x_i^{mc} + x_i^{u_c} + I_{(t_c=i)} \tilde{x}^c,\ \forall i \in \mathcal{N}, \forall c \in \mathcal{C},\end{aligned} \tag{7}$$

where $I_{(\cdot)}$ is the indicator function. The left hand side of equation (7) is the total flow rate of commodity c received at node i and the right hand side is the total flow of commodity c injected at node i for all the sessions.

For pairwise decomposition, there are $|\mathcal{C}|$ unicast sessions and up to $\binom{|\mathcal{C}|}{2}|\mathcal{N}|(|\mathcal{N}|-1)$ multicast sessions. We can generalize this approach by allowing a multicast session to involve a set $\bar{\mathcal{C}}$ of more than two commodities to be coded together, or more than two receiver nodes t_i^m. The commodities in $\bar{\mathcal{C}}$ are partitioned among the receiver nodes so that each commodity's packets are kept at only one receiver which becomes an intermediate source for those packets. It would incur high complexity if we want to explore and optimize over all the decomposition choices and coding opportunities. The complexity can, however, be traded off flexibly against performance by constraining the set of potential multicast receiver nodes, and the intermediate nodes may be chosen heuristically or randomly. For example, the algorithm proposed in section V considers the one-hop network coding where each coded packet is decoded at an immediate nexthop node.

Note that our formulation does not explicitly have remedy request and remedy flows. This allows more general coding of remedy flows with poison flows, and includes the XOR poison-remedy approach in [3], [4], [5] as a special case. Also there is no need for separate transmission of remedy requests.

V. Energy Efficient Opportunistic Backpressure Algorithms

A. Motivation and Key Idea

The algorithms proposed in [3], [4], [5] for intersession network coding have high complexity and overhead, while the experimental results reported in [3] show that the gain of intersession network coding with the poison-remedy approach is fairly modest in wireline networks compared with optimal routing. On the other hand, it is reported in [2] that simple network coding can achieve large gain in the total throughput in wireless networks. One fundamental difference between the setup in [2] and those in [3], [4], [5] is that the former uses the broadcast nature of wireless communication. All these motivate us to consider using "simple" intersession network codes to exploit the multiple-reception gain in wireless networks.

We consider the one-hop XOR intersession network coding strategy where each coded packet is decoded at the immediate nexthop node, which is similar to that in [2]. However, our approach differs from [2] in that we carefully design session scheduling policy while [2] simply apply round-robin packet scheduling at each node. Also we consider opportunistic reception of both coded and uncoded packets while [2] only considers uncoded packets. For example, in Fig. 2, the packet successful reception probabilities from node 3 to nodes 4-7 are labeled besides each link. Nodes 4 and 5 receive b_1 from node 1, nodes 6 and 7 receive b_2 from node 2, and node 3 receives both b_1 and b_2. By using the strategy in [2], node 3 sends an encoded packet b_1+b_2 to nodes 5 and 6, while our strategy also allows nodes 4-7 to receive the coded packet as all of them can decode the packet. The probability that at least one intended node receives the coded packet is 0.75 in the former case, while it increases to 0.9375 in the later case, which shows the potential benefit by using the proposed strategy. In our strategy, if both 4 and 5 receive the coded packet, a protocol is run to determine which one should keep the packet. Details of the protocol will be given in Section VI. When applying this strategy to a general wireless network, the key idea is to adopt the algorithm in Section IV by decomposing the network into superimposed wireless butterfly networks, each of which is similar to that in Fig. 2 centering around every node.

B. Opportunistic Unicast without Network Coding

We start from the simple case of unicast without network coding, which gives the intuition for the algorithm with network coding in Section V-C. We want to minimize the average power cost of the whole network. Assume that certain cost $\Omega_{iJ}(P_{iJ})$ is incurred when node i transmits at power P_{iJ} over hyperarc (i,J). The cost function $\Omega_{iJ}(\cdot)$ is usually

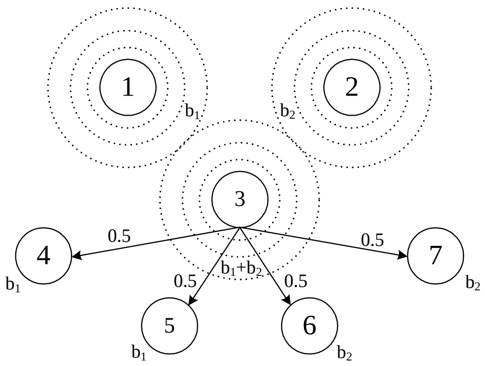

Fig. 2. Example of intersession coding with opportunistic reception. The packet successful reception probabilities from node 3 to nodes 4-7 are labeled besides each link. Nodes 4 and 5 receive b_1 from node 1, nodes 6 and 7 receive b_2 from node 2, and node 3 receives both b_1 and b_2.

convex, increasing, and with $\Omega_{iJ}(0)=0$. By equations (5)-(6), the percentage of time that hyperarc (i,J) is active is $\sum_{c\in\mathcal{C}}\sum_{j\in J}\frac{f^c_{iJj}}{(1-\eta_{i\emptyset})R_{iJ}}$. We can thus formulate the following joint opportunistic routing and scheduling problem so as to minimize the average power cost:

$$
\begin{aligned}
\min_{f,\alpha}\quad & \sum_{(i,J)\in\mathcal{A}}\sum_{c\in\mathcal{C}}\sum_{j\in J}\frac{f^c_{iJj}}{(1-\eta_{i\emptyset})R_{iJ}}\Omega_{iJ}\left(P^{\mathrm{tot}}_i\right)\\
\text{s.t.}\quad & \sum_{\{J|(i,J)\in\mathcal{A}\}}\sum_{j\in J}f^c_{iJj}-\sum_{j\in\mathcal{N}}\sum_{\{I|(j,I)\in\mathcal{A},\,i\in I\}}f^c_{jIi}=\sigma^c_i,\\
& f^c_{iJj}=\sum_{\{S|S\in 2^J,\,j\in S\}}f^{S,c}_{iJj},\ \forall i,\ j\in J,\\
& \sum_{c\in\mathcal{C}}\sum_{j\in S}f^{S,c}_{iJj}\le\alpha_{iJ}\eta_{iS}R_{iJ},\ \forall S\in 2^J/\{\emptyset\},\\
& \alpha\in\Pi,
\end{aligned}
\tag{8}
$$

where the first constraint comes from (1), the second and third constraints are equations (5) and (6), and the last one is the schedulability constraint.

Note that in real networks, whether a packet is received successfully by a node is a random process. Thus, solving (8) by using dual decomposition directly may not give a scheduling matched to current network state. However, the solution given by dual decomposition indeed sheds light on the optimal operation at each time slot. By relaxing only the first set of constraints in (8) and introducing Lagrange multipliers q^c_i at node i for commodity c, after simplification, we find that the Lagrangian dual of problem (8) is equivalent to solving

$$
\begin{aligned}
\max_{f,\alpha}\quad & \sum_{(i,J)}\sum_{c}\sum_{j\in J}f^c_{iJj}\left(q^c_i-q^c_j-\frac{\Omega_{iJ}(P^{\mathrm{tot}}_i)}{(1-\eta_{i\emptyset})R_{iJ}}\right)\\
\text{s.t.}\quad & f^c_{iJj}=\sum_{\{S|S\in 2^J,\,j\in S\}}f^{S,c}_{iJj},\ \forall i,\ j\in J,\\
& \sum_{c}\sum_{j\in S}f^{S,c}_{iJj}\le\alpha_{iJ}\eta_{iS}R_{iJ},\ \forall S\in 2^J/\{\emptyset\},\\
& \alpha\in\Pi.
\end{aligned}
\tag{9}
$$

By solving for f first, the above problem becomes

$$\begin{aligned} \max_{\alpha} \quad & \sum_{(i,J)} \alpha_{iJ}\nu_{iJ} \\ \text{s.t.} \quad & \alpha \in \Pi, \end{aligned} \tag{10}$$

with

$$\nu_{iJ} = R_{iJ} \max_{c} \sum_{S\in 2^J/\{\emptyset\}} \eta_{iS} b_i^{S,c}, \tag{11}$$

where

$$b_i^{S,c} = \max_{j\in S}\left[q_i^c - q_j^c - \frac{\Omega_{iJ}(P_i^{\text{tot}})}{(1-\eta_{i\emptyset})R_{iJ}}\right]^+ \tag{12}$$

with '+' denoting the projection onto the set $\mathbb{R}^+$ of nonnegative real numbers. Solving for f shows that, for given $q = \{q_i^c\}_{i\in\mathcal{N},c\in\mathcal{C}}$, node i will choose a commodity $c(q)$ that has maximal "weighted backpressure" $\max_c \sum_{S\in 2^J/\{\emptyset\}} \eta_{iS} b_i^{S,c}$ to transmit its packets; and the packets will be forwarded by the node $j(q)$ that has the maximal "backpressure" $\max_j(q_i^c - q_j^c)$ when the packets are received by only and all nodes in S; i.e.,

$$f_{iJj}^{S,c}(q) = \begin{cases} \alpha_{iJ}\eta_{iS}R_{iJ}, & \text{if } c = c(q) \ \&\ j = j(q), \\ 0, & \text{otherwise.}\end{cases} \tag{13}$$

Note that the maximization over f is a linear program. So, we can choose the above extreme point solution for f, which gives the aforementioned session scheduling and routing (i.e., opportunistic forwarding based on back-pressure).

The problem (10) is a maximum weighted hypergraph matching problem with ν_{iJ} being the weight of hyperarc (i, J). In [6], several distributed maximum weighted hypergraph matching algorithms are proposed, which can be applied to solve (10) directly.

The dual can be solved by the subgradient algorithm [21]. Denote by $q_i^c(k)$ the dual variable at the k-th iteration. It is updated according to the gradient algorithm as follows

$$\begin{aligned} q_i^c(k+1) \ = \ & [q_i^c(k) + \epsilon(\sigma_i^c - \sum_{(i,J)}\sum_{j\in J} f_{iJj}^c(k) \\ & + \sum_j \sum_{\{I|(j,I)\in\mathcal{A},\, i\in I\}} f_{jIi}^c(k))]^+. \end{aligned} \tag{14}$$

Note that here we require that the dual variable is nonnegative, since we can relax the first set of equality constraints to inequality constraints $\sum_{\{J|(i,J)\in\mathcal{A}\}}\sum_{j\in J} f_{iJj}^c - \sum_{j\in\mathcal{N}}\sum_{\{I|(j,I)\in\mathcal{A},\, i\in I\}} f_{jIi}^c \geq \sigma_i^c$ without affecting the optimal solution.

Remarks:

- The proposed algorithm can be readily modified for minimizing the power consumption per bit. In this case, the objective function is chosen as $\min_{f,\alpha}\sum_{(i,J)\in\mathcal{A}}\sum_{c\in\mathcal{C}}\sum_{j\in J}\frac{f_{iJj}^c}{(1-\eta_{i\emptyset})R_{iJ}\tilde{x}_c}\Omega_{iJ}\left(P_i^{\text{tot}}\right)$. The same dual decomposition approach applies to both cases. The only modification is that each packet needs to track the average power consumption during its transmission (for example add the power consumption into the header of each packet). The receiver needs to feed back the average power per packet periodically to the corresponding source.
- If both the power and the transmission rate of each node can be varied, $p_{iJj}(R_{iJ}, P_{iJ}, h_{ij})$ should be substituted into (8) and the same algorithm can be applied. If the function $p_{iJj}(\cdot)$ is known at each node, the power consumption can be minimized by varying R_{iJ} and P_{iJ}.

C. Opportunistic Unicast with XOR Intersession Coding

Motivated by the optimization decomposition in the last subsection, we propose a backpressure-based algorithm for joint scheduling and XOR intersession coding.

1) Backpressure algorithm: Each node maintains a queue for each commodity. The lengths of these queues are used to make coding, routing and scheduling decisions. Each node also maintains a *side information buffer* containing decoded packets obtained via transmissions or overhearing. Analogously to Section IV, we choose among possible uncoded transmissions (unicast) and coded transmissions (multicast) at each hyperarc (i, J).

Consider a hyperarc (i, J). Each packet p in the commodity c queue at node i is associated with a *side information set* O_p consisting of those neighbors of i whose side information buffers contain p. Suppose a set $\mathcal{M}$ of packets, each from a different commodity, is coded together and transmitted on (i, J). Then

$$\Gamma_{iJ}^{\mathcal{M}p} = \bigcap_{\{p'\in\mathcal{M}|p'\neq p\}} O_{p'} \cap J - O_p \tag{15}$$

is the set of nodes in J, excluding those in O_p, that can decode packet p if they receive the coded combination. A set $\mathcal{M}$ is a valid coding set (for multicast) iff $\Gamma_{iJ}^{\mathcal{M}p}$ is nonempty for all $p \in \mathcal{M}$.

We consider a time-slotted wireless network. Let $Q_i^c(t)$ denote the queue length of commodity c at node i at time (or timeslot) t. According to [7], [8], the dual variable q_i^c at time t can be written as $q_i^c(t) = \epsilon Q_i^c(t)$, where ϵ is a positive stepsize. The oblivious backpressure algorithm is described in detail in the following:

Algorithm 1: Oblivious backpressure algorithm for joint scheduling and XOR intersession coding

---

At time t:

• *Initialization:* Given $\{q(t)\}_{i\in\mathcal{N},c\in\mathcal{C}}$ at the beginning of timeslot t, at each node i, for each hyperarc (i, J), search through the queue of each commodity (or the head of line packets for each commodity) to find the valid coding sets $\mathcal{M}$. Let

$$b_i^{S,\mathcal{M}}(t) = \sum_{c\in\mathcal{C}_{\mathcal{M}}} \max_{j\in S\cap\Gamma_{iJ}^{\mathcal{M}c}}\left[q_i^c(t) - q_j^c(t) - \frac{\Omega_{iJ}(P_i^{\text{tot}})}{(1-\eta_{i\emptyset})R_{iJ}}\right]^+, \tag{16}$$

$$b_i^{S,u_c}(t) = \max_{j\in S}\left[q_i^c(t) - q_j^c(t) - \frac{\Omega_{iJ}(P_i^{\text{tot}})}{(1-\eta_{i\emptyset})R_{iJ}}\right]^+, \tag{17}$$

where $\mathcal{C}_{\mathcal{M}}$ denotes the set of commodities that have packets in the coding set $\mathcal{M}$. Let

$$\nu_{iJ}(t) = R_{iJ} \max \left\{ \max_{\mathcal{M}} \sum_{\substack{\{S|S\in 2^J, \\ S\cap\Gamma_{iJ}^{\mathcal{M}p}\neq\emptyset\}}} \eta_{iS} b_i^{S,\mathcal{M}}(t), \ \max_{u_c} \sum_{S\in 2^J} \eta_{iS} b_i^{S,u_c}(t) \right\}. \tag{18}$$

Denote $s_{iJ}(t)$ as the session attaining $\nu_{iJ}(t)$ (ties are broken randomly).

• *Link Scheduling:* A distributed hypergraph matching algorithm is executed to solve the scheduling problem (13) with the weights $\nu_{iJ}(t)$ obtained from the initialization step. If (i, J) is chosen by the matching algorithm or $\alpha_{iJ}(t) = 1$, node i becomes active.

• *Session Scheduling, Network Coding, and Data Transmission:* For each node i that is active, it decides to transmit and it then checks $s_{iJ}(t)$. If $s_{iJ}(t)$ is a unicast session u_c, node i simply transmits a packet from commodity c at rate R_{iJ}. If $s_{iJ}(t)$ corresponds to a coding set $\mathcal{M}$, a coded packet is formed by XOR-ing together the packets in $\mathcal{M}$. The coded packet is then sent at rate R_{iJ}.

• *Packet Reception and forwarding:* Upon receiving an uncoded packet from commodity c, for all nodes in J that receive the packet from node i, only the node j with the largest queue length difference $[q_i^c(t) - q_j^c(t)]^+$ puts the packet into its virtual queue corresponding to commodity c and the other nodes put the packet in their side information buffer. Moreover, the node keeping the packet tries to learn which nodes have also received this packet. Upon receiving a coded packet, for each commodity c in the coded packet, among all nodes in $\Gamma_{iJ}^{\mathcal{M}c}$, only the node j with the largest queue length difference $[q_i^c(t) - q_j^c(t)]^+$ decodes the packet using overheard packets in its side information buffer. A node drops the packet if the packet is not decoded for any commodity.

• *Queue Update:* In the end, each node updates its queue length $Q_i^c(t)$ and broadcasts it to its neighbors.

The intuition behind the above algorithm is obtained from the dual decomposition in Section V-B. This algorithm is oblivious to overhearing because it makes use of overheard packets whenever possible but it does not optimize over overheard flows as indicated by (16) and (17). Note that this algorithm only requires nodes to communicate with direct neighbors. Thus, it is a desired distributed algorithm.

In Algorithm 1, having each node check all the packets in the virtual queue of commodity c to get all the possible coding sets causes two problems in practice. First, packets may be reordered because the scheduling prefers to transmit packets which are overheard by many nodes rather than to transmit the head of the queue. Second, it is complicated to search through the queue. Therefore, in practice the algorithm only checks the head of each virtual queue for potential coding sets.

In the packet reception component of Algorithm 1, we have assumed that both the node with overheard packet and the node performing XOR coding receives the packet from a common one-hop neighbor. For example, in Fig. 2, both node 3 and node 4 receive b_1 from the common one hop neighbor node 1. In this case, overheard packets are from nodes one hop away from the node performing XOR coding. We call this overhearing scenario as *one-hop overhearing*. But, in Fig. 2, node 4 may overhear b_1 from another node 8 not from node 1. In this case, overheard packets are from nodes more than one hop away from the node performing XOR coding. We call this overhearing scenario as *multi-hop overhearing*. The derived scheduling policy can also be applied to multi-hop overhearing as long as each node knows that for each packet in its queue what other nodes have overheard this packet. We will discuss possible implementations for the proposed algorithms in Section VI.

2) Performance Analysis: To analyze the performance of our algorithm, we use a fluid model approach. For each hyperarc (i, J), we consider as part of a single multicast session m all coding sets $\mathcal{M}$ that involve a specific set $\mathcal{C}_m$ of commodities, where the packet p corresponding to commodity $c \in \mathcal{C}_m$ has a specific side information set,[3] denoted O_c. A multicast session m is thus defined by $\mathcal{C}_m$ and the set O_c for each $c \in \mathcal{C}_m$. Denote by $\mathfrak{M}_i$ the set of all such multicast sessions at node i.

As in (6), when node i only transmits packets from unicast session u_c over hyperarc (i, J), information flow f_{iJj}^{S,u_c} will also be the physical flow that takes up the capacity of (i, J). When node i only transmits packets from multicast session m over (i, J), we can introduce a physical flow $g_{iJj}^{S,m}$ that takes up the capacity of (i, J), and information flow $f_{iJj}^{S,mc}$ satisfies

$$f_{iJj}^{S,mc} = g_{iJj}^{S,m}, \ \forall c \in \mathcal{C}_m, \tag{19}$$

since packets of different commodities will be coded together and then transmitted. The information and physical flows then satisfy the following relations

$$f_{iJj}^{c} = f_{iJj}^{u_c} + \sum_{m\in\mathfrak{M}_i} f_{iJj}^{mc}, \forall i, j \in J, \tag{20}$$

$$f_{iJj}^{u_c} = \sum_{\{S|S\in 2^J, j\in S\}} f_{iJj}^{S,u_c}, \forall i, j \in J, \tag{21}$$

$$f_{iJj}^{mc} = \sum_{\{S|S\in 2^J, j\in S\cap\Gamma_{iJ}^{mc}\}} f_{iJj}^{S,mc}, \forall i, j \in J, \tag{22}$$

$$\sum_{u_c}\sum_{j\in S} f_{iJj}^{S,c} + \sum_{m\in\mathfrak{M}_i}\sum_{j\in S\cap\Gamma_{iJ}^{mc}} g_{iJj}^{S,m} \le \alpha_{iJ}\eta_{iS}R_{iJ}, \forall S \in 2^J/\{\emptyset\}. \tag{23}$$

For given coding opportunities, i.e, multicast sets $\mathfrak{M}_i$ at each node i, denote by $\Xi(\mathfrak{M})$ the achievable rate region defined by equations (19)-(23) and $\alpha \in \Pi$. When there is no coding opportunity, i.e., $\mathfrak{M} = \emptyset$, $\Xi(\emptyset)$ reduces to the achievable rate region defined in problem (8). Obviously, $\Xi(\emptyset) \subseteq \Xi(\mathfrak{M})$ for any $\mathfrak{M}$, and $\Xi(\mathfrak{M})$ has an enlarged capacity region at link layer because of the coding opportunity that is enabled by wireless broadcast and multiple receptions.

At runtime t of Algorithm 1, based on the coding opportu-

[3] Recall that the side information set of p is defined as the neighbors of i whose side information buffers contain p.

nities $\mathfrak{M}(t)$, we can define an energy minimization problem:

$$\begin{aligned}\min_{f,\alpha} \quad & \sum_{(i,J)\in\mathcal{A}} \sum_{c\in\mathcal{C}} \sum_{j\in J} \frac{f^c_{iJj}}{(1-\eta_{i\emptyset})R_{iJ}} \Omega_{iJ}\left(P_i^{\rm tot}\right) \\ \text{s.t.} \quad & (1), \\ & (f,\alpha) \in \Xi(\mathfrak{M}(t)).\end{aligned} \tag{24}$$

Note that these coding opportunities are not given a priori, but are dependent on the algorithm. Analogous to the mathematical manipulations in Subsection V-B, for each at time t, Algorithm 1 can be seen as motivated by the dual decomposition of the above problem. This is actually the insight behind Algorithm 1.

Based on the coding opportunities $\mathfrak{M}(t)$ at each time t in the running of the algorithm, we can define a time-average achievable rate region

$$\begin{aligned}\bar{\Xi} = \{(\bar{f},\bar{\alpha}) \quad : \quad & (\bar{f},\bar{\alpha}) = \lim_{t\to\infty} \frac{1}{t}\sum_{\tau=1}^{t} (\bar{f}(t),\bar{\alpha}(t)), \\ & (\bar{f}(t),\bar{\alpha}(t)) \in \Xi(\mathfrak{M}(t))\}.\end{aligned} \tag{25}$$

Obviously, $\Xi(\emptyset) \subseteq \bar{\Xi}$. If the coding opportunities $\mathfrak{M}(t)$ converge to certain stationary process,[4] denote by $h(\mathfrak{M})$ the corresponding probability when there are coding opportunities $\mathfrak{M}$. The above time-average becomes an ensemble average

$$\begin{aligned}\bar{\Xi} = \{(\bar{f},\bar{\alpha}) \quad : \quad & (\bar{f},\bar{\alpha}) = \sum_{\mathfrak{M}} h(\mathfrak{M})(\bar{f}(\mathfrak{M}),\bar{\alpha}(\mathfrak{M})), \\ & (\bar{f}(\mathfrak{M}),\bar{\alpha}(\mathfrak{M})) \in \Xi(\mathfrak{M})\}.\end{aligned} \tag{26}$$

For any particular realization of coding opportunities $\mathfrak{M}(t)$ at each time t, we would ideally want our algorithm to solve the following energy minimization problem

$$\begin{aligned}\min_{f,\alpha} \quad & \sum_{(i,J)\in\mathcal{A}} \sum_{c\in\mathcal{C}} \sum_{j\in J} \frac{f^c_{iJj}}{(1-\eta_{i\emptyset})R_{iJ}} \Omega_{iJ}\left(P_i^{\rm tot}\right) \\ \text{s.t.} \quad & (1), \\ & (f,\alpha) \in \bar{\Xi}.\end{aligned} \tag{27}$$

Let $C^*_{\rm XOR}$ be the optimal cost of (27), and $C^*_{\rm R}$ be the optimal cost of (8) without network coding. Since $\Xi(\emptyset) \subseteq \bar{\Xi}$, we have $C^*_{\rm R} \leq C^*_{\rm XOR}$.

The achievable rate region $\bar{\Xi}$ is the best possible under the particular coding opportunities. Algorithm 1 cannot be derived from the dual decomposition of problem (27). However, we will use the problem (27) as a reference system, and characterize the performance of Algorithm 1 with respect to it.

Denote by $f^{u_c}_{iJj}(t)$ and $f^{mc}_{iJj}(t)$ the unicast and multicast flows that are successfully transmitted at time t under Algorithm 1. Note that packet reception is a random process. Even though for given $q(t)$ link/session scheduling, network coding and data transmission are all deterministic, $f^{u_c}_{iJj}(t)$ and $f^{mc}_{iJj}(t)$ are random variables. Define

$$\begin{aligned}\rho_i^c(t) \;=\; & \sigma_i^c - \sum_{(i,J)}\sum_{j\in J}\left(\sum_{m\in\mathfrak{M}_i(t)} f^{mc}_{iJj}(t) + f^{u_c}_{iJj}(t)\right) \\ & + \sum_j \sum_{\{I|(j,I)\in\mathcal{A},\, i\in I\}} \left(\sum_{m\in\mathfrak{M}_i(t)} f^{mc}_{jIi}(t) + f^{u_c}_{jIi}(t)\right).\end{aligned} \tag{28}$$

The queue length will evolve as $Q_i^c(t+1) = [Q_i^c(t)+\rho_i^c(t)]^+$, and the "dual" variable $q_i^c(t) = \epsilon Q_i^c(t)$ will evolve as $q_i^c(t+1) = [q_i^c(t) + \epsilon\rho_i^c(t)]^+$. Let $\bar{\rho}(t) = \{\bar{\rho}_i^c(t)\}_{i\in\mathcal{N},c\in\mathcal{C}}$, where $\bar{\rho}_i^c(t)$ denotes the mean of $\rho_i^c(t)$. $\bar{\rho}(t)$ is the dual subgradient of problem (24) when we relax the flow conservation condition (1) as in Section V-B. If the norm of the subgradients is uniformly bounded, i.e., there exists G such that $\|\bar{\rho}(t)\|_2 \leq G$ for all t, we have the following result on the performance of Algorithm 1.

Theorem 1: Let $C(t)$ be the total power cost of the network at time t by using Algorithm 1. We have the following inequality

$$C^*_{\rm XOR} \leq \limsup_{t\to\infty} \frac{1}{t}\sum_{\tau=1}^{t} E\{C(\tau)\} \leq C^*_{\rm XOR} + \frac{\epsilon G^2}{2}. \tag{29}$$

Proof: The first inequality is obvious, as the objective function is linear, and the average flow is in the achievable rate region $\bar{\Xi}$. For the seond inequality, note that

$$\begin{aligned}& E\{\|q(t+1)\|_2^2|q(t)\} \leq E\{\|q(t)+\epsilon\rho(t)\|_2^2|q(t)\} \\ =& E\{\|q(t)\|_2^2 + 2\epsilon\rho^T(t)q(t) + \epsilon^2\|\rho(t)\|_2^2|q(t)\} \\ =& E\{\|q(t)\|_2^2 - 2\epsilon C(t) + 2\epsilon C(t) + 2\epsilon\rho^T(t)q(t) + \epsilon^2\|\rho(t)\|_2^2|q(t)\} \\ \leq& \|q(t)\|_2^2 - 2\epsilon E\{C(t)\} + 2\epsilon C_t^* + \epsilon^2 G^2,\end{aligned}$$

where C_t^* denotes the optimum of problem (24) with the achievable rate region enabled by the coding opportunities $\mathfrak{M}(t)$, and the last inequality has used the property for convex minimization that the dual function is no more than the primal optimum, i.e., $E\{C(t)\}+\bar{\rho}^T(t)q(t) \leq C_t^*$. Applying the above equation recursively, we obtain

$$\|q(t+1)\|_2^2 \leq \|q(1)\|_2^2 - 2\epsilon\sum_{\tau=1}^{t}\left(E\{C(\tau)\} - C_\tau^*\right) + t\epsilon^2 G^2.$$

Since $\|q(t+1)\|_2^2 \geq 0$, we obtain

$$\frac{1}{t}\sum_{\tau=1}^{t}(E\{C(\tau)\} - C_\tau^*) \leq \frac{|q(1)\|_2^2 + t\epsilon^2 G^2}{2t\epsilon}. \tag{30}$$

Again, since the objective function is linear, we have $\lim_{t\to\infty}\frac{1}{t}\sum_{\tau=1}^{t} C_\tau^* = C^*_{\rm XOR}$. By taking $t\to\infty$ of equation (30), we obtain the second inequality. ∎

We see that the performance of Algorithm 1 approaches the optimum of the ideal reference system (27) arbitrarily close, when the stepsize ϵ is small enough. If there are coding opportunities and thus a gap between coding optimum $C^*_{\rm XOR}$ and routing optimum $C^*_{\rm R}$, the performance of Algorithm 1 is better than the routing optimum. This means that, even though Algorithm 1 is an oblivious algorithm and does not optimize over overheard packets, it is guaranteed to achieve better performance than the optimal routing scheme.

[4]This may not necessarily happen, but intuitively we might expect approximate convergence in many cases of interest; e.g., in a dense wireless network with a large number of commodities.

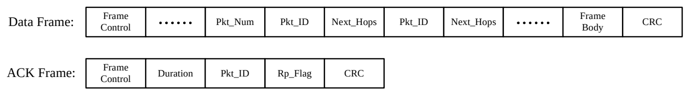

Fig. 3. Data and ACK frame formats in COPR.

Remarks:

- If we apply Algorithm 1 to the downlink of a cellular network where a base station transmits data to K mobile users and assume all the users have the same queue length at base station all the time, it reduces to the scheduling algorithm in [1], which allows only the user with the best channel to transmit at any time.
- Note that algorithms proposed in this section can also be applied to applications with predetermined routes, such as shortest path.

VI. COPR Protocol Design

In this section, we consider protocol design by using the opportunistic backpressure algorithms proposed in Section V-C in synchronous slotted wireless networks using CDMA or FDMA. A protocol called Coding with Opportunistic Reception (COPR) is developed. We briefly outline the features of COPR. Details of COPR can be found in [22]. We first consider COPR with one-hop overhearing. The medium access control (MAC) layer of COPR runs on top of 802.11b MAC. Each time-slot is partitioned into three phases: contention period, data transmission, and packet acknowledgement.

Contention Period: At the beginning of each time slot, one of distributed hypergraph matching algorithms in [6] is executed during the contention period. 802.11b MAC is used to resolve contention during hypergraph matching.

Data Transmission: Data transmission, session scheduling, and network coding follow Algorithm 1. Data frame format is depicted in Fig. 3, which follows that in 802.11 standards. New `Pkt_Num`, `Pkt_ID`, and `Next_Hops` fields are added before the frame body, where `Pkt_Num` indicates the number of native packets XOR-ed together, `Pkt_ID` is the ID of one of the native packets, and `Next_Hops` includes the MAC addresses of all possible next hop nodes corresponding to the native packet with `Pkt_ID`. The addresses in `Next_Hops` are in decreasing order of queue difference. The maximum number of next hop nodes for each `Pkt_ID` is denoted as `max_next_num`.

Packet Acknowledgement: COPR reserves multiple acknowledgement slots. If a node hears a packet and it checks that its address is the j-th address in `Next_Hops` corresponding to the i-th `Pkt_ID`, it decodes the coded packet to obtain `Pkt_ID` and then waits for $\texttt{SIFS}+((i-1)\cdot\texttt{max\_next\_num}+j-1)\cdot(\texttt{ack\_tx\_time}+\texttt{SIFS})$ before sending its acknowledgement, where `ack_tx_time` is the time to transmit an ACK packet and SIFS denotes the short interframe space in 802.11. The ACK frame format is shown in Fig. 3. A new `Rp_Flag` field of **max_next_num** bits is added before the CRC. If the j-th bit in `Rp_Flag` is 1, it indicates that the j-th node in `Next_Hops` also receives `Pkt_ID`. Each node maintains a vector $\underline{r}$ of length `max_next_num`. At the beginning of each time slot, it sets $\underline{r}$ as an all zero vector. If it receives a packet and its address is the j-th address in `Next_Hops` corresponding to the `Pkt_ID`, it sets the j-th entry of $\underline{r}$, r_j to be 1. During packet acknowledgement period, whenever it overhears an ACK packet, it checks whether the `Pkt_ID` of this ACK is equal to the `Pkt_ID` it receives. If yes, it sets r_i to 1 if the i-th entry of `Rp_Flag` in the ACK is 1 (simply OR `Rp_Flag` and $\underline{r}$). At end of each time slot, if $r_i = 1$, it means that the i-th node in `Next_Hops` has also received the `Pkt_ID`. To get a tradeoff between performance improvement and overhead of sending ACK, in [22], we find the optimal value of `max_next_num` is 3 by using 802.11b parameters in fast Rayleigh fading channels. COPR also allows nodes that are not added into the hypergraph matching to overhear packets. In this case, reception report still needs to send as COPE. But from experiments, we find this case rarely happens.

Multiple-hop overhearing may be exploited to improve the network performance. We create a new field, Overheard_Node, in the header of a packet and put into the addresses of nodes that have overheard this packet. To reduce the overhead of this approach, we only keep those nodes that are K hops away from current node, or the number of overhearing node addresses is set to be $K\cdot\texttt{max\_next\_num}$. A tradeoff between overhead and performance can be attained through K.

VII. Experimental Results

In this section, we study the performance of different algorithms via packet level simulation. In all experiments, all nodes use identical 802.11b cards based on at Conexant (formerly Intersil) PRISM 2.5 chip-set. The cards are configured to transmit at 1 Mbit/s using BPSK modulation. The transmission power of each node is set to be 23 dBm. Each packet is of length 1 KB. Different algorithms are compared using average power consumption per bit. Each node starts with enough energy so that it will not run out of its energy during the simulations. Only power consumption during data transmission and ACK transmission is taken into account. To isolate the impact of contention from network coding/routing, power consumption during contention period is not counted. The link delivery rate is assumed to be known at each node. We compare 4 different kinds of algorithms. The first one, called COPE_bp, employs the COPE opportunistic network coding algorithm with session scheduling performed by using the backpressure algorithm derived in Appendix, which is equivalent to setting `max_next_num`=1 in COPR. As this algorithm is different from COPE in [2], it is denoted as COPE_bp. The second one, COPR, performs session scheduling by searching only the head of queue[5]. The third one, opportunistic unicast without network coding, is denoted as ORouting (Note that this is different from the ExOR in [12]). The last one is simply shortest path routing based on backpressure weight $q_i^c - q_j^c$ for each link (i, j) for fair comparison, where q_i is the queue length for commodity c at node i.

[5]We do not compare with COPR by searching through the queue at each node as it is very complicated for a large network with many sessions. But from our simulation with 5 nodes wireless butterfly network, we indeed find that this scheme can improve performance especially in a very lossy network.

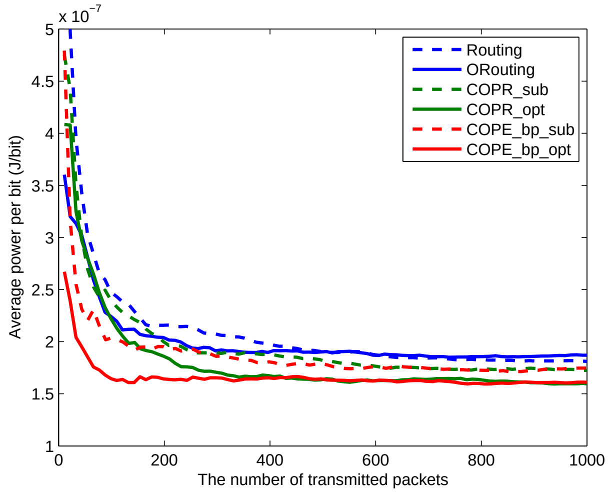

Fig. 4. Evolution of average power consumption per bit as the number of packets transmitted in the butterfly network when $D = 30$ m.

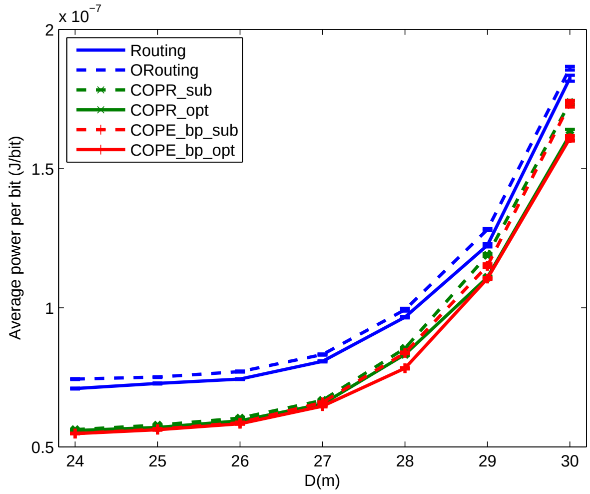

Fig. 5. Average power consumption per bit versus D of different algorithms in the butterfly network.

We first consider the wireless butterfly network with two sources s_1, s_2, two sinks t_1, t_2 and one relay node r, where the coordinates of s_1, s_2, t_1, t_2, r are $(0, 0)$, $(\sqrt{3}D, 0)$, $(\sqrt{3}D, D)$, $(0, D)$, and $(\frac{\sqrt{3}}{2}D, \frac{1}{2}D)$. Two unicast sessions exist in the network, where s_i and t_i are the source and the sink of unicast session i, $i = 1, 2$. The transmission power of each node is set to be 15 dBm. Two-ray ground model is used as propagation model. The relationship between BER and SNR is obtained from Intersil HFA3863 specification [24]. In COPR, an integer programming (IP) is solved to find the maximum weighted hypergraph matching. We choose $\epsilon = 0.005$.

Fig. 4 shows the evolution of average power consumption per bit as the number of packets transmitted in the butterfly network when $D = 30$ m. The rate of both sources is 87 Kbit/s. We can see that each algorithm oscillates around an equilibrium point after 400 packets are received. Fig. 5 compares different algorithms by varying D. The average power per bit is obtained after transmitting 1000 packets.

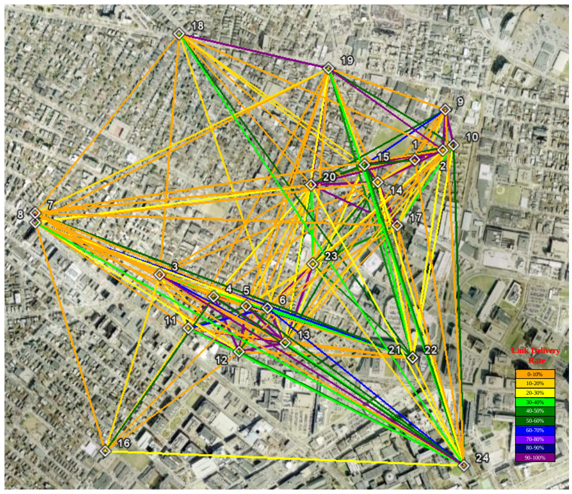

Fig. 6. Connectivity map of Roofnet in August 2004, with a diamond icon for each of the 24 nodes that participated in the experiments in this section (the .kml file of this map for Google earth is available online at `http://www.its.caltech.edu/~taocui/roof_net.kml`).

The rate of both sources is $225 \cdot p$ Kbit/s, where p is the delivery rate between s_1 and r. We can see that ORouting does not have any improvement over Routing, and due to overhead of sending ACK the former consumes even mover power than the latter. COPE and COPR perform similarly. When $D = 24$, both COPE and COPR achieve a 30% power saving over Routing, which agrees with theoretical analysis. Also COPE_sub performs close to COPE_opt and the same is true for COPR. As D increases, the power saving achieved by COPR and COPE reduces. This is because when r may not be able to XOR the packets from the two sessions together as the delivery rate of link (s_1, t_2) (or (s_2, t_1)) decreases when D increases and t_1 (or t_2) may not overhear the packet from session 2 (or session 1). Moreover, COPR_opt outperforms COPR_sub as the probability that there exists two packets in both queues can be XOR-ed together is not less than the probability that the packets in the head of two queues can be XOR-ed together, and when D increases the former is greater than the latter. This suggests that in a lossy network it is better to search through the queues for better coding opportunity.

Next, we study the performance of different algorithms via packet level simulation on Roofnet [25], whose connectivity map in August 2004 is shown in Fig. 6. Even though Roofnet is not designed for energy efficient communication, we use this network to evaluate different algorithms because of the availability of experimental data on link loss rates in [25], which is labeled with different colors in Fig. 6.

As the network is large, it is difficult to find the maximum weighted hypergraph matching. Instead, in COPR, we use distributed greedy hypergraph matching algorithms in [6]. COPR_gdy1 chooses a locally heaviest hyperarc every time, which COPR_gdy2 chooses a locally heaviest hyperarc discounted by the size of the hyperarc every time. COPR with multi-hop overhearing is denoted as COPR_multi, which keeps in each packet the nodes overheard this packet in previous 3 hops. We choose $\epsilon = 0.025$ in all algorithms. U unicast sessions exist in the network, where each session picks sender

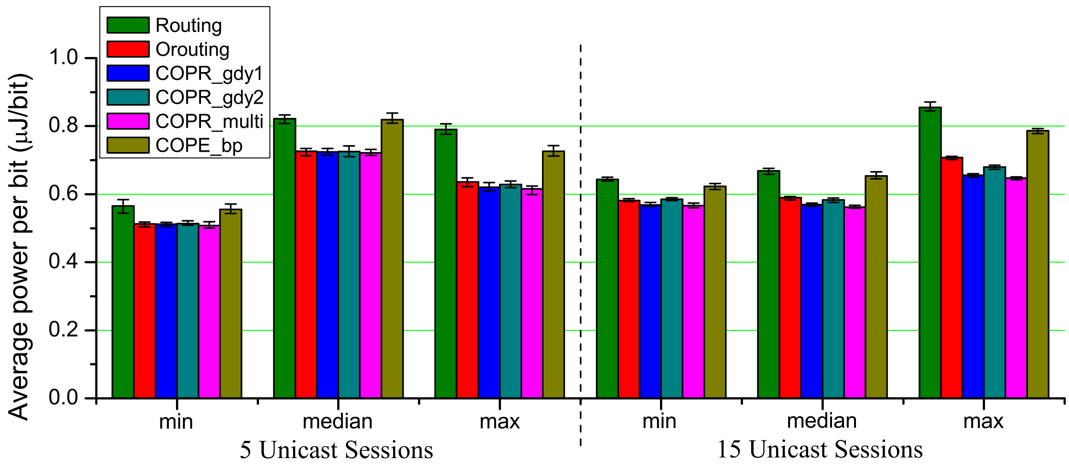

Fig. 7. Average power consumption per bit of different algorithms in Roofnet with 5 unicast sessions and 15 unicast sessions.

and receiver randomly. For each U, 20 realizations of unicast sessions are generated. A 1 megabyte file is split evenly among all the U sessions and each session transmits one piece. The average power per bit is obtained after transferring the file over UDP, which is insensitive to packet losses and packet reordering. As backpressure based algorithms usually need to learn efficient routes first, we first transmit another 1 megabyte file before transmitting the test file. The realizations with the maximum, the minimum, and the median power saving by using COPR_gdy1 over that by using Routing are used to compare different algorithms.

Fig. 7 shows performance of different algorithms on Roofnet with 5 and 15 unicast sessions. For 5 unicast sessions, on average, ORouting achieves an 11.45% power saving over Routing for all 20 realizations, while this number increases to 12.59% by using COPR_gdy1. Interestingly, COPR_gdy2 performs worse than COPR_gdy1, which shows that energy efficient applications prefer large hyperarcs. By using COPR_multi, an additional 1.24% power saving can be attained. In the maximum power saving case, the average percentage of encoded packets at all nodes is 12.52% using COPR_gdy1 and is 13.53% with COPR_multi. COPE_bp only achieves a 1.43% power saving over Routing on average. In the maximum power saving case, on average, 9.11% packets are coded with COPE_bp. Similar results hold for 15 unicast sessions. The power saving by using ORouting and COPR_gdy1 over that by Routing becomes to 10.08% and 13.60%, respectively. By using COPR_multi, the gain increases to 14.59%. COPE_bp also attains a 2.27% gain. In the maximum power saving case, 28.94%, 30.49%, and 8.40% packets are coded together with COPR_gdy1, COPR_multi, and COPE_bp, respectively, and the power savings are 23.40%, 24.37%, and 8.11%, respectively. By increasing the number of sessions from 5 to 15, we find that the percentage of packets coded together increases from 7.67% to 19.98%. Therefore, increasing the number of sessions creates more coding opportunities. We do not find any 3 packets are XORed together in all 5-session realizations while only 0.02% of coded packets are XORed with 3 packets in all 15-session realizations. Opportunistic network coding with shortest path routing does not find many coding opportunities partly because the number of sessions is small in our simulation. In [2], a large number of sessions exist in a network results in high congestion and many coding opportunities. On the other hand, the proposed algorithm still works even though the number of sessions is small.

Our experimental results suggest the following:

- Backpressure based algorithms create more coding opportunities than shortest path based algorithms in low congestion and energy-constrained settings;
- Coding gain increases with the number of sessions.
- Multi-hop overhearing seems to be not helpful. It is good enough to use one-hop overhearing with reduced overhead.

VIII. Conclusion

In this paper, we have investigated energy efficient backpressure algorithms by exploiting multiple reception gain in wireless networks. Based on optimization framework, backpressure algorithms are proposed for unicast without network coding and XOR intersession network coding. Link scheduling problem is found to be a maximum weighted hypergraph matching problem, which can be solved distributedly by using the algorithms in [6]. The optimal session scheduling algorithm requires searching through all the queues at each node. To reduce the complexity of session scheduling, a suboptimal algorithm is proposed to search only the head of queues at each node. By using proposed algorithms, a COPR protocol is proposed for unicast with XOR intersession network coding. COPR uses a specially designed MAC, which runs on top of

802.11b MAC. Packets' format and parameters' settings are also discussed for COPR. Our experimental results show that COPR achieves up to 25% power saving over pure routing. As a future work, it is interesting to develop low complexity algorithms for computing (16) and (17) in session scheduling.

Appendix: Oblivious Backpressure for COPE

In [2], COPE is proposed without giving the optimal scheduling policy and without considering multiple-reception gain. In this appendix, we derive oblivious backpressure for COPE. The problem formulation is similar to that in Section V-C. The only difference is the way they make use of wireless channel capacity. Without opportunistic reception, we replace (23) with the following constraints

$$\sum_{j\in\Gamma_{iJ}^{mc}} \frac{f_{iJj}^{mc}}{p_{iJj}} + \sum_{j\in J} \frac{f_{iJj}^{u_c}}{p_{iJj}} \le \alpha_{iJ} R_{iJ}, \tag{31}$$

where p_{iJj} is the packet successful reception probability from node i to node j over (i, J). By replacing the corresponding constraint in problem (24) with (31), we obtain the joint opportunistic network coding and scheduling problem as (24), which motivates an oblivious algorithm as follows. Let

$$\bar{\nu}_{iJ}^{m} = R_{iJ} \sum_{c} \max_{j\in\Gamma_{iJ}^{mc}} p_{iJj} \left[q_i^c - q_j^c - \frac{\Omega_{iJ}(P_i^{\text{tot}})}{(1-\eta_{i\emptyset})R_{iJ}} \right]^+, \tag{32}$$

$$(j^{mc})^* = \arg\max_{j\in\Gamma_{iJ}^{mc}} p_{iJj} \left[q_i^c - q_j^c - \frac{\Omega_{iJ}(P_i^{\text{tot}})}{(1-\eta_{i\emptyset})R_{iJ}} \right]^+, \tag{33}$$

$$\bar{\nu}_{iJ}^{u_c} = R_{iJ} \max_{j\in S} p_{iJj} \left[q_i^c - q_j^c - \frac{\Omega_{iJ}(P_i^{\text{tot}})}{(1-\eta_{i\emptyset})R_{iJ}} \right]^+, \tag{34}$$

$$(j^{u_c})^* = \arg\max_{j\in S} p_{iJj} \left[q_i^c - q_j^c - \frac{\Omega_{iJ}(P_i^{\text{tot}})}{(1-\eta_{i\emptyset})R_{iJ}} \right]^+. \tag{35}$$

Let $\nu_{iJ} = \max\{\max_m \bar{\nu}_{iJ}^m, \max_{u_c} \bar{\nu}_{iJ}^{u_c}\}$, and denote s_{iJ}^* as the session attained ν_{iJ} (ties are broken randomly). The link scheduling is similar to that in Section V-C. The session scheduling policy is summarized as follows:

Session Scheduling, Network Coding, and Data Transmission: For each node i that is active, it decides to transmit and it then checks the session with the maximum ν_{iJ}. If s_{iJ}^* is a unicast session, node i simply transmits a packet from commodity c at rate R_{iJ} to node $(j^{u_c})^*$ defined in (35). If s_{iJ}^* is a multicast session, the queue of commodity c is checked (not only the head of each queue is checked) to get all the possible O_c assuming that each packet knows the set of nodes that have overheard this packet, and ν_{iJ} is computed according to (32) and (34). A coded packet is formed by XOR-ing together one packet from each commodity c with overhearing set O_c. The coded packet is then sent at rate R_{iJ} and is intended to $(j^{mc})^*$ defined in (33) for each commodity c.

No special action is taken for packet reception. Each intended node only sends an acknowledgement after receiving a packet. Queue update can also be performed similarly as Algorithm 1. Each node maintains $|\mathcal{C}|$ virtual queues.

As commented in Section VI, optimal scheduling algorithm may reorder packets and increase the complexity. A suboptimal and practical approach is to consider only the head of each virtual queue.

References

[1] R. Knopp and P. A. Humblet, "Information capacity and power control in single-cell multiuser communications," in *Proc. of IEEE ICC*, June 1995, pp. 331–335.

[2] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, "XORs in the air: Practical wireless network coding," in *Proc. of ACM SIGCOMM*, Oct. 2006, pp. 243–254.

[3] D. Traskov, N. Ratnakar, D. S. Lun, R. Koetter, and M. Médard, "Network coding for multiple unicasts: An approach based on linear optimization," in *Proc. of IEEE ISIT*, July 2006, pp. 1758–1762.

[4] T. Ho, Y. Chang, and K. J. Han, "On constructive network coding for multiple unicasts," in *Proc. of Allerton Conf. on Comm., Contr. and Comput.*, Sept. 2006.

[5] A. Eryilmaz and D. S. Lun, "Control for inter-session network coding," in *Proc. of the Workshop on Network Coding, Theory and Applications*, 2007.

[6] T. Cui, L. Chen, and T. Ho, "Distributed optimization in wireless networks using broadcast advantage," in *Proc. of IEEE CDC*, Dec. 2007.

[7] L. Chen, S. H. Low, M. Chiang, and J. C. Doyle, "Cross-layer congestion control, routing and scheduling design in ad hoc wireless networks," in *Proc. of IEEE Infocom*, Apr. 2006.

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

[9] M. Neely, E. Modiano, and C. Rohrs, "Dynamic power allocation and routing for time-varying wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 1, pp. 89–103, Jan. 2005.

[10] M. Neely, "Energy optimal control for time varying networks," in *Proc. IEEE Infocom*, Apr. 2005.

[11] ——, "Optimal backpressure routing for wireless networks with multi-receiver diversity," in *Proc. of CISS*, Mar. 2006, pp. 18–25.

[12] S. Biswas and R. Morris, "ExOR: Opportunistic routing in multi-hop wireless networks," in *Proc. of ACM SIGCOMM*, Aug. 2005, pp. 133–144.

[13] D. S. Lun, N. Ratnakar, M. Médard, R. Koetter, D. R. Karger, T. Ho, E. Ahmed, and F. Zhao, "Minimum-cost multicast over coded packet networks," *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2608–2623, June 2006.

[14] L. Chen, T. Ho, S. H. Low, M. Chiang, and J. C. Doyle, "Rate control for multicast with network coding," in *Proc. of IEEE Infocom*, Apr. 2007.

[15] T. Ho and H. Viswanathan, "Dynamic algorithms for multicast with intra-session network coding," in *Proc. of Allerton Conf. on Comm., Contr. and Comput.*, Sept. 2005.

[16] S. Sengupta, S. Rayanchu, and S. Banerjee, "An analysis of wireless network coding for unicast sessions: The case for coding-aware routing," in *Proc. of IEEE Infocom*, May 2007, pp. 1028–1036.

[17] S. Rayanchu, S. Sen, J. Wu, S. Banerjee, and S. Sengupta, "Loss-aware network coding for unicast wireless sessions: design, implementation, and performance evaluation," in *Proc. of ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, 2008, pp. 85–96.

[18] C.-C. Wang, A. Khreishah, and N. Shroff, "On cross-layer optimizations for intersession network coding on practical 2-hop relay networks," in *Proc. of the 43rd Asilomar Conference on Signals, Systems and Computers*, 2009.

[19] C.-C. Wang, "On the capacity of wireless 1-hop intersession network coding ł a broadcast packet erasure channel approach," in *Proc. of IEEE International Symposium on Information Theory*, 2010, pp. 1893–1897.

[20] D. S. Lun, M. Médard, and R. Koetter, "Efficient operation of wireless packet networks using network coding," in *Proc. International Workshop on Convergent Technologies*, Jun. 2005.

[21] D. P. Bertsekas, *Nonlinear Programming*. Athena Scientific, 1995.

[22] T. Cui, L. Chen, and T. Ho, "Energy efficient opportunistic network coding for wireless networks," Caltech, Tech. Rep., July 2007.

[23] J. Bicket, D. Aguayo, S. Biswas, and R. Morris, "Architecture and evaluation of an unplanned 802.11b mesh network," in *Proc. of ACM MobiCom*, 2005, pp. 31–42.

[24] Intersil, "HFA3863 Direct Sequence Spread Spectrum Baseband Processor Data Sheet," Apr. 2000.

[25] Roofnet, http://pdos.csail.mit.edu/roofnet/.

PLACE PHOTO HERE

Tao Cui (M'04) received the M.Sc. degree in the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada, in 2005, and the M.S. and Ph.D. degree from the Department of Electrical Engineering, California Institute of Technology, Pasadena, USA, in 2006 and 2010. His research interests are in the interactions between networking theory, communication theory, and information theory.

PLACE PHOTO HERE

Lijun Chen (M'05) received his B.S. from University of Science and Technology of China, M.S. from Institute of Theoretical Physics, Chinese Academy of Sciences and from University of Maryland at College Park, and Ph.D. from Caltech. He is a Research Scientist in the Control and Dynamical Systems Department at Caltech. His current research interests are in communication networks and power networks, optimization, game theory and their engineering application.

PLACE PHOTO HERE

Tracey Ho (M'06) is an Assistant Professor in Electrical Engineering and Computer Science at the California Institute of Technology. She received a Ph.D. (2004) and B.S. and M.Eng degrees (1999) in Electrical Engineering and Computer Science (EECS) from the Massachusetts Institute of Technology (MIT). She was a co-recipient of the 2009 Communications & Information Theory Society Joint Paper Award for the paper "Random Linear Network Coding Approach to Multicast. Her primary research interests are in information theory, network coding and communication networks.