# Distributed CSMA with Pairwise Coding

Nathaniel M. Jones*§, Brooke Shrader§, and Eytan Modiano*
*LIDS, Massachusetts Institute of Technology, Cambridge, MA 02139
§Lincoln Laboratory, Massachusetts Institute of Technology, Lexington, MA 02420

***Abstract*—We consider distributed strategies for joint routing, scheduling, and network coding to maximize throughput in wireless networks. Network coding allows for an increase in network throughput under certain routing conditions. We previously developed a centralized control policy to jointly optimize for routing and scheduling combined with a simple network coding strategy using max-weight scheduling (MWS) [9]. In this work we focus on pairwise network coding and develop a distributed carrier sense multiple access (CSMA) policy that supports all arrival rates allowed by the network subject to the pairwise coding constraint. We extend our scheme to optimize for packet overhearing to increase the number of beneficial coding opportunities. Simulation results show that the CSMA strategy yields the same throughput as the optimal centralized policy of [9], but at the cost of increased delay. Moreover, overhearing provides up to an additional 25% increase in throughput on random topologies.**

## I. Introduction

Network coding, originally introduced in [1], can increase network throughput by allowing intermediate nodes to combine or encode the data they receive, rather than simply forwarding it. The benefit of this approach for wireless transmissions was clearly demonstrated by COPE [10], an opportunistic network coding protocol that allows encoding of packets between multiple unicast sessions using binary XOR operations. The authors combine their coding strategy with a modified MAC protocol to show significant throughput improvements versus a standard 802.11 MAC on a wireless testbed. While the original work on COPE [10] explored the interplay between coding and scheduling, subsequent work in [21] motivated the need for routing protocols to be aware of network coding by formulating an offline linear program to show that significant throughput improvements are possible. In this work, we address the joint design and performance of routing, scheduling, and network coding in a wireless network.

Numerous previous works have considered joint routing and scheduling in the absence of network coding. In their seminal paper on network control [23], Tassiulas and Ephremides introduce the max-weight scheduling (MWS) and differential backlog routing policy to provide throughput optimal network control. The policy has an attractive property for dynamic control in that decisions rely only on current queue state information, without requiring knowledge of the long-term arrival rates. The authors are able to prove, using Lyapunov stability theory, that their policy can stabilize the network queues for any stochastic arrival process within the stability region of the network. In [16], MWS is extended to optimize for routing, scheduling, and power control in wireless networks. MWS is a very powerful scheduling technique, but the benefits do not come without cost. Even [23] notes that it can be cumbersome to collect queue state information from across a wireless network to a centralized controller. Additionally, MWS requires the solution to the maximum weight independent set (MWIS) problem, which is known to be NP-Hard under general interference constraints.

Jiang and Walrand [7] recently developed an adaptive carrier sense multiple access (CSMA) policy based on queue size information, and proved their policy to be throughput optimal. This adaptive CSMA policy is a randomized scheduler and operates under distributed control, addressing some of the main concerns with the scalability of MWS. In [17], the adaptive CSMA scheduler is extended by relaxing some ideal assumptions from [7], maintaining throughput optimality in the presence of collisions in control traffic. An alternate proof of optimality is provided in [14] for queue-based CSMA policies on wireless networks with primary interference constraints. In [13], the authors provide another proof of CSMA rate convergence and study the effects of collisions. A throughput optimal ALOHA policy that chooses transmission probabilities as a function of queue backlog is developed in [19]. Other works ([3], [15]) have focused on distributed queue-based scheduling, and can be extended to incorporate backpressure routing. Performance bounds are characterized in [3] for a distributed maximal scheduler with imperfect matchings. A distributed scheduler that achieves 100% throughput using a randomized gossip algorithm is developed in [15].

Recently, network coding has been incorporated into the design of scheduling and routing schemes. A number of recent works, including [4], [11], [18], and [20], develop joint scheduling and coding schemes in a network control framework, either for single-hop transmissions, or under the assumption that routes are fixed and specified *a priori*. In addressing the routing problem, [24] provides a linear optimization approach for identifying network coding opportunities on butterfly subgraphs with multiple unicast sessions, while [6] develops a policy for dynamic routing and scheduling to provide stability throughout the region from [24]. Using a different approach, [5] provides a distributed backpressure routing and maximum weight scheduling policy for a generalized COPE coding scheme, making

This work is sponsored by the Department of the Air Force and the Assistant Secretary of Defense for Research and Engineering under Air Force Contract FA8721-05-C-0002, and is supported by NSF grant CNS-0915988, ONR grant N00014-12-1-0064, and ARO MURI grant W911NF-08-1-0238. Opinions, interpretations, conclusions and recommendations are those of the authors and are not necessarily endorsed by the United States Government.

opportunistic coding decisions to increase throughput, and [22] formulates a linear program to solve the offline problem for joint routing, scheduling and pairwise coding. Finally, in our previous work we developed the LCM-Frame policy [9] that jointly optimizes for routing, scheduling, and a simple network coding scheme using an online scheduler under centralized control.

This paper differs from previous works in that we develop a distributed online queue-size based policy that is throughput optimal subject to our coding constraints. We modify the adaptive CSMA policy from [7] to incorporate a simple network coding scheme that we first proposed in [9]. We focus on pairwise coding, combined with a packet overhearing feature that can increase the number of coding opportunities with only a constant increase in algorithmic complexity. Our main contributions include:

- We propose a distributed CSMA policy for routing, scheduling, and pairwise coding that supports all arrival rates within the stability region of pairwise coding;
- We develop an extension to our coding strategy to allow for additional coding opportunities via overhearing of uncoded transmissions, and update our policy to optimize for these overhearing opportunities;
- We address several practical implementation issues;
- We provide results from packet simulation and linear program evaluation to compare the performance of our policy under various settings.

This paper is organized as follows. We describe our system model in Section II, and characterize the stability region under this model in Section III. In Section IV we design a distributed routing, scheduling, and pairwise coding policy. Section V adds a packet overhearing option to our coding strategy and updates the policy to take advantage of coding opportunities with overhearing. We address implementation issues in Section VI, provide numerical results in Section VII, and offer concluding remarks in Section VIII.

## II. Model

### A. Wireless Network

We model the wireless network as a directed hypergraph, $G = (\mathcal{N}, \mathcal{H})$, where $\mathcal{N}$ is the set of nodes in the network and $\mathcal{H}$ is the set of directed hyperedges supported by the network. Hyperedge $(a, J)$ allows node $a$ to communicate directly with a set of tail nodes $J$ using a single transmission, where $J$ is always in alphabetical order. For example, in Fig. 1a node $a$ can transmit to nodes $b$ and $c$ simultaneously over hyperedge $(a, J), J = (b, c)$. Standard edge $(a, b)$ is a special case of a hyperedge where node $b$ is the only tail node. In this paper we consider hyperedges with at most two tail nodes, $|J| \leq 2$ (corresponding to pairwise coding).

We consider unicast traffic, but utilize wireless multicast (*i.e.* transmit on hyperedges) for network coded packets and to enable a packet overhearing feature. We assume time to be continuous, and for simplicity assume unit rate links and that exogenous arrivals are for packets of a fixed size corresponding to one time unit. Packets destined for node $c$ are called *commodity* $c$ packets. Let $\lambda_a^c$ be the average rate of exogenous arrivals at node $a$ for commodity $c$, and let $\boldsymbol{\lambda} = (\lambda_a^c)$ be a vector of arrival rates for all sources $a$ and commodities $c$.

We assume that non-interfering transmissions are reliable, but otherwise allow arbitrary interference constraints. Let $\mathcal{L}$ be the set of all feasible schedules on the network. Here, schedule $\ell$ is a group of simultaneous (hyper)edge activations, and $\ell$ is feasible if these activations don't violate the network interference constraints. While our policy supports general interference models, our simulations were conducted using two simple interference models, known as 1-hop and 2-hop interference. The 1-hop interference model allows any node to transmit or receive at most one packet at a time. The 2-hop interference model requires at least two hops in the network between any simultaneous transmissions, else they will interfere.

### B. Adaptive CSMA

Wireless networks are subject to packet losses from interfering transmissions, and thus benefit from a *scheduling* policy that prevents interfering transmissions from becoming simultaneously active. CSMA is a random access scheduler where each node listens to the channel for interfering transmissions, and competition for the channel is mitigated using random backoff times. Our CSMA policy is based on the policy from [7], which we extend to account for hyperedges with our coding scheme.

Jiang and Walrand [7] developed an adaptive CSMA policy that operates in continuous time, choosing exponentially distributed backoff times for each edge $i$ as a function of the queue backlog on that edge $U_i(t)$. The policy assumes an idealized setting where each node can sense any transmission that it would interfere with and channel sensing is instantaneous. Combined with backoff times drawn from a continuous distribution[1], this ideal setting avoids packet collisions. The *backoff rate* $R_i(t)$ is updated at periodic times $t = nT$, where $T$ is the duration of the update interval. The weight of edge $i$ is chosen as $W_i(t) = U_i(t)$, and the backoff rate is chosen as:

$$r_i(t) = \alpha \cdot W_i(nT),\ \forall t : nT \leq t < (n+1)T, \tag{1}$$

$$R_i(t) = \exp\left(r_i(t)\right). \tag{2}$$

Here, $r_i(t)$ is called the *transmission aggressiveness* parameter, and $\alpha$ is a step size parameter controlling the convergence of the algorithm. The mean backoff time $1/R_i(t)$ decreases as the backlog increases, giving preference to transmissions on edges with higher backlog. In this policy, each edge $i$ transitions between *idle*, *wait*, and *transmit* states as follows.

- *Idle State:* Edge $i$ remains in the *idle* state while the channel is sensed to be busy, in that an interfering edge is active. When the channel is later sensed to be inactive[2], draw a backoff timer from an exponential distribution with mean $1/R_i$ and switch to the *wait* state.

[1]The probability that any two edges choose the same backoff time from the exponential distribution is 0, independent of the edge backoff rates.

[2]We allow for multiple simultaneous activations outside of the sensing range.

- *Wait State:* Edge $i$ remains in the *wait* state while the channel is sensed to be inactive and the backoff timer is non-zero. If the channel becomes busy, switch to the *idle* state. Else, when the backoff timer expires switch to the *transmit* state.
- *Transmit State:* Transmit packet of unit duration[3]. When the transmission has completed, switch to the *idle* state.

### C. Backpressure Routing

Combined with an optimal scheduler, backpressure routing was proved to be a throughput optimal routing strategy in [23]. The idea is simple: choose the weight of each edge as the difference in backlog across the edge for the commodity that maximizes the difference. For example, edge $(a, b)$ has weight:

$$W_{ab}(t) = \max_{c \in \mathcal{N}} [U_a^c(t) - U_b^c(t)]^+, \quad (3)$$

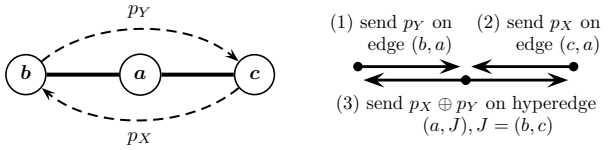
where notation $[x]^+$ represents $\max(x, 0)$. Backpressure routing was combined with adaptive CSMA for multihop traffic in [7], where the weight from Eqn. (3) is used to calculate aggressiveness parameter $r_{ab}(t)$ in Eqn. (1). The backoff rate $R_{ab}(t)$ is then calculated as in Eqn. (2).

### D. Network Coding

Network coding is a technique that allows for increased throughput by encoding packets at intermediate nodes in the network. Our network coding scheme allows data to be exchanged in fewer transmissions by strategically combining packets such that each recipient has previously seen some portion of the encoded set. In [9] we described a simple network coding scheme that under specific routing conditions allows intermediate nodes to exchange $k$ packets in $k-1$ transmissions. When evaluating this scheme on random wireless topologies, we observed that the majority of coding gains are generated by $k = 2$ pairwise coding operations. Similar observations are noted in [10], [5], and [22]. Therefore, here we limit our consideration to the pairwise coding case. We describe pairwise coding in the following example.

Consider the wireless network in Fig. 1a with 1-hop interference. We would like to exchange packets $p_Y$ and $p_X$ between nodes $b$ and $c$ via a relay at node $a$. Without network coding it takes 2 transmissions to exchange each packet, for a total of 4 transmissions. With network coding, however, these same packets can be exchanged in only 3 transmissions: (1) send $p_Y$ from $b$ to $a$; (2) send $p_X$ from $c$ to $a$; and (3) send coded packet $p_X \oplus p_Y$ as a binary XOR combination of $p_X$ and $p_Y$ from node $a$ to nodes $b$ and $c$ simultaneously via a single wireless multicast transmission. Using the packets that they contributed, nodes $b$ and $c$ can each recover the packet destined for them. In this example, network coding has increased throughput by a factor of $4/3$. As in [9], here our coded transmissions are decoded hop-by-hop and each node maintains a *side information buffer* of packets that it previously transmitted (so that they can be used to decode coded transmissions).

[3]Both exponentially distributed and unit duration transmissions are considered in [7]. The authors cite the main result from [12], which states that for an ideal CSMA network, edge activation frequencies are insensitive to the distributions of backoff and transmit times when given their means.




(a) Pairwise Coding Scenario (b) Edge Activations

Fig. 1. Pairwise coding operation at node $a$. (a) Standard edges shown with solid lines, with all hyperedges available; traffic demands shown with dashed arrows. (b) Edge activations shown with solid arrows.

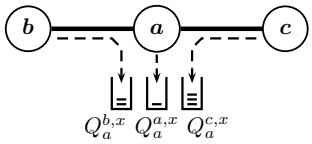



Fig. 2. Subqueues at node $a$ for commodity $x$. Subqueue $Q_a^{b,x}$ contains network arrivals from neighbor $b$; subqueue $Q_a^{a,x}$ contains local exogenous arrivals; subqueue $Q_a^{c,x}$ contains network arrivals from neighbor $c$. Packet arrivals shown in dashed arrows.

A pairwise **coding opportunity** $\big(s, (a, J)\big)$, is formed by the combination of hyperedge $(a, J)$, $J = (b, c)$, and commodity pair $s = (x, y)$ for which: (1) a packet of commodity $x$ was received at node $a$ from neighbor $c$, and (2) a packet of commodity $y$ was received at node $a$ from neighbor $b$. Identifying coding opportunities requires that nodes keep track of which one-hop neighbor supplied each packet. While other works on differential backlog routing (*e.g.* [23],[7]) track the number of packets for each commodity at each node, we further divide the queues into *subqueues* to track the number of packets from each neighbor for each commodity. For example, subqueue $Q_a^{b,x}$ at node $a$ contains $U_a^{b,x}$ number of packets received from neighbor $b$ for commodity $x$. This is illustrated in Fig. 2 for commodity $x$ packets received at node $a$ from various sources.

## III. Stability Region

The stability region $\Lambda_{NC}$ of our network coding strategy is the set of all arrival rate vectors $(\lambda_a^c)$ that can be supported while ensuring that all packet queues are stable. This region is independent of the control policy chosen, and is a special case of the stability region that we specified in [9] for network coding with maximum code size of 2. We specify the region here for convenience.

Let $f_{ab}^{d,c}$ be the rate of uncoded flow of commodity $c$ packets supplied by node $d$ and sent over edge $(a, b)$, and let $f_{aJ}^s$ be the rate of coded flow over hyperedge $(a, J)$ for each commodity in set $s$, where $\big(s, (a, J)\big)$ is a coding opportunity. For simplicity, we use the following $\hat{f}$ notation to represent a sum over a set of underlying flow variables. Let $\hat{f}_{ab}^{d,c}$ be the total uncoded and coded flow rate from node $a$ to neighbor $b$ for commodity $c$ from subqueue $Q_a^{d,c}$, where node $a$ received the packets from one-hop neighbor $d$. Thus,

$$\hat{f}_{ab}^{d,c} = f_{ab}^{d,c} + \sum_{g: s=(c,g)} f_{aJ}^s, \ \forall a, b, c, d \in \mathcal{N}, \ J = (b, d), \quad (4)$$

where the summation is over all commodities $g$ such that $\big(s,(a,J)\big), s=(c,g)$, is a coding opportunity. Let $\hat{f}^c_{ab}$ be the total coded and uncoded flow rate from $a$ to $b$ for commodity $c$ traffic from all one-hop subqueues.

$$\hat{f}^c_{ab} = \textstyle\sum_d \hat{f}^{d,c}_{ab}, \quad \forall a,b,c \in \mathcal{N} \tag{5}$$

We define the stability region by starting with some efficiency assumptions: nodes don't transmit to themselves and nodes don't transmit any traffic destined for themselves. Also, all flow variables are non-negative. The remaining constraints are as follows.

$$\lambda^c_a = \sum_b \hat{f}^c_{ab} - \sum_d \hat{f}^c_{da}, \ \forall a,c \in \mathcal{N} : a \neq c \tag{6}$$

$$\sum_b \left(\hat{f}^{d,c}_{ab} - f^{d,c}_{ab}\right) \ \leq \ \hat{f}^c_{da}, \ \forall\, a,c,d \in \mathcal{N} \tag{7}$$

$$G_{aJ} = \sum_{\ell\in\mathcal{L}} \gamma_\ell\, \mathcal{I}_{(a,J)\in\ell}, \ \forall (a,J), \ \sum_{\ell\in\mathcal{L}} \gamma_\ell = 1, \ \gamma_\ell \geq 0 \ \forall \ell \tag{8}$$

$$\sum_{d,c\in\mathcal{N}} f^{d,c}_{ab} \leq G_{aJ}, \ \forall (a,b) : J = \{b\} \tag{9}$$

$$\sum_{s\in\{\mathcal{N}\}^2} f^s_{aJ} \leq G_{aJ}, \ \forall (a,J) : \ |J| = 2 \tag{10}$$

Eqn. (6) is the *flow conservation* constraint, stating that all flow entering any node $a$ for commodity $c$ must leave node $a$, except at the destination ($a = c$). The *coding constraint* in Eqn. (7) states that the total flow into subqueue $Q^{d,c}_a$ from node $d$ gives an upper bound on the total *coded* flow out of $Q^{d,c}_a$ to all neighbors $b$. Eqn. (8) is a *convexity constraint*, stating that activation frequencies $G_{aJ}$ for all edges and hyperedges $(a,J)$ must be in the convex hull of the set of all feasible schedules $\mathcal{L}$. Here, indicator $\mathcal{I}_{(a,J)\in\ell} = 1$ if $(a,J)$ is active in schedule $\ell$, and 0 otherwise. The edge and hyperedge *rate constraints* in Eqns. (9-10) state that activation frequency $G_{aJ}$ gives an upper bound on the total flow for all commodities over edge or hyperedge $(a,J)$. The stability region for our pairwise coding strategy is the polytope bounded by the set of constraints in Eqns. (6-10).

## IV. Distributed CSMA

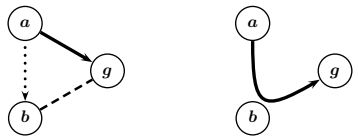
Our proposed policy adapts that of [7] to account for pairwise network coding as follows. The policy is parameterized for step-size $\alpha$ and update interval $T$. The policy updates backoff rate parameters every $T$ time units and maintains *edge* timers associated with transitions between *idle*, *transmit*, and *wait* states. Each node requires backlog information only for the queues of one-hop neighbors, therefore this policy is distributed.

### A. Distributed CSMA Policy for Pairwise Coding

**Parameter Updates:** For each edge or hyperedge $(a,J)$, we maintain a *transmission aggressiveness* (TA) parameter $r_{aJ}(t)$ and a *backoff rate* $R_{aJ}(t)$. At times $t = nT$, for integer values of $n \geq 0$, these parameters are updated as follows.

For each standard edge $(a,b)$, calculate edge weight as

$$W_{ab}(t) = \max_{d,c}[U^{d,c}_a(t) - U^{a,c}_b(t)]^+, \tag{11}$$



(a) (b)

Fig. 3. Simple packet overhearing operation. (a) Transmission from $a$ to $g$, overheard by $b$. (b) Analogous routing scenario.

where $c^*$ is the optimal commodity and $d^*$ identifies the optimal subqueue $Q^{d^*,c^*}_a$ that maximize Eqn. (11). TA parameter $r_{ab}(t)$ and backoff rate $R_{ab}(t)$ for edge $(a,b)$ can then be calculated as in Eqns. (1) and (2).

For each hyperedge $(a,J), J=(b,g)$, calculate weight as

$$W_{aJ}(t) = \max_x[U^{g,x}_a(t) - U^{a,x}_b(t)]^+ + \max_y[U^{b,y}_a(t) - U^{a,y}_g(t)]^+, \tag{12}$$

where $x^*$ identifies the optimal commodity to send from node $a$ to $b$ and $y^*$ identifies the optimal commodity to send from node $a$ to $g$. This optimal commodity pair $s = (x^*, y^*)$ and hyperedge $(a,J), J=(b,g)$, form a coding opportunity $\big(s,(a,J)\big)$ as long as there is (1) a packet at node $a$ of commodity $x^*$ from neighbor $g$, *i.e.* $U^{g,x^*}_a(t) > 0$, and (2) a packet at node $a$ of commodity $y^*$ from neighbor $b$, *i.e.* $U^{b,y^*}_a(t) > 0$. Next, calculate TA parameter $r_{aJ}(t)$ and *backoff rate* $R_{aJ}(t)$ as in Eqns. (1) and (2).

**State Transitions:** The *Idle*, *Wait*, and *Transmit* states are handled as in Section II-B. For a transmission on standard edge $(a,b)$, transmit an uncoded packet $p_C$ for optimal commodity $c^*$ from subqueue $Q^{d^*,c^*}_a$. For a transmission on hyperedge $(a,J), J=(b,g)$, transmit a coded packet $p_{XY} = p_X \oplus p_Y$, where packet $p_X$ is from subqueue $Q^{g,x^*}_a$ and packet $p_Y$ is from subqueue $Q^{b,y^*}_a$. If a subqueue is ever found to be empty, the policy creates a null packet to send.

### B. Rate Stability

It can be shown that distributed CSMA with pairwise coding stabilizes the network for all arrival rate vectors strictly interior to the stability region $\Lambda_{NC}$ specified in Eqns. (6-10). The proof follows the method shown in [7]. We give a sketch of this proof in the Appendix. Whenever the packet queues are stable, the distributed CSMA policy also stabilizes all side information buffers in the network. This is clear from the discussion of maintenance operations on side information buffers in Section VI-D.

## V. Packet Overhearing Extension

Network coding can be combined with packet overhearing to yield additional coding opportunities. Packet overhearing occurs when any nodes receive a packet concurrently with that packet's intended next-hop recipient. These additional nodes can then use their knowledge of the overheard packet in future decoding operations. The use of overhearing has been explored in [10], [5], [11], [18], and [20].

We consider a simple packet overhearing scheme to improve our network coding strategy, as shown in Fig. 3. A transmission

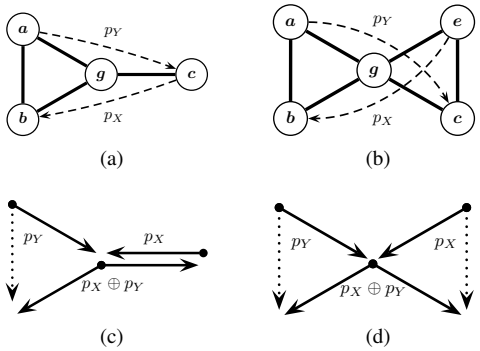



Fig. 4. Pairwise coding scenarios with overhearing shown in (a) and (b), where solid lines indicate edges and dashed lines indicate traffic demands. Associated edge activations shown below each overhearing scenario in (c) and (d), where dotted arrows indicate overheard transmissions.

from node $a$ to node $g$ that is overheard by node $b$, where nodes $b$ and $g$ are neighbors as shown in Fig. 3a, is analogous to a special routing operation where a transmission is sent from node $a$ to node $b$ to node $g$ all at once, as shown in Fig. 3b. We allow for overhearing of uncoded transmissions, creating two additional pairwise coding scenarios as shown in Fig. 4. A single overhearing operation leads to the pairwise coding opportunity shown in Fig. 4a, using edge activations in Fig. 4c. Here, node $a$ transmits packet $p_Y$ to node $g$, and this packet is overheard by node $b$, allowing $b$ to later decode the coded packet $p_X \oplus p_Y$ from $g$. The addition of a second overhearing operation leads to the pairwise coding opportunity shown in Fig. 4b, using edge activations in Fig. 4d. In addition to the overhearing at node $b$, node $e$ transmits packet $p_X$ to node $g$, and $p_X$ is overheard by node $c$. Nodes $b$ and $c$ can both decode the coded packet $p_X \oplus p_Y$ from $g$.

A standard uncoded transmission from $a$ to $g$ for commodity $x$ has weight $W_{ag}$ from Eqn. (13), while the same transmission overheard by node $b$ has weight $W_{abg}$ from Eqn. (14). (Here $d$ is the source of the subqueue at node $a$ containing the commodity $x$ packet.)

$$W_{ag} = [U_a^{d,x}(t) - U_g^{a,x}(t)]^+ \tag{13}$$

$$W_{abg} = [U_a^{d,x}(t) - U_g^{b,x}(t)]^+ \tag{14}$$

For both transmissions, the packet exits subqueue $Q_a^{d,x}$ at node $a$, but the packet enters different subqueues at node $g$. For the standard transmission from $a$ to $g$, the packet enters subqueue $Q_g^{a,x}$ and a copy is stored in the side information buffer at node $a$. However, for the overheard transmission, the packet enters subqueue $Q_g^{b,x}$ because we treat the packet as if it was received at $g$ from node $b$, as shown in Fig. 3b. The overheard packet is then stored in the side information buffer at node $b$ instead of at node $a$.

### A. Improved Stability Region

Overhearing leads to minor changes to the stability region. We represent the overhearing transmission as flow variable $\dot{f}_{dab}^{j,c}$, which is the flow from subqueue $Q_d^{j,c}$ at node $d$ to node $b$ and overheard by node $a$. We introduce an *Overhearing Constraint* as a prerequisite for our overhearing strategy: overhearing flow variables can only represent positive flow for hyperedges $(d, J), J = (a, b)$, where edge $(a, b)$ is also available in the network; otherwise the overhearing flow variable must take the value of zero flow. The total uncoded and coded flows $\hat{f}_{ab}^{d,c}$ from Eqn. (4) becomes:

$$\hat{f}_{ab}^{d,c} = f_{ab}^{d,c} + \sum_j \dot{f}_{dab}^{j,c} + \sum_{g:s=(c,g)} f_{aJ}^s, \quad \begin{matrix} \forall a,b,c,d \in \mathcal{N}, \\ J = (b,d). \end{matrix} \tag{15}$$

Eqns. (5) and *Flow Conservation* (6) incorporate the addition of overhearing from Eqn. (15) but otherwise remain unchanged. The *Coding Constraint* (7) changes to account for outgoing overheard transmissions:

$$\textstyle\sum_b \left(\hat{f}_{ab}^{d,c} - f_{ab}^{d,c} - \sum_g \dot{f}_{abg}^{d,c}\right) \le \hat{f}_{da}^c, \quad \forall\, a,c,d \in \mathcal{N} \tag{16}$$

The *Hyperedge Rate Constraints* in Eqns. (8-10) remain unchanged. However, note that we have generalized the hyperedge activation rate $G_{aJ}$ in Eqn. (10) to include both pairwise coding and uncoded overhearing, as these both operate over hyperedges. The stability region with overhearing is then given by the constraints in Eqns. (6, 8-10, 16).

### B. Policy Modification for Overhearing

The overhearing extension requires only minor changes to how hyperedge rate parameters are handled by our distributed CSMA policy. Parameter updates for standard edges remain unchanged, and the state transitions behave exactly as without the overhearing feature.

**Parameter Updates for Hyperedges:** At each time $t = nT$, for integer $n \ge 0$, for each hyperedge $(a, J), J = (b, g)$, calculate three weights:

1) For transmissions from $a$ to $g$ overheard by $b$, $W_{aJ}^1 = \max_{c,d}[U_a^{d,c}(t) - U_g^{b,c}(t)]^+$ if $(b, g) \in \mathcal{H}$, else $W_{aJ}^1 = 0$.
2) For transmissions from $a$ to $b$ overheard by $g$, $W_{aJ}^2 = \max_{c,d}[U_a^{d,c}(t) - U_b^{g,c}(t)]^+$ if $(g, b) \in \mathcal{H}$, else $W_{aJ}^2 = 0$.
3) For coded transmission from $a$ to $b$ and $g$, $W_{aJ}^3$ is calculated as in Eqn. (12).

We then choose the coding or overhearing operation that maximizes the weight of the hyperedge:

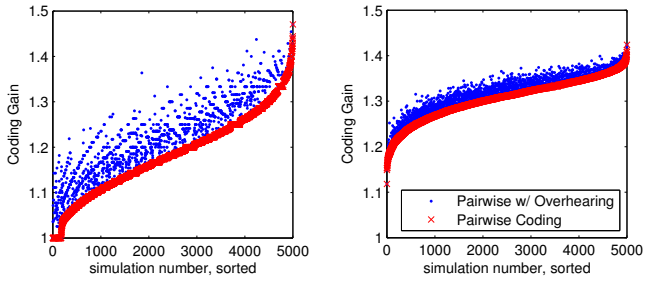
$$W_{aJ}(t) = \max\left\{W_{aJ}^1, W_{aJ}^2, W_{aJ}^3\right\}. \tag{17}$$

TA parameter $r_{aJ}(t)$ and *backoff rate* $R_{aJ}(t)$ are calculated as in Eqns. (1) and (2).

### C. Linear Program Results

We compare coding gains directly by evaluating the bounds of the stability region using an LP solver. We generate 100 random 16 node topologies, where there are $16 \times 15 = 240$ possible traffic demands on each topology. We choose traffic demand vector $\boldsymbol{\lambda} \in \{0, 1\}^{240}$, where each demand is activated with probability $p$, and find the maximum offered load without coding $\rho_1$ such that $\boldsymbol{\lambda} \cdot \rho_1 \in \Lambda$ and the maximum offered load with network coding $\rho_{NC}$ such that $\boldsymbol{\lambda} \cdot \rho_{NC} \in \Lambda_{NC}$. Coding gain is then the ratio $\rho_{NC}/\rho_1$. These topologies are evaluated with 2-hop interference constraints.

Fig. 5 shows coding gains for traffic demand probabilities $p = 1/16$ and $p = 1/2$. For each demand probability, 5000




(a) $p = 1/16$ (b) $p = 1/2$

Fig. 5. Comparison of pairwise coding gains with and without overhearing for individual traffic vectors. Random traffic demands with probability $p$. Results sorted in order of increasing gain for coding without overhearing.

| **Data Type of $R_i$** | **Max. $R_i$ before overflow** | **Max. Associated $r_i = \log(R_i)$** |
|---|---|---|
| Double Precision Floating Point | 1.7977e+308 | 709.78 |
| Single Precision Floating Point | 3.4028e+38 | 88.72 |
| Unsigned 64-bit Integer | 1.8447e+19 | 44.36 |

TABLE I
MAXIMUM VALUES FOR $r_i$ BEFORE OVERFLOW OF RATE $R_i$.

individual arrival rate vectors are generated (50 per topology). Coding gain is then evaluated for each arrival rate vector, both with and without overhearing. For each demand probability, the vectors are sorted in increasing order of coding gain for pairwise coding without overhearing, and the values for coding gain are plotted in that order. In Fig. 5a where $p = 1/16$, we observe up to 25% additional gain from overhearing, although these additional gains are only present in 32% of our observations. In Fig. 5b where $p = 1/2$, the additional gain from overhearing is at most 5%, and these additional gains are present in 50% of our observations. In both scenarios, the median additional coding gain from overhearing is around 2%, however the small computational cost to include overhearing and the potential increase in coding gain make it a worthwhile extension. It is interesting to note that the gain from overhearing is greatest when the traffic vector is sparse. Additional traffic demands increase the likelihood of coding opportunities without the need for overheard transmissions, so overhearing provides only small incremental gains when the traffic vector is dense.

## VI. IMPLEMENTATION CONSIDERATIONS

Next we discuss some details related to implementation of the distributed CSMA policy.

### A. Backoff Times

Backoff rate $R_i$ grows exponentially with aggressiveness parameter $r_i$, and for any finite precision computation this can lead to overflow of variable $R_i$. This occurs, for example, in the case of a bursty source node, and is exacerbated on systems that require the use of fixed-point arithmetic. Table I shows values of $r_i$ that lead to overflow for various data types of variable $R_i$. When the differential backlog is large, multiple outgoing edges $i$ can be assigned backoff rate $R_i = \infty$ and the node will not be able to correctly discriminate between exponentially distributed backoff times $B_i \sim \text{Exp}(R_i = \infty) = 0$.

Larger values of $r_i$ can be supported by comparing logarithms of the backoff times instead of comparing the backoff times directly. We use the inverse transform method to generate backoff times $B_i \sim \text{Exp}(R_i)$ as follows. Generate random variable $Z \sim \text{Uniform}[0,1]$, where the CDF of $Z$ is $F_Z(z) = \mathbb{P}(Z \le z) = z$. Then choose backoff times using the function $B_i = -\log(Z)/R_i$. The CDF of $B_i$ is $F_{B_i}(b_i) = \mathbb{P}(B_i \le b_i) = \mathbb{P}(-\log(Z)/R_i \le b_i) = \mathbb{P}(Z \ge e^{-b_i R_i}) = 1 - e^{-b_i R_i}$, so $B_i$ is exponentially distributed with rate $R_i$. Taking the logarithm of $b_i$ and using $R_i = e^{r_i}$, we have

$$\log(b_i) = \log\left(\frac{-\log(z)}{e^{r_i}}\right) = \log\big(-\log(z)\big) - r_i, \qquad (18)$$

which allows almost the full range of values supported by variable $r_i$, except when $z$ is extremely close to 0 or 1. The earliest of a group of backoff times can then be chosen as

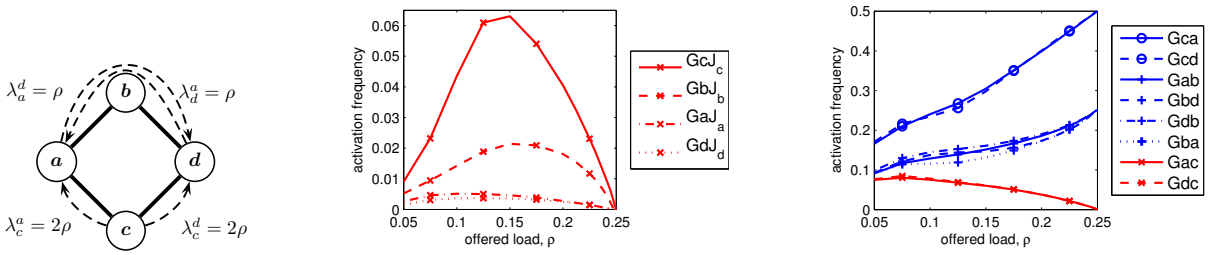
$$\min_i b_i = \exp\big(\min_i \log(b_i)\big). \qquad (19)$$

A node can then choose the minimum backoff time between interfering edges with the correct activation probabilities, or a simulation engine can choose between all waiting edges in the network. New backoff times can be drawn at each comparison due to the memoryless property of the exponential distribution.

### B. Avoiding Greedy Application of Network Coding

It may be tempting to opportunistically promote edge activations into hyperedge activations. However, it is known that greedy application of network coding can reduce throughput [4]. One such scenario is the 4 node diamond topology with 1-hop interference and arrival rates as indicated in Fig. 6a. Here the network can be stabilized for offered loads $\rho < 1/4$. With 1-hop interference, edges $(c,a)$ and $(c,d)$ mutually interfere with all hyperedges in the network, $(a, J_a)$, $(b, J_b)$, $(c, J_c)$, and $(d, J_d)$, where $J_a = (b,c)$, $J_b = (a,d)$, $J_c = (a,d)$, and $J_d = (b,c)$. Thus, a greedy application of network coding on any hyperedge reduces the fraction of time that edges $(c,a)$ and $(c,d)$ can be active. This problem is illustrated as follows. Without loss of generality, assume that traffic only flows on efficient paths (*e.g.* traffic from $c$ to $a$ doesn't go the long way around the diamond), and let $\rho$ be feasible. By Eqn. (9) we find activation frequency $G_{ca} \ge f_{ca}^{c,a} = 2\rho$ and likewise $G_{cd} \ge 2\rho$. Using the convexity of schedules from Eqn. (8), $G_{ca} + G_{cd} + G_{aJ_a} + G_{bJ_b} + G_{cJ_c} + G_{dJ_d} \le 1$, and thus $G_{aJ_a} + G_{bJ_b} + G_{cJ_c} + G_{dJ_d} \le 1 - 4\rho$. Therefore, as the offered load $\rho$ approaches the stability bound $1/4$, all hyperedge activation frequencies must go to 0 as a prerequisite for stability.

We evaluate distributed CSMA with pairwise coding on the scenario from Fig. 6a by simulating our policy using Poisson arrivals, $\alpha = 1/10$, and $T = 10$. The simulations are run for 10 million time units for each value of offered load considered. Fig. 6b shows the activation frequency of each hyperedge versus offered load $\rho$, while Fig. 6c shows activation frequencies for standard edges in the same scenario. As $\rho$ approaches 1/4, we observe that $G_{ca}$ and $G_{cd}$ each converge to $1/2 = 2\rho$, $G_{ab}$, $G_{bd}$, $G_{db}$, and $G_{ba}$ all converge to $1/4 = \rho$, and all other edges and hyperedges converge to 0, as desired.




(a) Scenario (b) Hyperedge Activation Frequencies (c) Edge Activation Frequencies

Fig. 6. The 4 node diamond scenario. Under 1-hop interference, greedy application of network coding can reduce throughput. Stability requires all activation frequencies $G_i \to 0$ for each hyperedge $i$ as offered load approaches stability bound $\rho = 0.25$. Our policy satisfies this condition.

### C. Minimum Queue Size with Network Coding

As the arrival rate vector approaches the upper bound of the stability region, our policy requires use of small values of step-size $\alpha$ to achieve necessary service rates. However, we observe that queues grow large for small values of $\alpha$. As a function of step-size $\alpha$ and offered load $\rho$, we find a lower bound on the average network queue size required for rate convergence on the 3 node scenario in Fig. 1a. In particular we show that the queue size must be inversely proportional to $\alpha$. For simplicity of this example, let the arrival rates be symmetric, *i.e.* $\rho = \lambda_b^c = \lambda_c^b$, and let $\rho$ be in the range $1/4 < \rho < 1/3$ such that pairwise coding is required to stabilize the network.

Using the result from [12], we model schedule activations of our policy as a Markov chain. In this simple 3 node scenario, at most one edge can be active at a time, so activation frequency $\pi_i$ of each schedule $i$ is the service rate for edge $i$. Note convexity constraint $\pi_\emptyset + \pi_{ba} + \pi_{ca} + \pi_{ac} + \pi_{ab} + \pi_{aJ} = 1$, where $J = (b, c)$, $\pi_i \geq 0$, and $\pi_\emptyset$ is the activation frequency of the empty schedule. By symmetry, $\pi_{ca} = \pi_{ba}$ and $\pi_{ac} = \pi_{ab}$. Combine the convexity constraint with service requirements $\pi_{ba} = \pi_{ca} \geq \rho$, and $\pi_{ac} + \pi_{aJ} \geq \rho$, we find upper bound $\pi_{ab} \leq 1 - 3\rho$. Applying this bound to service requirement $\pi_{ab} + \pi_{aJ} \geq \rho$, we find lower bound $\pi_{aJ} \geq 4\rho - 1$. Taking the ratio between $\pi_{aJ}$ and $\pi_{ab}$,

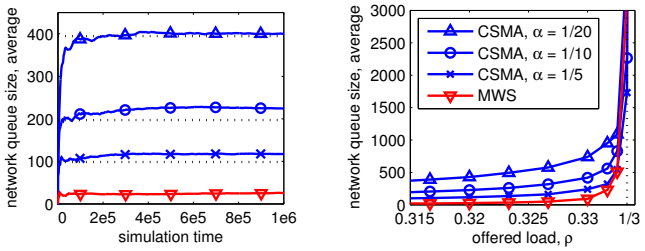
$$\frac{4\rho - 1}{1 - 3\rho} \leq \frac{\pi_{aJ}}{\pi_{ab}} = \frac{\pi_\emptyset R_{aJ}}{\pi_\emptyset R_{ab}} = \frac{e^{r_{ab} + r_{ac}}}{e^{r_{ab}}} = e^{r_{ac}}, \qquad (20)$$

where $\pi_i = \pi_\emptyset R_i$ is given by the stationary distribution of the Markov chain, $R_{aJ} = \exp(r_{aJ})$, and $r_{aJ} = r_{ab} + r_{ac}$. Solving for $r_{ac}$ yields $r_{ac} \geq \log \frac{4\rho - 1}{1 - 3\rho}$. By a similar method, we find $r_{ba} \geq \log \frac{\rho}{1 - 3\rho} + r_{ac}$. By symmetry, $r_{ca} = r_{ba}$ and $r_{ac} = r_{ab}$.

When rate parameters are stable, average queue sizes can be found as follows. Applying Eqn. (1), $U_a^{b,c} = U_a^{c,b} = \frac{r_{ac}}{\alpha}$, and accounting for differential backlog, $U_b^{b,c} = U_c^{c,b} = \frac{r_{ba} + r_{ac}}{\alpha}$. The policy will back-fill packets to learn the forward direction of traffic flow, so $U_b^{a,c} = U_c^{a,b} = U_a^{b,c}$. Taking a sum over all queues, a lower bound on average network queue size is:

$$\sum_{i,c,d} U_i^{d,c} \geq \frac{2}{\alpha} \log \frac{\rho}{1 - 3\rho} + \frac{8}{\alpha} \log \frac{4\rho - 1}{1 - 3\rho}. \qquad (21)$$

Considering offered load $\rho = 0.32$ in Eqn. (21), we find that convergence of service rates requires a minimum network queue size of $19.73/\alpha$, which is inversely proportional to $\alpha$ as expected from Eqn. (1). We evaluate this lower bound on network queue size for various values of $\alpha$, as shown in Fig. 7a. Simulations for this scenario are discussed in Section VII.




(a) Queue Size vs. Time (b) Queue Size vs. Offered Load

Fig. 7. Simulations on 3 node scenario from Fig. 1a. Legend applies to both subplots. (a) Offered load $\rho = 0.32$ for various $\alpha$. Dotted lines show lower bound on stable queue size from Eqn. (21). (b) Stability bound at $\rho = 1/3$.

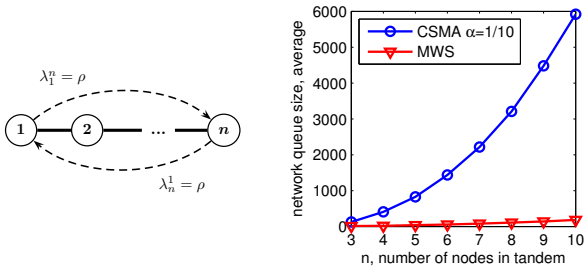
### D. Managing Side Information Buffers

This subsection describes a distributed method to determine when packets can be discarded from side information buffers. Let $S_a^{b,c}$ be the size of the side information buffer at node $a$ for packets sent to neighbor $b$ for commodity $c$. The policy exchanges backlog information with neighbors every $T$ units of time. Side information buffers are kept in FIFO order, so when node $b$ sends backlog information $U_b^{a,c}$ to node $a$, the associated side information buffer at node $a$ can be reduced such that it contains only the most recent $S_a^{b,c} = U_b^{a,c}$ packets. Without loss of generality, assume node $b$ can transmit at most one packet at a time. Therefore node $b$ can transmit at most $T$ packets between sending backlog updates to node $a$. Thus, $S_a^{b,c} \leq U_b^{a,c} + T$, and the side information buffers are stable whenever the queues are stable.

## VII. Numerical Results

We simulate our policy using Poisson arrivals, and compare distributed CSMA with our MWS policy from [9]. All configurations were simulated for 10 million time units.

We first consider the performance of our CSMA policy on the 3 node scenario from Fig. 1a with symmetric offered load $\rho = \lambda_b^c = \lambda_c^b$. We simulate CSMA with $\alpha = \{1/5, 1/10, 1/20\}$ and update interval $T = 10$. Fig. 7a shows the network queue size as a function of time for offered load $\rho = 0.32$. Here we see that CSMA operates with queue size at roughly $1/\alpha$ times that of MWS. The lower bound on CSMA queue size from Eqn. (21), shown as a dotted horizontal lines, appears reasonably close to actual network queue size in this scenario.




(a) $n$ Node Tandem Scenario (b) Packet Simulation

Fig. 8. Queue size of CSMA with pairwise coding for tandem network with increasing number of intermediate nodes with symmetric end-to-end traffic.

However, the distance between the bound and actual queue size will vary based on offered load and arrival process. Fig. 7b shows average network queue size versus offered load, where the bound of the stability region is indicated with a dashed vertical line at $\rho = 1/3$. For all configurations, we see that queues remain relatively small when the offered load is interior to the stability region, and the queues grow large as the offered load approaches the stability bound.

We next consider how queue sizes scale with the number of nodes $n$ on a tandem configuration with symmetric end-to-end traffic, as shown in Fig. 8a. We configure CSMA with $\alpha = 1/10$, $T = 10$, $\rho = \lambda_1^n = \lambda_n^1 = 0.3$, and evaluate this scenario under the 1-hop interference model. Fig. 8b shows average network queue size for our CSMA and MWS policies on networks with $n = \{3, 4, ..., 10\}$ nodes. For both policies we observe that the queues grow quadratically with number of nodes $n$ due to differential backlog routing, which is consistent with findings from [2]. The ratio between CSMA and MWS network queue sizes is roughly 10 for $n = 3$ nodes and increases to around 30 for $n = 10$ nodes.

Finally, we consider queue size versus offered load for a 16 node scenario with 11 traffic demands as shown in Fig. 9a, with 2-hop interference. (This is the same scenario considered in Fig. 3 of [9].) MWS results are shown on Fig. 9b, while CSMA results are shown on Fig. 9c. The dotted vertical lines indicate the bounds of the stability region (computed using an LP solver) at $\rho = 1/19$ without coding, $\rho = 1/17.5$ for pairwise coding, and at $\rho = 1/16$ for pairwise coding with overhearing. This yields a pairwise coding gain of $19/17.5 = 1.086$ without overhearing and $19/16 = 1.188$ with overhearing. We see that the queues remain relatively small for values of $\rho$ interior to the stability bound, and the queues grow rapidly when $\rho$ exceeds the bound. We also observe that CSMA queues operate at between 10 and 20 times those for MWS, although this will vary with $\alpha$.

## VIII. Conclusion

In this paper, we consider distributed techniques for joint routing, scheduling, and pairwise network coding to maximize throughput in wireless networks. We presented the distributed CSMA policy for pairwise coding, and showed that this policy can come arbitrarily close to supporting the full stability region allowed by our coding constraint. We developed a packet overhearing extension to increase the number of beneficial coding opportunities and evaluated our policy with and without overhearing on multiple scenarios. On random scenarios we find the additional gains from our overhearing scheme are low on average at around 2%, but occasionally we observe larger gains of up to 25% that make this simple extension worthwhile.

In comparing performance of our CSMA and MWS policies, we find that the distributed control of the CSMA policy comes at the expense of growth in average queue size. For a simple pairwise coding scenario, we provide a lower bound on stable CSMA queue size as a function of the offered load and $\alpha$. This bound is inversely proportional to $\alpha$, and we found it useful for approximating the network queue size in our simulations. We evaluated stable queue size as a function of the number of nodes in a tandem network, and observe quadratic growth in stable CSMA queue size. While MWS also experiences quadratic growth, the growth rate is noticeably faster for CSMA.

## Appendix
## Rate Stability

Using appropriate choices for parameters $\alpha$ and $T$, we wish to show that for any strictly feasible arrival rate vector $\lambda$ and any flow decomposition $\hat{f}$, the distributed CSMA policy chooses TA parameters $r_i$ such that service $s_i(r)$ dominates arrivals $\hat{f}_i$ for each edge $i$. Here, $\lambda$ is strictly feasible if $(\lambda + \epsilon) \in \Lambda_{NC}$, for $\epsilon \geq 0$, and $\hat{f}$ is a flow decomposition of $\lambda$ according to Eqns. (6-10). First, we show that if a solution is attainable for finite $r^*$, then $s_i(r^*) \geq \hat{f}_i$, $\forall i$. Second, we show that the solution is attainable whenever the arrival rate is strictly feasible. Combining the first and second steps gives the desired result.

Let $\gamma_\ell$ be an activation probability for schedule $\ell$ satisfying flow decomposition $\hat{f}_i$, and let $\pi_\ell(r)$ be the actual activation frequency of each schedule $\ell$ according to service rates $s_i(r) \forall i$. Indicator $\mathcal{I}_{i\in\ell} = 1$ if edge $i$ is active in schedule $\ell$, and 0 otherwise. Then $\hat{f}_i = \sum_\ell \gamma_\ell \mathcal{I}_{i\in\ell}$. Again using the result from [12], we model schedule activations of our policy as a continuous time Markov chain where the schedule activation frequencies conditioned on $r_i$ are given by:

$$\pi_\ell(r) = \exp(\textstyle\sum_i r_i \mathcal{I}_{i\in\ell}) \,/\, C(r) \qquad (22)$$

$$C(r) = \textstyle\sum_j \exp(\sum_i r_i \mathcal{I}_{i\in j}) \qquad (23)$$

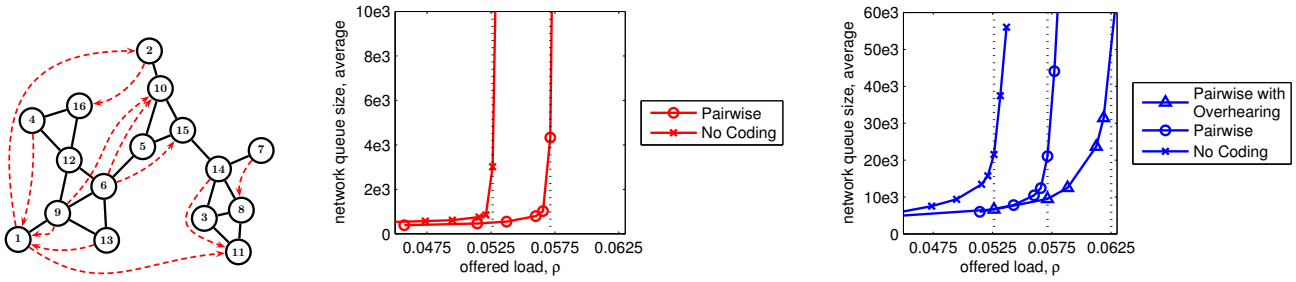
We can minimize the Kullback-Leibler divergence between distributions $\gamma$ and $\pi(r)$ by solving $\sup_{r\geq 0} F(r)$, where $F(r)$ is non-positive for $r \geq 0$ and is defined as

$$F(r) = \textstyle\sum_\ell \gamma_\ell \log\left(\pi_\ell(r)\right) = \sum_i \hat{f}_i r_i - \log(C(r)). \qquad (24)$$

Note that $\frac{\partial}{\partial r_i} F(r) = \hat{f}_i - s_i(r)$, so a *distributed* gradient algorithm to solve $\sup_{r\geq 0} F(r)$ is

$$r_i(n+1) = [r_i(n) + \alpha(n)(\hat{f}_i - s_i(r(n)))]^+, \ \forall i. \qquad (25)$$

Choosing $r_i(0) = 0$, $\alpha(n) = \alpha$, interval $n$ of duration $T$, and observing that $\hat{f}_i$ and $s_i(r)$ correspond to queue arrivals and departures, respectively, we obtain $r_i(nT) = \alpha U_i(nT)$. This is in the form of Eqn. (1).




(a) Scenario (b) MWS (c) CSMA

Fig. 9. Comparing MWS and CSMA for a 16 node scenario. (a) Traffic demands as dashed lines with arrows. (b)&(c) Stability bounds as dotted vertical lines.

**Existence Proposition:** If $r^* \geq 0$ exists such that $F(r^*) = \sup_{r \geq 0} F(r)$, then $s_i(r^*) \geq \hat{f}_i, \forall i$. Dualize each constraint $r_i \geq 0$ with dual variables $d_i \geq 0$: $\mathcal{L}(r,d) = F(r) + \sum_i d_i r_i$. At solution $r^*$ we have $\frac{\partial}{\partial r_i}\mathcal{L}(r^*,d^*) = \hat{f}_i - s_i(r^*) + d_i^* = 0$. We know $d_i \geq 0$, so $s_i(r^*) \geq \hat{f}_i \forall i$.

**Attainability Proposition:** If $\lambda$ is strictly feasible, then $F(r^*) = \sup_{r \geq 0} F(r)$ is attainable. In [8], it is shown that the dual of $\sup_{r \geq 0} F(r)$ is

$$\max_{u} \quad -\textstyle\sum_\ell u_\ell \log(u_\ell) \quad \text{s.t.} \quad \sum_\ell (u_\ell \mathcal{I}_{i\in\ell}) \geq \hat{f}_i, \forall i$$
$$\textstyle\sum_\ell u_\ell = 1,\ u_\ell \geq 0. \qquad (26)$$

The optimal value for Eqn. (26) occurs when

$$u_\ell^* = \exp(\textstyle\sum_i y_i^* \mathcal{I}_{i\in\ell}) \,/\, \big(\sum_j \exp(\sum_i y_i^* \mathcal{I}_{i\in j})\big),\ \forall \ell, \qquad (27)$$

where $y_i$ is the dual variable for constraint $\sum_\ell (u_\ell \mathcal{I}_{i\in\ell}) \geq \hat{f}_i$. Observe that $u_\ell^*$ is in the form of $\pi_\ell(r^*)$ from Eqns. (22-23), where $y_i^* = r_i^* \,\forall i$. Then the optimal value for Eqn. (26) equals $F(r^*)$ and is obtained whenever $\lambda$ is strictly feasible.

**Combining the two propositions:** If $\lambda$ is strictly feasible, then $s_i(r) \geq \hat{f}_i$, $\forall i$. Note that for fixed values of parameters $\alpha$ and $T$, we are only guaranteed that the service rates will converge to the neighborhood of the link arrivals $\hat{f}_i$. For rate stability, it is sufficient for the convergence neighborhood to be fully contained in the stability region. By assumption, arrival rates are strictly feasible, so there always exists a value of $\alpha$ small enough that the neighborhood of convergence is fully within the stability region. Thus, the parameterized policy can come arbitrarily close to supporting the full stability region.

Note that for pairwise coding, *e.g.* in Fig. 1a, we have assumed that TA parameter for hyperedge $(a, J), J = (b, c)$, is $r_{aJ} = r_{ab} + r_{ac}$. This assumption is confirmed by verifying that the total service rate $s_{ab}(r)$ on edge (a,b) is $\pi_{ab} + \pi_{aJ}$: $s_{ab}(r) = \frac{\partial}{\partial r_{ab}} \log\big(C(r)\big) = \big(\exp(r_{ab}) + \exp(r_{ab}+r_{ac})\big)/C(r)$ $= \pi_\emptyset\big(R_{ab} + R_{aJ}\big) = \pi_{ab} + \pi_{aJ}$, where $\pi_\emptyset = 1/C(r)$.

## References


[1] R. Ahlswede, N. Cai, S.Y.R. Li, and R.W. Yeung. Network information flow. *IEEE Trans. on Info. Theory*, 46(4):1204–1216, 2000.

[2] L. Bui, R. Srikant, and A. Stolyar. Novel architectures and algorithms for delay reduction in back-pressure scheduling and routing. In *Proc. INFOCOM*, 2009.

[3] P. Chaporkar, K. Kar, and S. Sarkar. Throughput guarantees through maximal scheduling in wireless networks. In *Proc. Allerton*, 2005.

[4] P. Chaporkar and A. Proutiere. Adaptive network coding and scheduling for maximizing throughput in wireless networks. In *Proc. MobiCom*, 2007.

[5] T. Cui, L. Chen, and T. Ho. Energy efficient opportunistic network coding for wireless networks. In *Proc. INFOCOM*, 2008.

[6] A. Eryilmaz and D. S. Lun. Control for inter-session network coding. In *NetCod: Workshop on Network Coding*, 2007.

[7] L. Jiang and J. Walrand. A distributed CSMA algorithm for throughput and utility maximization in wireless networks. In *Proc. Allerton*, 2008.

[8] L. Jiang and J. Walrand. A distributed algorithm for maximal throughput and optimal fairness in wireless networks with a general interference model. *EECS Department, Univ. of California, Berkeley, Tech. Rep*, 2008.

[9] N. M. Jones, B. Shrader, and E. Modiano. Optimal routing and scheduling for a simple network coding scheme. In *Proc. INFOCOM*, 2012.

[10] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft. XORs in the air: Practical wireless network coding. *IEEE/ACM Trans. on Networking*, 16(3):497–510, 2008.

[11] A. Khreishah, C.C. Wang, and N.B. Shroff. Cross-layer optimization for wireless multihop networks with pairwise intersession network coding. *IEEE Journal on Selected Areas in Comm.*, 27(5):606–621, 2009.

[12] Soung Chang Liew, Caihong Kai, J. Leung, and B. Wong. Back-of-the-envelope computation of throughput distributions in csma wireless networks. In *Proc. IEEE ICC*, 2009.

[13] J. Liu, Y. Yi, A. Proutiere, M. Chiang, and H. V. Poor. Towards utility-optimal random access without message passing. *Wireless Communications and Mobile Computing*, 10(1):115–128, 2010.

[14] P. Marbach and A. Eryilmaz. A backlog-based csma mechanism to achieve fairness and throughput-optimality in multihop wireless networks. In *Proc. Allerton*, 2008.

[15] E. Modiano, D. Shah, and G. Zussman. Maximizing throughput in wireless networks via gossiping. In *Proc. ACM SIGMETRICS*, 2006.

[16] M. J. Neely, E. Modiano, and C. E. Rohrs. Dynamic power allocation and routing for time-varying wireless networks. *IEEE Journal on Selected Areas in Comm.*, 23(1):89–103, 2005.

[17] J. Ni and R. Srikant. Distributed CSMA/CA algorithms for achieving maximum throughput in wireless networks. In *Proc. ITA Workshop*, 2009.

[18] G. S. Paschos, L. Georgiadis, and L. Tassiulas. Optimal scheduling of pairwise XORs under statistical overhearing and feedback. In *Proc. IEEE RAWNET*, 2011.

[19] S. Rajagopalan, D. Shah, and J. Shin. Network adiabatic theorem: an efficient randomized protocol for contention resolution. In *Proc. ACM SIGMETRICS*, 2009.

[20] Y.E. Sagduyu, D. Guo, and R. Berry. Throughput and stability of digital and analog network coding for wireless networks with single and multiple relays. In *Proc. WICON*, 2008.

[21] S. Sengupta, S. Rayanchu, and S. Banerjee. An analysis of wireless network coding for unicast sessions: The case for coding-aware routing. In *Proc. INFOCOM*, 2007.

[22] S. Shabdanov, C. Rosenberg, and P. Mitran. Joint routing, scheduling, and network coding for wireless multihop networks. In *Proc. WiOpt*, 2011.

[23] L. Tassiulas and A. Ephremides. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. *IEEE Trans. Auto. Control*, 37(12):1936–1948, Dec. 1992.

[24] 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. ISIT*, 2006.