# Control of Multi-hop Communication Networks for Inter-session Network Coding

Atilla Eryilmaz, Desmond S. Lun, and B. T. Swapna

**Abstract**

This paper provides a solution to the question of how, when and where to perform inter-session network coding for a general network model both under wired and wireless conditions. In particular, an original queuing architecture and a dynamic routing-scheduling-coding strategy are introduced for serving multiple sessions when linear network coding is allowed across sessions. This policy provides a novel extension to the class of back-pressure policies by incorporating inter-session coding decisions via simple rules on the relevant queue-length levels. Despite the fact that the capacity region of inter-session coding is a challenging open problem, in this paper, we prove that our algorithm can support any set of rates in a non-trivial characterized region of achievable rates. In addition to its practical implications, this work also provides a theoretical framework in which the gains of inter-session network coding and pure routing can be compared.

**Index Terms**

Inter-session Network Coding, Backpressure Policies, Lyapunov Stability, Stochastic Networks, Butterfly Network, Dynamic Algorithms

## I. Introduction

The efficient use of the scarce resources of communication networks is critical in providing high quality service to the flows (or sessions) that compete for these resources. Thus, any method which conveys more information with the same number of transmissions is extremely valuable for network performance. Network Coding is one of the most promising methods that suggests significant throughput, delay and energy gains over existing strategies [1].

Traditionally, packets that are generated by the source of a session are forwarded through intermediate nodes towards their destinations without modification. This approach has been challenged by Ahlswede et al. in their seminal work [2] where they introduced the concept of Network Coding, which allows intermediate nodes to *mix* the information contained in different packets into a newly generated packet. It has been shown in [2] that such an operation has the effect of relieving congested areas of the network while spreading the information transfer over less congested parts, thus achieving significantly higher end-to-end throughput than is possible with traditional methods.

Network Coding in packet networks can be classified into two types: *intra*-session coding (where coding is restricted to packets belonging to the same session or connection) and *inter*-session coding (where this restriction is lifted and coding is allowed among packets belonging to possibly different sessions). The former, which is also referred to as superposition coding [3], has been extensively studied. It is well-known that intra-session coding improves the throughput of lossless multicast sessions (see, for example, [2], [4], [5]) and of lossy sessions—unicast or multicast (see, for example, [6], [7]). It is also known, however, that intra-session coding is suboptimal [3]: inter-session coding is necessary to achieve optimal throughput in general.

Performing inter-session coding, however, is not straightforward. To perform inter-session coding optimally, linear coding operations are not sufficient [8], and, even if we limit ourselves to a particular class of linear coding operations, deciding what operations to perform is an NP-hard problem [4]. Optimal inter-session coding, however, is not necessary:

An earlier version of the material in this paper was presented at the Information Theory and Applications Workshop, San Diego, CA, 2007.

This work was supported in part by DTRA grant HDTRA1-08-1-0016, and NSF Awards 0916664-CCF and 0953515-CNS.

A. Eryilmaz and B. T. Swapna are with the Department of Electrical and Computer Engineering, Ohio State University, Columbus, OH 43210, USA (e-mail: eryilmaz@ece.osu.edu, buccapat@ece.osu.edu).

D. S. Lun is with the Phenomics and Bioinformatics Research Centre, School of Mathematics and Statistics, and Australian Centre for Plant Functional Genomics, University of South Australia, Mawson Lakes, SA 5095, Australia (e-mail: desmond.lun@unisa.edu.au).

sub-optimal inter-session coding can achieve significant performance gains in practice, as demonstrated by COPE [9]—a protocol for inter-session coding that has been evaluated in simulation studies and testbed implementations. Remarkably, COPE simply exploits a rudimentary form of inter-session coding that generalizes the "physical piggybacking" discussed in [10]. We are therefore motivated to develop methods for inter-session coding that, though not optimal, achieve significant throughput gains over intra-session coding for a wide range of networks.

The main contributions of this work are as follows.

- We introduce a novel queueing architecture to capture the decodability constraint of inter-session coding operations and propose a dynamic algorithm that utilizes the resulting queue-length information to make simple scheduling, routing and inter-session network coding decisions. The adaptive nature of the algorithm allows its use in networks with unknown topologies and arrival statistics. In the algorithm, every node utilizes only that information which is relevant to its decision, which naturally facilitates distributed operation.
- We provide a rigorous analysis of the network performance, and prove that our algorithm can support any flow rate which lies within an inter-session achievable rate region characterized in [11]—a region that we refer to as $\Lambda$. The region $\Lambda$ is not the capacity region of inter-session network coding, since this region is unknown, except under some additional restrictions (e.g. [12]); rather it is a non-trivial expansion of the capacity region of intra-session network coding that relies only on simple, practicable XOR coding. The region is constructed by exploiting coding opportunities of the "butterfly" type (see Figure 1), which is the type of inter-session coding that is most overwhelmingly considered in the literature.

This paper continues the work along the line of suboptimal, yet improved, methods for inter-session coding, which includes [13], [14], [11], [15]. The defining characteristic of this paper is that, rather than proposing an algorithm that operates on given flow rates (or ones it measures), we propose a dynamic routing-scheduling-coding strategy that operates solely on appropriately maintained queue-state information. Thus, although the algorithm described in [15] (which is the result of independent work by Ho et al.) bears some similarities to our strategy, it nevertheless differs in this defining aspect. Dynamic strategies such as ours do not require flow rates as an input and can be run "on-line". They will generally take some time to find the desired operating point, but they are robust to dynamics of flows and network topology because they react to present circumstances as measured by the state of the queues.

Our strategy extends that of COPE and can be seen, moreover, as an extension of the dynamic routing-scheduling strategies of Tassiulas and Ephremides [16] and others (e.g. [17], [18], [19], [20], [21]), which do not allow for coding; and of the dynamic routing-scheduling-coding strategy [22], which allows for only intra-session coding. We also note more recent works on the dynamic coding-scheduling strategy [23], coding-aware queue management for unicast flows [24], [25], and energy efficient network coding design in wireless networks [26] that utilize inter-session coding. These works, however, differ from ours in that they are built upon COPE-like coding that explore inter-session coding opportunities where decoding must happen within one hop. In comparison, our strategy allows -in fact seeks- inter-session coding opportunities beyond the next hop neighbors, and thus can achieve a larger throughput region, as demonstrated in our numerical results (see Section VII).

This line of work suggests a paradigm shift in the way we approach the notion of capacity. In particular, rather than aiming for explicit characterizations of the capacity region of complex networks, which is generally a difficult task, we aim to develop distributively computable decision rules that can be shown, via control theoretic techniques, to achieve close to capacity performance. In fact, this approach suggests an algorithmic way to accurately estimate the capacity region of complex networks.

The rest of the paper is organized as follows. In Section II, we introduce a general system model and describe our goal. In Sections III and IV, we respectively focus on the wired butterfly and wireless exchange networks, which form the building block for the general scenario, to describe our dynamic algorithm and provide insight into its operation. In Section V, we consider general wired networks, and we describe our joint routing, scheduling, and coding strategy and prove its stabilizing properties. In Section VI, we describe how our algorithm and performance analysis can be extended to the case of wireless networks. After discussing several numerical results in Section VII, we complete with concluding comments in Section VIII.

## II. System Model and Goal

Although our algorithm and analysis apply to both wireline and wireless networks, for purposes of exposition we provide most of our descriptions and analysis for the wireline network model (Sections III and V), and subsequently point out to the main components necessary to extend the arguments to the wireless network case (Sections IV and VI). Next, we describe the system model for wired and wireless networks, the traffic model, and the coding capabilities.

In both the wired and wireless context, we assume that the system operates in a time-slotted manner, where all nodes are assumed to be synchronized to a common clock[1]. We consider a packet network, where each *packet*, denoted by $\mathbf{P}$, is a vector of length $m$ over a finite field $\mathbb{F}_q$ for fixed integer values of $m$ and $q$.

**Coding Capability**: Traditional networks treat packets as unalterable objects and are concerned with their forwarding from the source to destinations. Thus, they are concerned with the *scheduling and routing* problems of network communications. In this work, we allow each node to perform *linear network coding* over the packets it holds. In particular, for a set $\{\mathbf{P}_1, \cdots, \mathbf{P}_K\}$ of packets, a node can create a coded packet

$$\mathbf{P} = \sum_{k=1}^{K} a_k \mathbf{P}_k,$$

where $a_k \in \mathbb{F}_q$ for $k = 1, \cdots, K$, and the summation is over the finite field $\mathbb{F}_q$ [2]. Hence, the contents of packets are allowed to be modified in the interior of the network. Such a network coding operation is shown to provide throughput gains compared to traditional forwarding strategies ([2]). It is also known that random coding of packets that belong to the same session (*intra-session* coding) is always advantageous ([27], [7]). However, arbitrary random coding of packets across different sessions (*inter-session* coding) may be either advantageous or harmful to performance. This is because the receivers may not be able to accumulate sufficient side information to decode the randomly coded packets unless proactive effort is exerted to convey critical side information to the intended receivers.

**Wireline Network Model**: We model a wireline network as a directed graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N}$ is the set of nodes and $\mathcal{E}$ is a set of directed edges (links) that represent point-to-point connections. We incorporate effects of link failures into the model by assuming that link qualities change over time according to some unknown stochastic process. In particular, every link takes one of a set of states in every slot which determines the number of packets that can be transmitted over that link in that slot. It is assumed that nodes know the state of their outgoing links at the beginning of each time slot. Notice that we do not assume the knowledge of channel statistics which may be difficult to obtain, but limit the knowledge to locally available channel state information. To facilitate exposition, we assume that the state space for channel quality of each link consists of two states: ON and OFF, where in the ON state a single packet can be transmitted, while in the OFF state no packet can be transmitted. We let $C_{(i,j)}[t] \in \{0, 1\}$ denote the channel state of the link $(i, j) \in \mathcal{E}$ at timeslot $t$ and denote its mean by $\gamma_{(i,j)}$. Although $\gamma_{(i,j)}$ is a function of the underlying channel statistics, its value is not known since the statistics are assumed to be unknown. Moreover, we assume that the link state of a given link is independently and identically distributed (i.i.d.) in every time slot. These assumptions are not restrictive and can be extended to allow more states (each representing the supportability of a different number of packet transmissions), and stationary and ergodic channel state processes (see for example [19]).

**Wireless Network Model**: Wireless network model differs from its wireline counterpart in two aspects: availability of broadcast transmissions and existence of interference. We capture the broadcast nature of transmissions by modeling the network as a directed *hypergraph* $\mathcal{H} = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N}$ is the set of nodes and $\mathcal{E}$ is a set of directed hyperedges[3] that represent broadcast transmissions. To capture the interference between concurrent transmissions, we let $\Theta$ denote the set of all feasible hyperlink activation vectors, where each feasible vector is a $0-1$ vector that yields a set of nonconflicting hyperlinks that can be simultaneously active. If a hyperlink $(m, N)$ is allowed to transmit, we use $\gamma_{(m,N)}$ to denote the mean transmission rate that is achievable over it. Similar to the wireline case, a description for the wireless network of the region $\Lambda$ in which intersession coding is allowed is provided in Section VI.

[1] This assumption, commonly made in the literature, can be relaxed by including a buffer zone between subsequent slots, and hence rendering an imperfectly synchronized system effectively synchronized.

[2] In our discussions, for ease of exposition, we will assume that $q = 2$ and the summation is a simple XOR operation.

[3] A hyperedge is a generalization of an edge that starts at a single node and ends at possibly more than one node.

**Traffic and Queueing Model**: A set $\mathcal{F}$ of *flows* (or *sessions*) compete for the resources of the network, where each flow, say $f \in \mathcal{F}$, is described with its beginning node, denoted by $s(f) \in \mathcal{N}$, and its destination node, denoted by $d(f) \in \mathcal{N}$. We assume that associated with each flow, say $f$, there is a fixed route with $\mathcal{N}_f$ and $\mathcal{E}_f$ that respectively denotes the set of nodes and the set of links that the flow traverses. Thus, unless packets of a flow is coded with another flow's packets, their route to the destination is set. Our interest is in optimally identifying inter-session coding opportunities that will exploit network resources. We let $A^{(f)}[t]$ denote the number of exogenously generated Flow-$f$ packets that enter $s(f)$ at the beginning of time slot $t$ to be transmitted to $d(f)$. We assume that $A^{(f)}[t]$ i.i.d.[4] over $t$ with a mean of $\lambda^{(f)}$ and finite second moment.

The coded or uncoded packets are maintained in infinite size queues as they traverse the network nodes. We note that the architecture of the queueing network is a decision choice in the overall design. Thus, the goal of the design is to develop the queuing architecture and the routing-scheduling-coding operations to be performed within the network so as to achieve the stability of all the queues as long as the mean arrival rates are achievable. Next, we define a strong notion of stability that is commonly required.

*Definition 1 (Stability, Achievability):* A queue with queue length $Q[t]$ at time $t$ is said to be 'stable' if

$$\limsup_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[Q[t]] < \infty.$$

Further, we say that a mean arrival rate vector $\lambda = (\lambda^{(f)})$ is 'achievable for a given policy' if, under that policy, all the queues in the network are stable when the mean arrival rates are $\lambda$.

In our discussions, we let $\Gamma$ denote the *capacity* of linear inter-session coding for a given network and set of flows, which describes the largest set of achievable flow rates when linear network coding is allowed across sessions. The characterization of the capacity region $\Gamma$ is currently unknown for a general network topology, but there are works which characterize an achievable region (e.g. [11]) or the full capacity region under pairwise coding limitation ([12]).

Our work builds on the work by Traskov et al. [11], which provides a non-trivial achievable region that we denote by $\Lambda$. We propose a novel queueing architecture and a dynamic algorithm that reacts to the network state to achieve high throughput. In particular, we develop a decentralized routing-scheduling-coding algorithm which, in addition to scheduling and routing decisions, determines when and where inter-session coding operations should be performed to improve achievable throughput performance. Our solution applies both to the wired and wireless networks and is shown to achieve any set of mean rates that lies inside the corresponding achievable rate region $\Lambda$. In this paper, we refrain from discussing $\Lambda$ in depth, but provide its description as it pertains to our analysis in the appendix.

## III. The Butterfly Network

We first describe our algorithm for the canonical butterfly network introduced in [2], depicted in Figure 1. This allows us to explain the essential components of, and give general intuition for, our algorithm without complicated notation. In Sections V, and VI, we extend our results to general wireline and wireless networks. Since the $\Lambda$ region is essentially obtained by decomposing a general wireline network into superimposed butterfly networks, a deep understanding of the butterfly network is critical for the extension.

### A. Achievable Rate Region

Suppose there are two unicast flows: Flow-$f$ from node $b$ to $c'$, i.e., $s(f) = b$, and $d(f) = c'$; and Flow-$g$ from node $b'$ to $c$, i.e., $s(g) = b'$, and $d(g) = c$. The exogenous arrivals for these flows have mean rates $\lambda^{(f)}$ and $\lambda^{(g)}$ packets/slot, respectively. Assuming that each available link $(i, j)$ has capacity $\gamma_{(i,j)} = 1$, Figure 1 describes the operation of inter-session coding: packets from these two flows are mixed together using the XOR operation and "remedy" packets are supplied to allow the coding operation to be undone and the independent flows recovered downstream. Notice that when coding is not allowed, i.e. only routing is available, link $(m, n)$ becomes the bottleneck link and the total achievable rate $\lambda^{(f)} + \lambda^{(g)}$ cannot exceed 1 packet per slot.

[4]Similar to the channel state process case, it is possible to relax this assumption to cover general stationary and ergodic arrival processes by extending the analysis of this paper to block of time slots (see e.g. [19]). We avoid such analysis for clarity.

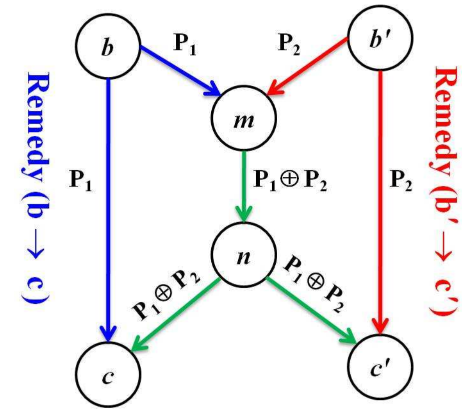


Fig. 1. The butterfly network. Flows $f$ and $g$ begin at nodes $s(f) = b$ and $s(g) = b'$, respectively, and end at nodes $d(f) = c$ and $d(g) = c'$, respectively. A packet $\mathbf{P}_1$, from Flow $f$, is XORed with a packet $\mathbf{P}_2$, from Flow $g$, thus requiring only one "coded" packet to be sent over link $(m, n)$. In exchange, two "remedy" packets must be sent for decoding the coded packet at the destinations $c$ and $c'$. In particular, remedy packet $\mathbf{P}_1$ must be sent from node $b$ to node $c$ and remedy packet $\mathbf{P}_2$ must be sent from node $b'$ to node $c'$.

In the more general case of an ON/OFF channel with average link rate $\gamma_{(i,j)} \in [0, 1]$ for each available link $(i, j)$, the set of $(\lambda^{(f)}, \lambda^{(g)})$ that are achievable by simple routing and coding strategies are given next.

*Proposition 1 (Butterfly Achievable Rate Region with Routing ($\Gamma^R_{BF}$)):* The achievable rate region of the butterfly network with routing contains the set of all $(\lambda^{(f)}, \lambda^{(g)})$ that satisfy:

$$\lambda^{(f)} \leq \min(\gamma_{(b,m)}, \gamma_{(n,c')}), \tag{1}$$

$$\lambda^{(g)} \leq \min(\gamma_{(b',m)}, \gamma_{(n,c)}), \tag{2}$$

$$\lambda^{(f)} + \lambda^{(g)} \leq \gamma_{(m,n)}, \tag{3}$$

$$\lambda \geq 0. \tag{4}$$

*Proof:* This result trivially follows from the max-flow-min-cut theorem, and is omitted for brevity. ■

$\Gamma^R_{BF}$ gives the largest set of mean flow rates that can be achieved with a routing strategy. When network coding is allowed, the set of achievable $(\lambda^{(f)}, \lambda^{(g)})$ increases as shown next.

*Proposition 2 (Butterfly Achievable Rate Region with Coding ($\Gamma^C_{BF}$)):* The achievable rate region of the butterfly network with coding contains the set of all $(\lambda^{(f)}, \lambda^{(g)})$ that satisfy: for some $\lambda^{(f,g)}$, we have (1), (2), (4), and

$$\lambda^{(f)} + \lambda^{(g)} - \lambda^{(f,g)} \leq \gamma_{(m,n)}, \tag{5}$$

$$\lambda^{(f,g)} \leq \gamma_{(b,c)}, \tag{6}$$

$$\lambda^{(f,g)} \leq \gamma_{(b',c')}, \tag{7}$$

$$0 \leq \lambda^{(f,g)} \leq \min(\lambda^{(f)}, \lambda^{(g)}). \tag{8}$$

*Proof:* We describe the need for each constraint. (1), (2), and (4) are as in the routing case. For decodability at the destinations, for each coded packet $\mathbf{P}_1 \oplus \mathbf{P}_2$, there must be a remedy packet transmitted over the side links. Noting that $\lambda^{(f,g)}$ captures the rate of coded packets generated at node $m$, the side links must be able support this rate, which is given in (6) and (7). That the rate of coded packets cannot be more than the rate of each flow is given in (8). Finally, since over link $(m, n)$ one coded packet is transmitted instead of two uncoded flow packets, the actual rate of flow over it is given by $\lambda^{(f)} + \lambda^{(g)} - \lambda^{(f,g)}$, which leads to (5). ■

In the definition, $\lambda^{(f,g)}$ captures the flow rate of the coded packets generated at node $m$. We illustrate the potential gains of coding in a butterfly network with error-free links (i.e. $\gamma_{(i,j)} = 1$ for all $(i, j) \in \mathcal{E}$) in Figure 2. We can observe that the set of arrival rates supportable with inter-session coding can be as large as twice of what can be supportable with routing. In Section VII, we also depict the regions of an asymmetric case (cf. Figure 10).

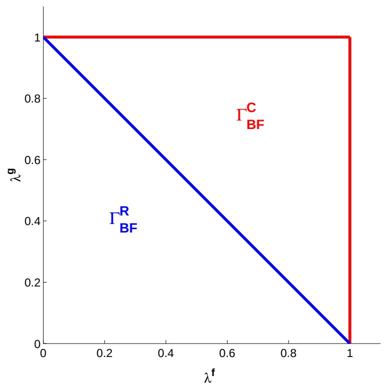


Fig. 2. Capacity regions of routing and coding for the butterfly network of Figure 1 with unit capacity links.

### B. Dynamic Coding and Scheduling Algorithm

In the butterfly scenario of Figure 1, if the exogenous arrival rates $(\lambda^{(f)}, \lambda^{(g)})$, the channel statistics and the complete network topology were known at some centralized point, then $\lambda^{(f,g)}$ in Proposition 2 could, in theory[5], be found for any $(\lambda^{(f)}, \lambda^{(g)})$ in the achievable rate region $\Gamma^{C}_{BF}$, thus allowing the rate pair to be achieved. In many network settings, however, $(\lambda^{(f)},\ \lambda^{(g)})$ and the channel statistics are not known, and we moreover do not have a centralized point with complete network information. Also, the arrivals and link states are stochastically varying. Therefore, we wish to make decisions on coding, routing, and scheduling on-the-fly with decentralized operations. This is the type of dynamic policy that we seek.

Intuitively, a good place to make the coding decision is at node $m$. If node $m$ observes that its instantaneous packet queue has many packets queued for Flow-$f$ and many packets queued for Flow-$g$, then it is likely that node $m$ represents a bottleneck. In this case, we could alleviate the congestion at node $m$ by coding, which introduces remedy packets at node $b$ and $b'$. We assume that node $m$ is capable of sending small, *remedy request*, protocol messages[6] to nodes $b$ and $b'$, requesting that these additional packets be sent. If the links $(b, c)$ and $(b', c')$ are themselves congested, however, it may not be a good idea for node $m$ to code. So it is not clear what the decision rule must be to exploit the network coding advantage while guaranteeing decodability at the receivers and stability of the system. We propose a dynamic algorithm that achieves this goal in the next section.

Next, we give a dynamic policy that yields coding decisions based on the occupancies of neighboring queues. These queue-lengths must be maintained so that they serve as a measure of decodability of the coded packets.

*Definition 2 (Dynamic Coding and Scheduling Algorithm for the Butterfly Network):* At the beginning of slot $t$, let $Q_k^{(f)}[t]$ be the number of uncoded Flow-$f$ packets at node $k$; let $Q_k^{rem(c)}$ be the number of remedy packets destined for node $c$ at node $k$; and let $Q_k^{(c,(c,c'))}[t]$ be the number of coded packets destined for node $c$ at node $k$. At each time slot, Flow-$f$ packets arrive at node $b$ and are placed into queue $Q_b^{(f)}$, and Flow-$g$ packets arrive at node $b'$ and are placed into queue $Q_{b'}^{(g)}$.

We consider each of the nodes in turn.

- Node $b$ maintains two queues, $Q_b^{(f)}$ and $Q_b^{rem(c)}$, and its policy is straightforward: At each time slot, it uses whatever capacity is available on link $(b, m)$ to serve $Q_b^{(f)}$, removing served packets from the queue and placing them into $Q_m^{(f)}$, and it uses whatever capacity is available on link $(b, c)$ to serve $Q_b^{rem(c)}$, removing served packets from the queue, which then reach their destination. The situation at node $b'$ is similar to that at node $b$.
- Node $n$ maintains four queues, $Q_n^{(f)}$, $Q_n^{(g)}$, $Q_n^{(c,(c,c'))}$, and $Q_n^{(c',(c,c'))}$. It checks to see if $Q_n^{(g)}$ or $Q_n^{(c,(c,c'))}$ is greater, and serves the greater of the two using whatever capacity it has on link $(n, c)$; likewise, it checks to see if $Q_n^{(f)}$ or $Q_n^{(c',\{c,c'\})}$ is greater, and serves the greater of the two using whatever capacity it has on link $(n, c')$. Nodes $c$ and $c'$ are final destination nodes and do not maintain queues.

[5]This operation would become computationally complex as the size of the network scales, quickly rendering it impractical.
[6]Note that these messages are simple signals much shorter than packet lengths. We assume that their consumption of link capacity is negligible.

- The coding decision of node $m$ is based on

$$\rho^{(f)}[t] \triangleq \left(Q_m^{(f)}[t] - Q_n^{(f)}[t]\right),$$
$$\rho^{(g)}[t] \triangleq \left(Q_m^{(g)}[t] - Q_n^{(g)}[t]\right),$$
$$\sigma[t] \triangleq \left[Q_m^{(f)}[t] - (Q_n^{(c',\{c,c'\})}[t] + Q_b^{rem(c')}[t])\right] + \left[Q_m^{(g)}[t] - (Q_n^{(c,\{c,c'\})}[t] + Q_{b'}^{rem(c)}[t])\right].$$

  - If $\max\left(\rho^{(f)}[t], \rho^{(g)}[t], \sigma[t]\right) \leq 0$, then no packet is served over link $(m, n)$.
  - Otherwise, if $\sigma[t]$ is greater than $\max(\rho^{(f)}[t], \rho^{(g)}[t])$, then coding is performed: Node $m$ removes one packet from $Q_m^{(f)}$ and one packet from $Q_m^{(g)}$, forms a single coded packet from the XOR of the two, and transmits the coded packet on link $(m, n)$. If one of the queues is empty the other packet is coded with an all zero packet, and if both queues are empty no packet is served over $(m, n)$. Upon reception at node $n$, the coded packet is placed into both queues $Q_n^{(c,(c,c'))}$ and $Q_n^{(c',\{c,c'\})}$. As well as transmitting the coded packet on link $(m, n)$, node $m$ transmits two remedy request protocol messages, one to $b$ and one to $b'$. These remedy request protocol messages ultimately result in a remedy packet being placed into each queue $Q_b^{rem(c)}$ and queue $Q_{b'}^{rem(c')}$. Node $m$ repeatedly forms coded packets and sends remedy request protocol messages for them for as much capacity is available on link $(m, n)$ in time slot $t$.
  - If either $\rho^{(f)}[t]$ or $\rho^{(g)}[t]$ is greater than $\sigma[t]$, then simple routing instead of coding is performed. Specifically, if $\rho^{(f)}[t] > \rho^{(g)}[t]$, then $Q_m^{(f)}$ is served using all the available capacity of link $(m, n)$, otherwise $Q_m^{(g)}$ is served over link $(m, n)$.

$\diamond$

We can understand the policy employed on node $m$ as an extension of *differential backlog* (see [16], [20], [19]): $\rho^{(f)}$ and $\rho^{(g)}$ give the traditional differential backlog associated with $f$ and $g$, respectively, and the factor $\sigma[t]$ represents the differential backlog associated with coding. To calculate the latter correctly, we need to account for the following two effects of coding: first, by coding, we effectively serve two packets for the price of one, removing a packet from both $Q_m^{(f)}$ and $Q_m^{(g)}$ while transmitting only a single packet on link $(m, n)$; second, we have to pay for this advantage of coding with remedy packets, which create packets in $Q_b^{rem(c)}$ and $Q_{b'}^{rem(c')}$, one for each flow. The first effect causes $Q_m^{(f)}[t] - Q_n^{(c',(c,c'))}$ to be summed with $Q_m^{(g)}[t] - Q_n^{(c,(c,c'))}$ when calculating the differential backlog, and the second effect causes $Q_b^{rem(c)}[t]$ and $Q_{b'}^{rem(c')}[t]$ to be subtracted from the differential backlog, finally yielding $\sigma[t]$ as the correct differential backlog associated with coding.

Our main result, Theorem 1 in Section V, proves that the policy we describe above will stabilize all exogenous arrival rates, $\lambda^{(f)}$ and $\lambda^{(g)}$, that lie strictly in the interior of the achievable rate region given in Proposition 2.

## IV. The Wireless Exchange Network

In this section, we consider another special case. This special case relates to wireless networks and is relevant, in particular, to COPE [9], a practical wireless network coding protocol that has shown significant performance improvements over routing. Our approach not only establishes a theoretical framework for inter-session wireless network coding, but also allows the intersession coding performed by COPE to be generalized beyond a single hop.

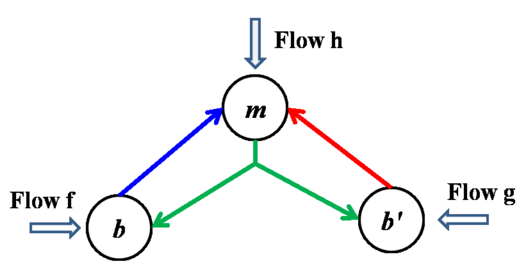


Fig. 3. The wireless exchange network. Flow-$f$ and Flow-$g$ begin at nodes $s(f) = b$ and $s(g) = b'$, respectively, and end at nodes $d(f) = b'$ and $d(g) = b$, respectively. Further, Flow-$h$ begins at node $s(h) = m$ and ends at node $d(h) = b$.

### A. Achievable Rate Region

The network we now consider is shown in Figure 3. This network has been considered in the context of physical piggybacking [10] and has been referred to previously as the modified wireless butterfly network [1]. Here, we refer to it as the *wireless exchange network*. The network consists of three links, $(b, m)$, $(b', m)$ and $(m, (b, b'))$. The third of the three links, represented by a hyperedge, is a broadcast link from node $m$ to nodes $b$ and $b'$. For simplicity, we assume that each link $(i, J)$ has capacity $\gamma_{(i,J)} = 1$, though it is equally valid for the capacities of the links to be dependent on each other if the wireless medium is not shared orthogonally.

In this network, we suppose there are three unicast flows: Flow-$f$ from node $b$ to $b'$, Flow-$g$ from $b'$ to $b$, and Flow-$h$ from node $m$ to $b$. Flow-$f$ and Flow$g$ are *information exchange* flows. Inter-session coding is performed by mixing these two flows together with an XOR operation at node $m$. This set-up is akin to a butterfly network where no remedy packets are required because they are already available at the relevant destination nodes. Flow-$h$ is a cross-traffic flow that competes for resources with Flow-$f$ and Flow-$g$. It is unimportant whether the destination of Flow-$h$ is $b$, $b'$, or both. Balancing the amount of resources given to Flow-$h$ with those given to Flow-$f$ and Flow-$g$ is of vital importance to performance in the wireless exchange network, and we will see that our approach leads naturally to a fair policy for balancing these flows. The following two propositions follow as in Section III-A.

*Proposition 3 (Wireless Exchange Achievable Rate Region with Routing ($\Gamma^R_{WE}$)):* The achievable rate rregion of the wireless exchange network with routing contains the set of all $(\lambda^{(f)}, \lambda^{(g)}, \lambda^{(h)})$ that satisfy:

$$\lambda^{(f)} \leq \gamma_{(b,m)}, \tag{9}$$

$$\lambda^{(g)} \leq \gamma_{(b',m)}, \tag{10}$$

$$\lambda^{(f)} + \lambda^{(g)} + \lambda^{(h)} \leq \gamma_{(m,(b,b'))}, \tag{11}$$

$$\lambda \geq 0. \tag{12}$$

*Proposition 4 (Wireless Exchange Achievable Rate Region with Coding ($\Gamma^C_{WE}$)):* The achiveable rate region of the wireless exchange network with coding contains the set of all $(\lambda^{(f)}, \lambda^{(g)}, \lambda^{(h)})$ that satisfy: for some $\lambda^{(f,g)}$, we have (9), (10), (12), and

$$\lambda^{(f)} + \lambda^{(g)} + \lambda^{(h)} - \lambda(f, g) \leq \gamma_{(m,(b,b')}, \tag{13}$$

$$0 \leq \lambda^{(f,g)} \leq \min(\lambda^{(f)}, \lambda^{(g)}). \tag{14}$$

### B. Dynamic Coding and Scheduling Algorithm

*Definition 3 (Dynamic Coding and Scheduling Algorithm for the Wireless Exchange Network):* As with Definition 2, let $Q_k^{(f)}$ be the number of Flow-$f$ packets at node $k$ at the beginning of slot $t$. At each time slot, Flow-$f$ packets arrive at node $b$ and are placed into queue $Q_b^{(f)}$, Flow-$g$ packets arrive at node $b'$ and are placed into queue $Q_{b'}^{(g)}$, and Flow-$h$ packets arrive at node $m$ and are placed into queue $Q_m^{(h)}$.

Each node applies the following policy.

- Nodes $b$ and $b'$ each serve packets on their queues according to the available capacity on their respective outgoing links, removing served packets from their queues and placing them onto the corresponding queue at node $m$.
- Node $m$ calculates the following:

$$\rho^{(h)}[t] \triangleq Q_m^{(h)}[t],$$
$$\sigma[t] \triangleq \left(Q_m^{(f)}[t] - Q_b^{(f)}[t]\right) + \left(Q_m^{(g)}[t] - Q_{b'}^{(g)}[t]\right).$$

  - If $\sigma[t] > \rho^{(h)[t]}$, then coding is performed as described in Definition 2. If either $Q_m^{(f)}$ or $Q_m^{(g)}$ is zero, then packets from the non-empty one of the two queues is sent.
  - Otherwise, $Q_m^{(h)}$ is served using all the available capacity of link $(m, (b, b'))$.

Thus, node $m$ maintains three queues, namely, $Q_m^{(f)}$, $Q_m^{(g)}$, and $Q_m^{(h)}$, and decides whether or not to perform coding based on the length of these three queues. COPE in fact uses a very similar policy, but it instead maintains a single queue and always serves the head-of-line packet, coding it with other packets with possible.

Using similar arguments as those employed in Theorem 1, we can show that the policy described in Definition 3 will stabilize all exogenous arrivals $\lambda^{(f)}$, $\lambda^{(g)}$ and $\lambda^{(h)}$ that lie strictly in the interior of the achievable rate region given in Proposition 4.

## V. General Wireline Networks

In Section III, we considered the canonical butterfly network. In this section, we extend the algorithm to be implemented in more general networks, and prove its stabilizing properties. For general wireline networks, we can consider superimposing or overlaying butterfly networks into the network to extend the butterfly network case. In general, this kind of superimposing of butterfly networks will not achieve the capacity region, but it will expand the region that is achievable by routing. In Appendix I, we describe such an achievable rate region, $\Lambda$, obtained by superimposing butterfly networks as introduced by Traskov et al. [11]. We note that $\Lambda$ essentially considers all possible ways in which butterfly networks can appear in a general wireline network, and, for each butterfly network, it allows a coded packet to be transmitted on the center link (or path) as long as remedy packets are transmitted on the side links (or paths).

### *A. Dynamic Routing-Scheduling-Coding Algorithm*

Recall that flow $f \in \mathcal{F}$ originates at node $s(f)$ and is destined to node $d(f)$ (see Figure 4 for an example). We assume that associated with each flow, say $f$, there is a fixed route with $\mathcal{N}_f$ and $\mathcal{E}_f$ that respectively denotes the set of nodes and the set of links that the flow traverses. For each node $k$ on the route of flow $f$, we let $\mathcal{U}_k^{(f)}$ denote the set of upstream nodes visited by Flow-$f$ packets before they arrive at $k$, and $\mathcal{D}_k^{(f)}$ denote the downstream nodes that Flow-$f$ packets will traverse after $k$. Also, we let $pt^{(f)}(k)$ and $ch^{(f)}(k)$ denote the immediate parent and child, respectively, of node $k$ on the route of Flow-$f$.

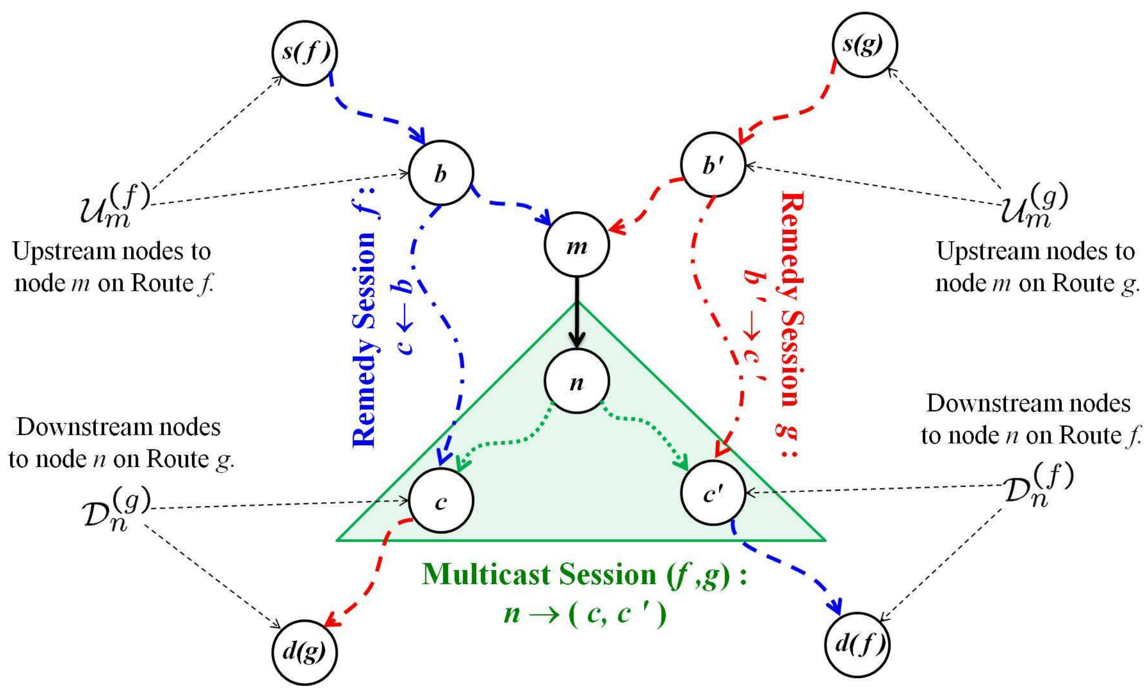


Fig. 4. Flow-$f$ goes from $s(f)$ to $d(f)$ and Flow-$g$ goes from $s(g)$ to $d(g)$, both traversing link $(m, n)$. The dashed lines indicate simple paths composed of multiple links; the dash-dotted lines indicate potentially multiple simple paths between its end points; and the dotted lines indicate end-to-end communication, possibly over a general subgraph rather than a tree. A decision to perform inter-session coding across flows $f$ and $g$ over link $(m, n)$ with remedy nodes $(b, b')$ and decoding nodes $(c, c')$ results in: two unicast sessions for remedy packets [$b \rightarrow c$ and $b' \rightarrow c'$] whose routes are to be dynamically constructed; and one multicast session [$n \rightarrow \{c, c'\}$] over a subgraph to be dynamically constructed.

Next, we introduce some new notation to distinguish different service types and a queueing architecture to be used in the algorithm.

**Notation for Service Processes**: Recall that, when it is ON each link $(m,n) \in \mathcal{E}$ can serve a single packet in every time slot. We distinguish the different types of service on a link as follows:

- $S_{(m,n)}^{(f)}[t]$ : Number of uncoded Flow-$f$ packets served over $(m,n) \in \mathcal{E}_f$, in slot $t$.
- $S_{(m,n)}^{rem(c)}[t]$ : Number of remedy packets served over link $(m,n) \in \mathcal{E}$ in slot $t$ that are destined to node $c$.
- $S_{(m,n)}^{((f,g),(c,c'))}[t]$ : Number of intra-session coded $(f,g)$ packets that are multicast to the destination set $(c,c')$.
- $S_{(m,n)}^{((f,g),(b,b'),(c,c'))}[t]$ : Number of newly inter-session coded $(f,g)$ packets served over $(m,n) \in \mathcal{E}$ in slot $t$ that originate from $(b,b')$ and are destined for nodes $(c,c')$.

Notice that $S_{\cdot}^{(\cdot)}[t]$ denotes the actual amount of service provided, not the offered service, which may be larger due to the unavailability of packets. To distinguish, we will use $M_{\cdot}^{(\cdot)}[t]$ for the offered service with the same subscript-superscript convention used for $S_{\cdot}^{(\cdot)}[t]$. Since there will be a single packet transmission in every ON channel state over link $(m,n)$, the vector of offered services $\mathbf{M}[t]$ must satisfy: for each link $(m,n) \in \mathcal{E}$, we have

$$\sum_f M_{(m,n)}^{(f)}[t] + \sum_c M_{(m,n)}^{rem(c)}[t] + \sum_{(f,g),(c,c')} M_{(m,n)}^{((f,g),(c,c'))}[t] + \sum_{(f,g),(b,b'),(c,c')} M_{(m,n)}^{((f,g),(b,b'),(c,c'))}[t] \leq C_{(m,n)}[t], \quad (15)$$

which merely states that the available capacity $C_{(m,n)}[t]$ over link $(m,n)$ in slot $t$ must be divided to serve uncoded, remedy, intra/inter-session coded packets.

**Queueing Architecture**: We propose the following queueing architecture to be used in our algorithm:

- $Q_j^{(f)}$ : The queue at node $j$ that holds uncoded Flow-$f$ packets.
- $Q_j^{rem(c)}$ : The queue at node $j$ that holds remedy packets destined to node $c$.
- $Q_j^{(c,(f,g),(c,c'))}$ : The queue at node $j$ that holds coded $(f,g)$ packets destined to $c$ in the pair $(c,c')$.

Then, the evolution of these queues' sizes can be described by (all the unspecified queue-lengths are always zero):

$$Q_j^{(f)}[t+1] = Q_j^{(f)}[t] + A_{in(j)}^{(f)}[t] + Y_{in(j)}^{(f)}[t] + S_{in(j)}^{(f)}[t] - S_{out(j)}^{(f)}[t], \qquad \forall j \in \mathcal{N}_f \backslash d(f) \quad (16)$$

$$Q_j^{rem(c)}[t+1] = Q_j^{rem(c)}[t] + X_{in(j)}^{rem(c)}[t] + S_{in(j)}^{rem(c)}[t] - S_{out(j)}^{rem(c)}[t], \qquad \forall j \neq c \quad (17)$$

$$Q_j^{(c,(f,g),(c,c'))}[t+1] = Q_j^{(c,(f,g),(c,c'))}[t] + S_{in(j)}^{(c,(f,g),(c,c'))}[t] - S_{out(j)}^{(c,(f,g),(c,c'))}[t], \forall f,g \in \mathcal{F}, \text{ and } c \in \mathcal{N}_g, c' \in \mathcal{N}_f, \quad (18)$$

where $A_{in(j)}^{(f)}[t] := A^{(f)}[t]\mathcal{I}_{j=s(f)}$, with $\mathcal{I}_E$ representing the indicator of event $E$, denotes the exogenous arrivals of flow $f$ packets into node $j$;

$$Y_{in(j)}^{(f)}[t] := \sum_{i:(i,j)\in\mathcal{E}} \sum_{g\in\mathcal{F}} \sum_{c'\in\mathcal{N}_g} S_{(i,j)}^{((f,g),(j,c'))}[t]$$

denotes the number of $(f,g)$ packets entering node $j$ that are multicast for decoding at nodes $(j,c')$ due to an intersession operation performed at an upstream node (in particular, node $j$ will recover the flow $f$ packet and place it into $Q_j^{(f)}$; and node $c'$ will recover the flow $g$ packet and place it into $Q_{c'}^{(g)}$); $S_{in(j)}^{(f)}[t] := S_{(pt^{(f)}(j),j)}[t]$ denotes the number of flow $f$ packets that node $j$ receives from its parent node;

$$S_{out(j)}^{(f)}[t] := S_{(j,ch^{(f)}(j))}[t] + \sum_{\{g\in\mathcal{F}:j\in\mathcal{N}_g\}} \sum_{\{k:(j,k)\in\mathcal{E}_f\cap\mathcal{E}_g\}} \sum_{b\in\mathcal{U}_j^{(f)}} \sum_{b'\in\mathcal{U}_j^{(g)}} S_{(j,k)}^{((f,g),(b,b'),(c,c'))}[t]$$

denotes the total number of flow $f$ packets served at node $j$ with the first term counting the number of flow $f$ packets that are forwarded by node $j$ to its child node and the second multiple sum counting the inter-session coded packets that are newly generated at $j$ and forwarded to its child node.;

$$X_{in(j)}^{rem(c)}[t] := \sum_{\{f\in\mathcal{F}:j\in\mathcal{N}_f\}} \sum_{m\in\mathcal{D}_j^{(f)}} \sum_{\{g\in\mathcal{F}:(m,ch^{(f)}(m))\in\mathcal{E}_g\}} \sum_{b'\in\mathcal{U}_m^{(g)}} \sum_{c'\in\mathcal{D}_{ch^{(f)}(m)}^{(f)}} S_{(m,ch^{(f)}(m))}^{((f,g),(j,b'),(c,c'))}[t]$$

denotes the number of remedy packets newly generated at node $j$ due to intersession coding operations at one of its downstream nodes;

$$S^{rem(c)}_{in(j)}[t] := \sum_{i:(i,j)\in\mathcal{E}} S^{rem(c)}_{(i,j)}[t]$$

denotes the number of remedy packets that are indigenously routed from a neighbor of $j$ into $j$;

$$S^{rem(c)}_{out(j)}[t] := \sum_{k:(j,k)\in\mathcal{E}} S^{rem(c)}_{(j,k)}[t]$$

denotes the number of remedy packets that are indigenously routed from $j$ to a neighbor of $j$;

$$S^{(c,(f,g),(c,c'))}_{in(j)}[t] := \sum_{i:(i,j)\in\mathcal{E}} S^{((f,g),(c,c'))}_{(i,j)}[t] + \sum_{\{i:(i,j)\in\mathcal{E}_f\cap\mathcal{E}_g\}} \sum_{b\in\mathcal{U}^{(f)}_i} \sum_{b'\in\mathcal{U}^{(g)}_i} S^{((f,g),(b,b'),(c,c'))}_{(i,j)}[t]$$

denotes the number of $(f,g)$ coded packets entering node $j$ with the first sum counting the number of intra-session coded packets entering $j$ and the second multiple sum counting the inter-session coded packets entering $j$ that are newly generated at its neighbors;

$$S^{(c,(f,g),(c,c'))}_{out(j)}[t] := \sum_{k:(j,k)\in\mathcal{E}} S^{((f,g),(c,c'))}_{(j,k)}[t]$$

denotes the number of $(f,g)$ intra-session coded packets leaving node $j$. We note that, while the routes of the original flows are fixed, the routes for the remedy packets are established dynamically by the algorithm we will propose.

We now describe our policy for performing routing, scheduling, and coding decisions for each link. The intuition behind the policy is that it uses the knowledge of queue-lengths to decide whether coding operations are feasible. Also, compared to the algorithm of Definition 2, this algorithm must also dynamically find routes for the remedy packets.

*Definition 4 (Routing-Scheduling-Coding (RSC) Algorithm):* At every time slot $t$, for each link $(m,n)\in\mathcal{E}$ with an ON state, four sets of weights, provided next, are computed at node $m$ as simple functions of queue-lengths: the first two corresponding to the routing of original flows and the remedy flows, respectively; and the last two corresponding to intra and inter-session coding operations, respectively.

$$\begin{aligned}
\rho^{(f)}_{(m,n)}[t] &\triangleq \left(Q^{(f)}_m[t] - Q^{(f)}_n[t]\right), \qquad \forall f\in\{f\in\mathcal{F}:(m,n)\in\mathcal{E}_f\}, && (19)\\
\xi^{rem(c)}_{(m,n)}[t] &\triangleq \left(Q^{rem(c)}_m[t] - Q^{rem(c)}_n[t]\right), \qquad \forall c\in\mathcal{N}\backslash\{m\}, && (20)\\
\chi^{((f,g),(c,c'))}_{(m,n)}[t] &\triangleq \left[Q^{(c,(f,g),(c,c'))}_m[t] - \left(Q^{(c,(f,g),(c,c'))}_n[t] + Q^{(g)}_n[t]\mathcal{I}_{n=c}\right)\right] && (21)\\
&\quad + \left[Q^{(c',(f,g),(c,c'))}_m[t] - \left(Q^{(c',(f,g),(c,c'))}_n[t] + Q^{(f)}_n[t]\mathcal{I}_{n=c'}\right)\right], \quad \forall f,g\in\mathcal{F}; c\in\mathcal{N}_g; c'\in\mathcal{N}_f, &&\\
\sigma^{((f,g),(b,b'),(c,c'))}_{(m,n)}[t] &\triangleq \left[Q^{(f)}_m[t] - \left(Q^{(c',(f,g),(c,c'))}_n[t] + Q^{rem(c)}_b[t]\right)\right] && (22)\\
&\quad + \left[Q^{(g)}_m[t] - \left(Q^{(c,(f,g),(c,c'))}_n[t] + Q^{rem(c')}_{b'}[t]\right)\right],&&\\
&\qquad \forall f,g\in\mathcal{F}; b\in\mathcal{U}^{(f)}_m; c'\in\mathcal{D}^{(f)}_n; b'\in\mathcal{U}^{(g)}_m; c\in\mathcal{D}^{(g)}_n. &&
\end{aligned}$$

Here, $\rho^{(f)}_{(m,n)}[t]$ represents the weight associated with serving uncoded packets of flow $f$ over link $(m,n)$; $\xi^{rem(c)}_{(m,n)}[t]$ represents the weight associated with serving remedy packets over $(m,n)$ that are destined to node $c$; $\chi^{((f,g),(c,c'))}_{(m,n)}[t]$ represents the weight associated with performing intra-session coding between already coded $(f,g)$ packets with destination pair $(c,c')$; and $\sigma^{((f,g),(b,b'),(c,c'))}_{(m,n)}[t]$ is the weight associated with performing intersession coding between $f$ and $g$ packets with remedies created at $(b,b')$, and decoding to be performed at $(c,c')$ (see Figure 4). Also, we introduce the following notation to denote the maximum weight for each category and the maximizing parameters.

$$\rho^\star_{(m,n)}[t] \triangleq \max_{f\in\mathcal{F}} \rho^{(f)}_{(m,n)}[t], \qquad \chi^\star_{(m,n)}[t] \triangleq \max_{((f,g),(c,c'))} \chi^{((f,g),(c,c'))}_{(m,n)}[t],$$

$$f^\star_{(m,n)}[t] \triangleq \arg\max_{f\in\mathcal{F}} \rho^{(f)}_{(m,n)}[t], \qquad ((f,g)^\star,(c,c')^\star)_{(m,n)}[t] \triangleq \arg\max_{((f,g),(c,c'))} \chi^{((f,g),(c,c'))}_{(m,n)}[t],$$

$$\xi^\star_{(m,n)}[t] \triangleq \max_{c\in\mathcal{N}} \xi^{rem(c)}_{(m,n)}[t], \qquad \sigma^\star_{(m,n)}[t] \triangleq \max_{((f,g),(b,b'),(c,c'))} \sigma^{((f,g),(b,b'),(c,c'))}_{(m,n)}[t],$$

$$c^\star_{(m,n)}[t] \triangleq \arg\max_{c\in\mathcal{N}} \xi^{rem(c)}_{(m,n)}[t], \qquad ((f,g)^\star,(b,b')^\star,(c,c')^\star)_{(m,n)}[t] \triangleq \arg\max_{((f,g),(b,b'),(c,c'))} \sigma^{((f,g),(b,b'),(c,c'))}_{(m,n)}[t],$$

Let us denote[7] the overall weight of link $(m,n)$, denoted $\omega^\star_{(m,n)}[t]$ as the maximum of these four weights, i.e.,

$$\omega^\star_{(m,n)}[t] \triangleq \max\left(\rho^\star_{(m,n)}[t], \xi^\star_{(m,n)}[t], \chi^\star_{(m,n)}[t], \sigma^\star_{(m,n)}[t]\right).$$

The final decision of which packet to transmit over $(m,n)$ is given based on the maximizer of this expression, where potential ties between these cases are broken uniformly at random. In particular, we have five cases:

- DO NOT SERVE ANY PACKET: If $\omega^\star_{(m,n)}[t] \leq 0$, then do not serve any packet over link $(m,n)$ in slot $t$.
- SERVE AN UNCODED PACKET: If $\omega^\star_{(m,n)}[t] = \rho^\star_{(m,n)}[t]$, then transmit the Head-of-Line (HOL) packet of the queue $Q_m^{(f^\star_{(m,n)}[t])}$. If there are no packets in that queue, do not transmit. In other words, set $M_{(m,n)}^{(f^\star_{(m,n)}[t])}[t] = 1$.
- SERVE A REMEDY PACKET: If $\omega^\star_{(m,n)}[t] = \xi^\star_{(m,n)}[t]$, then transmit the HOL packet of the queue $Q_m^{rem(c^\star_{(m,n)}[t])}$. If there are no packets in that queue, do not transmit. In other words, set $M_{(m,n)}^{rem(c^\star_{(m,n)}[t])}[t] = 1$.
- SERVE AN INTRA-SESSION CODED PACKET: If $\omega^\star_{(m,n)}[t] = \chi^\star_{(m,n)}[t]$, then perform random linear network coding on the HOL packets of the queues $Q_m^{((c)^\star_{(m,n)},((f,g)^\star,(c,c')^\star)_{(m,n)}[t])}$ and $Q_m^{((c')^\star_{(m,n)},((f,g)^\star,(c,c')^\star)_{(m,n)}[t])}$ and transmit the resulting packet. If one of the queues is empty, then transmit the HOL packet of the other queue without coding. If both queues are empty, do not transmit. This equivalent to setting $M_{(m,n)}^{(((f,g)^\star,(c,c')^\star)_{(m,n)}[t])}[t] = 1$.
- SERVE AN INTER-SESSION CODED PACKET: If $\omega^\star_{(m,n)}[t] = \sigma^\star_{(m,n)}[t]$, then XOR the HOL packets of the queues $Q_m^{(f)^\star_{(m,n)}}$ and $Q_m^{(g)^\star_{(m,n)}}$, transmit the resulting packet to be destined to $(c,c')^\star_{(m,n)}[t]$, and signal the node pair $(b,b')^\star_{(m,n)}[t]$ to generate the remedy packets to be transmitted from $(b)^\star_{(m,n)}[t]$ to $(c)^\star_{(m,n)}[t]$ for flow $(f)^\star_{(m,n)}[t]$ and from $(b')^\star_{(m,n)}[t]$ to $(c')^\star_{(m,n)}[t]$ for flow $(g)^\star_{(m,n)}[t]$. If any one of the two queues is empty, perform the encoding with a *dummy* packet whose content is all zeros. If both queues are empty, do not transmit. This corresponds to setting $M_{(m,n)}^{((f,g)^\star,(b,b')^\star,(c,c')^\star)_{(m,n)}[t]}[t] = 1$.

This completes the description of the RSC Algorithm. ◇

Notice that the RSC Algorithm inherits the essential characteristics of the algorithm for the butterfly network described in Definition 2. In particular, it utilizes a form of differential backlog expressions to decide whether to perform coding or just routing. The differential backlog expressions indicate the success of prior decisions and hence determine whether the link should continue performing those decisions. However, the RSC Algorithm also possesses new components that do not appear in the butterfly setting. Next, we comment on the key differences:

- In the butterfly network, the routes of the remedy packets were fixed due to the knowledge of the topology. In the general network topology, these routes need to be dynamically established in an optimal fashion. To that end, the queueing architecture is designed to distinguish original flow packets with fixed routes from remedy packets without known routes.
- In the butterfly network (see Figure 1), the candidate remedy nodes, i.e. $(b,b')$, and decoding nodes, i.e. $(c,c')$, are apriori known, which obviates the need for a wider search of butterfly structures. In a general multihop network, the optimal choice of remedy and decoding nodes are to be selected. Figure 4 reveals the possibility of having remedy and decoding nodes other than at the source and destination nodes. Thus, the RSC Algorithm also contains rules for the optimal selection of remedy and decoding nodes.
- In the butterfly network, there is no need to consider the possibility of intra-session network coding since the paths of the $(f,g)$ coded packets from node $n$ to the two destinations $(c,c')$ are fixed and disjoint (cf. Figure 1).

[7] We also note that potential ambiguity between the notations $f^\star_{(m,n)}$ and $c^\star_{(m,n)}$ will be avoided by always using $f,g$ for flows and $c,c'$ for the destination of remedy packets.

In the general scenario, the subgraph to support the multicast session from $n$ to $(c, c')$ (cf. Figure 4) must be dynamically established and is not necessarily a tree with disjoint paths. As the $(f, g)$ coded packets with destinations $(c, c')$ traverse the network, intra-session coding operations need to be performed across them in order to exploit the network coding advantage within a multicast session. Thus, the RSC Algorithm also contains rules for such decisions.

In the next section, we will show that the achievable rate region of our RSC Algorithm contains $\Lambda$ described by Definition 5 in Appendix I, i.e., it stabilizes all the queues as long as the mean flow arrival rates lie inside $\Lambda$.

### *B. Stochastic Analysis*

The main result of this paper is provided in the following theorem, which establishes that our RSC Algorithm stabilizes the network queues for any throughput within $\Lambda$ by dynamically searching for butterfly opportunities.

*Theorem 1:* Let $\mathcal{F}$ denote the set of unicast flows that enter at node $s(f) \in \mathcal{N}$ and that are destined to node $d(f) \in \mathcal{N}$ for each $f \in \mathcal{F}$. We assume that associated with each flow, say $f$, there is a fixed route with $\mathcal{N}_f$ and $\mathcal{E}_f$ that respectively denotes the set of nodes and the set of links that the flow traverses. Associated with each flow $f$, there is an arrival process $\{A^{(f)}[t]\}_t$ that is assumed to be i.i.d over $t$ with mean $\{\lambda^{(f)}\}_f$ and bounded second moment. Also, assume that for some $\varepsilon > 0$, the set of mean flow rates $\{\lambda^{(f)}\}_f$ are such that $\{\lambda^{(f)} + \varepsilon\}_f$ lies in the rate region $\Lambda$ that is described in Definition 5 (see Appendix I). Then the system under the RSC Algorithm (Definition 4) is stable.

*Proof:* The proof of the above theorem is given in the Appendix III. ■

Next, we remark on the nature of RSC Algorithm as it relates to earlier dynamic algorithms, and discuss several issues related to the practical implementation of the policy.

- In the proposed policy, the link weights $(\omega^{\star}_{(m,n)}[t])(m, n)$ is a substantial generalization of the concept of *differential backlog* introduced in [16]. In many earlier works (e.g. [16], [28], [29], [30], [21], [31], [22]), which do not consider inter-session coding possibilities, this term dynamically establishes routes by steering packets in the largest differential backlog direction. However, the availability of inter/intra-session coding decisions necessitates a novel improvement to the link weight definition that provides significant insight. In particular, the proposed weight computation contains two new differential backlog terms computed in (21) and (22) that respectively measure the decodability of intra-session and inter-session coded packets that traverse link $(m, n)$. This improvement also marks a drastic conceptual novelty: that properly maintained queue-lengths not only measures congestion levels, but can also measure sophisticated constraints such as end-to-end decodability.
- The policy requires the knowledge of the occupancy levels of those nodes at which decoding and remedy packet generation is to be performed. In practice, such information may be available only for those nodes in a local neighborhood of each node. Also, the search space of possible remedy and decoding nodes scales as $O(m^4)$, where $m$ is the number of potential nodes in the decision space. Although the performance of the policy will improve as the span of the search space increases, it has been observed in empirical studies [32] that even a one-hop neighborhood knowledge and search improves the achievable throughput considerably. Our model is general enough to accommodate the extreme scenarios of more practical implementation with weaker but still good performance, and less practical implementation with better performance. We also note that the complexity associated with the maintenance of the proposed multi-dimensional queueing architecture and the overhead associated with the dissemination of relevant queue-length information also arise in earlier dynamic policies that do not exploit inter-session coding opportunities (e.g. [16], [28], [29], [30], [21], [31], [22]). In this sense, our policy expands the policy space of dynamic policies in an orthogonal dimension to provide a means of using appropriate queue-length information to capture inter-session coding opportunities. Accordingly, the improvements in complexity (e.g. [21], [33]) that have been suggested for earlier dynamic policies are also applicable to our policy.
- It is well-known that adaptive routing capability of queue-length-based policies may cause large delays, especially for under-loaded networks as packets incessantly seek new routes to the destinations while the queues do not grow sufficiently to accurately indicate the congestion levels to the destinations. In our setup, we have assumed the presence of fixed routes for unicast flows, which partially prevents the complications associated with this

issue. Specifically, when the load of the network is low, routing over the given fixed routes will suffice to support the traffic, hence avoiding extra delay associated with dynamic routing. Only when the load exceeds the level supportable by routing will inter-session coding be performed, and dynamic routing for remedy flows and inter-session coded flows will be needed. Fortunately, in this regime, the recent developments (e.g. [34], [35], [36]) that yield delay-aware dynamic routing for queue-length-based policies are directly applicable. Thus, similar to the complexity issue, the delay issue associated with dynamic policies appears orthogonal to the inter-session coding capabilities investigated in this work.

## VI. General Wireless Networks

In this section, our goal is to provide the essential elements in the extension of our RSC algorithm together with the corresponding stability analysis. It will be seen that our approach can easily be extended to cover the challenging scenario of wireless communication. For wireless networks, the characterization of $\Lambda$ is given in Appendix II (see Definition 6).

### A. On the Dynamic Algorithm Description and Analysis

In the wireless case, we modify the RSC algorithm, as specified by Definition 4, as follows. For each hyperedge, $(m,N)\in\mathcal{E}$, we compute the following weights:

$$\rho^{(f)}_{(m,N)}[t] \triangleq \left(Q^{(f)}_m[t]-Q^{(f)}_n[t]\right)\mathcal{I}_{(m,n)\in N}, \qquad \forall f\in\{f\in\mathcal{F}:(m,n)\in\mathcal{E}_f\}, \tag{23}$$

$$\xi^{rem(c)}_{(m,N)}[t] \triangleq \left(Q^{rem(c)}_m[t]-\min_{(m,n)\in N} Q^{rem(c)}_n[t]\right), \qquad \forall c\in\mathcal{N}\backslash\{m\}, \tag{24}$$

$$\begin{aligned}\chi^{((f,g),(c,c'))}_{(m,N)}[t] \triangleq & \sum_{(m,n)\in N}\left[Q^{(c,(f,g),(c,c'))}_m[t]-\left(Q^{(c,(f,g),(c,c'))}_n[t]+Q^{(g)}_n[t]\mathcal{I}_{n=c}\right)\right] \\ & + \left[Q^{(c',(f,g),(c,c'))}_m[t]-\left(Q^{(c',(f,g),(c,c'))}_n[t]+Q^{(f)}_n[t]\mathcal{I}_{n=c'}\right)\right], \quad \forall f,g\in\mathcal{F};c\in\mathcal{N}_g;c'\in\mathcal{N}_f,\end{aligned} \tag{25}$$

$$\begin{aligned}\sigma^{((f,g),(b,b'),(c,c'))}_{(m,N)}[t] \triangleq & \sum_{(m,n)\in N}\left[Q^{(f)}_m[t]-\left(Q^{(c',(f,g),(c,c'))}_n[t]+Q^{rem(c)}_b[t]\right)\right] \\ & + \left[Q^{(g)}_m[t]-\left(Q^{(c,(f,g),(c,c'))}_n[t]+Q^{rem(c')}_{b'}[t]\right)\right], \\ & \forall f,g\in\mathcal{F};b\in\mathcal{U}^{(f)}_m;c'\in\mathcal{D}^{(f)}_n;b'\in\mathcal{U}^{(g)}_m;c\in\mathcal{D}^{(g)}_n.\end{aligned} \tag{26}$$

The definitions for $\rho^\star_{(m,N)}, \xi^\star_{(m,N)}, \chi^\star_{(m,N)}$ and $\sigma^\star_{(m,N)}$ are similar to those in Section V-A. In addition, let

$$W_{(m,N)}[t]\triangleq\max\left(\rho^\star_{(m,N)}[t],\xi^\star_{(m,N)}[t],\chi^\star_{(m,N)}[t],\sigma^\star_{(m,N)}[t]\right)$$

denote the weight of the hyperedge $(m,N)$, and find hyperedge rate vector $\mathbf{M}^\star[\mathbf{t}]$ every time-slot such that

$$\mathbf{M}^\star[t]\in\arg\max_{\mathbf{M}[t]\in\Theta}\sum_{(m,N)\in\mathcal{E}} M_{(m,N)}[t]W_{(m,N)}[t]. \tag{27}$$

This corresponds to picking the maximum weight hyperedge rates with respect to the weights $\mathbf{W}[\mathbf{t}]$ from within the set of feasible hyperedge rates $\Theta$ available in that time slot. Depending on the weights, inter-session coding is or is not performed according to Definition 4. In wireless networks, however, we generally cannot assume that one packet is transmitted on every link. Rather, $M^\star_{(m,N)}[t]$ packets are transmitted on link $(m,N)$.

Solving (27) exactly to determine $M^\star_{(m,N)}[t]$ may in general be difficult. The structure of $\Gamma[t]$ is determined by how the physical layer of the wireless network is implemented, and it may be difficlt to compute (27) or even to know $\Gamma[t]$. There are numerous distributed methods that can be used to reduce the complexity associated with this optimization while sacrificing from rate-of-convergence and/or maximal throughput performance (see e.g. [21], [37], [38], [39], [40], [41], [42]). These mechanisms can directly be applied to our setting to enable the practical use of our RSC algorithm in wireless networks.

The analysis of the RSC algorithm for wireless networks proceeds along the same lines as that for wireline networks, whereby we use the achievable rate region $\Lambda$ given by Definition 6 in Appendix II to show that the mean drift of a quadratic Lyapunov function is negative outside a bounded set of queue states. The details of this derivation is omitted since it does not add any new insight.

## VII. Simulations

In this section, we provide simulations of our dynamic routing-scheduling-coding strategy when implemented in the butterfly network of Figure 1 and the wireless multi-hop butterfly shown in Figure 7. In both the networks, two flows, namely $f$ and $g$, share the resources of the network: Flow $f$ enters the network at node $b$ destined for $c'$; Flow $g$ enters the network at node $b'$ destined for $c$ (see Figures 1 and 7). In the simulations, we set the number of arrivals for flows $f$ and $g$ to be Poisson distributed with means $\lambda^{(f)}$ and $\lambda^{(g)}$, respectively, in each time slot. Also, the link states are taken to be Bernoulli distributed with mean $\gamma_{(m,n)}$ for link $(m,n) \in \mathcal{E}$.

Our goals are: to confirm the stabilizing nature of our algorithm as suggested by our stochastic analysis; to investigate the throughput gains of inter-session coding to traditional routing-scheduling strategies by comparing our RSC with traditional back-pressure (BP) strategies; and to observe the adaptive nature of our RSC algorithm in making routing and coding decisions by using queue-length information. Using the multi-hop wireless butterfly (see Figure 7), we also demonstrate the local coding capability of COPE as opposed to a wider coding ability of RSC scheme leading to the RSC strategy potentially supporting a larger throughput region. To address these issues, we provide a set of simulations under different scenarios.

*a) Error-free Butterfly:* We first consider the scenario of the butterfly with no link failures. For this case, we can see from Definition 1 that we have

$$\Gamma_{BF}^{R} = \{\lambda^{(f)} \geq 0, \lambda^{(g)} \geq 0 : \lambda^{(f)} + \lambda^{(g)} \leq 1\}.$$

Similarly, from Definition 2, we have

$$\Gamma_{BF}^{C} = \{\lambda^{(f)} \geq 0, \lambda^{(g)} \geq 0 : \lambda^{(f)} \leq 1, \lambda^{(g)} \leq 1\}.$$

These regions are depicted in Figure 2. To facilitate illustrations, we present the simulation results for the symmetric arrival scenario where $\lambda^{(f)} = \lambda^{(g)} = \lambda$ for varying values of $\lambda \in (0,1)$. It is well-known (see, for example, [43], [28], [19]) that traditional back-pressure (BP) policy stabilizes its queues for any arrival rate that lies strictly within $\Gamma_{BF}^{R}$, and hence is called *throughput-optimal* in this sense. However, the BP policy only makes scheduling and routing decisions using the differential backlog parameter $\rho$ of Section III, and does not allow coding operations. Therefore, BP cannot stabilize arrival rates that lie inside $\Gamma_{BF}^{C} \backslash \Gamma_{BF}^{R}$.

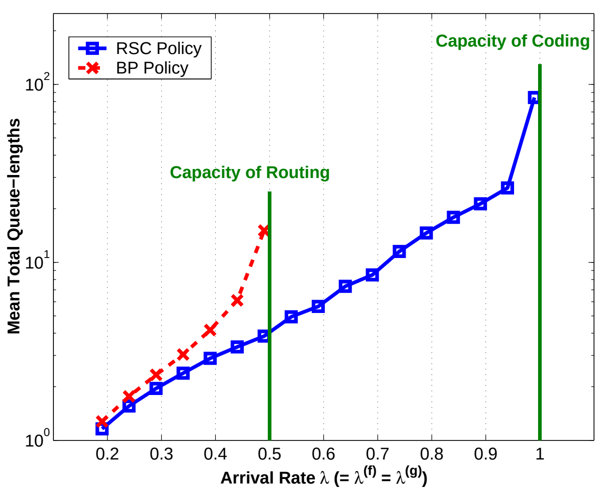


Fig. 5. Mean total queue-length under RSC and BP policies for the butterfly network with symmetric arrivals.

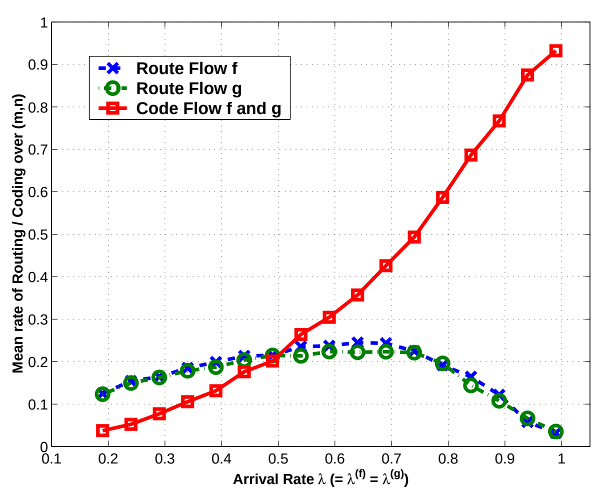


Fig. 6. Mean rate of flow allocated to routing and coding decisions at node $m$ for the butterfly network of Figure 1.

In Figure 5, we plot the mean total queue-length levels of the BP and RSC policies for varying values of $\lambda$. This simulation also confirms the throughput-optimality of our RSC policy for the butterfly network with the ability of coding. For the RSC policy, it is interesting to study the average flow rate over link $(m, n)$ that is allocated to Flow $f$ and Flow $g$ packets as well as coded $f \oplus g$ packets. This is shown in Figure 6. We observe that the rate allocated to coded packets gradually increase as $\lambda$ increases, while the pure routing decisions occur at a decreasing rate. These simulations show that RSC dynamically takes advantage of the coding capability as the load of the system increases, while BP suffers from the inability to perform coding.

*b) Error-free Multi-hop Wireless Butterfly:* We consider the error-free multi-hop wireless butterfly shown in Figure 7, where the hyperlink $(n, \{c, c'\})$ is a broadcast link. To capture wireless communication limitations, we assume a primary interference model where no node can receive and transmit simultaneously in any given time slot and adjacent links cannot be active simultaneously in any given time slot. We assume that nodes $c$ and $c'$ cannot overhear the transmissions of nodes $b$ and $b'$. Hence, under COPE, node $n$ cannot perceive the coding opportunity and simply performs routing. However, under RSC strategy, coding occurs at node $m$ which actively sends remedy requests to nodes $b$ and $b'$, forcing the discovery of alternative multi-hop remedy routes to the destinations. Hence, the remedy packets are routed to nodes $c$ and $c'$ via nodes $p$ and $q$. This leads to a higher throughput region under the RSC strategy as seen in the Figure 8.

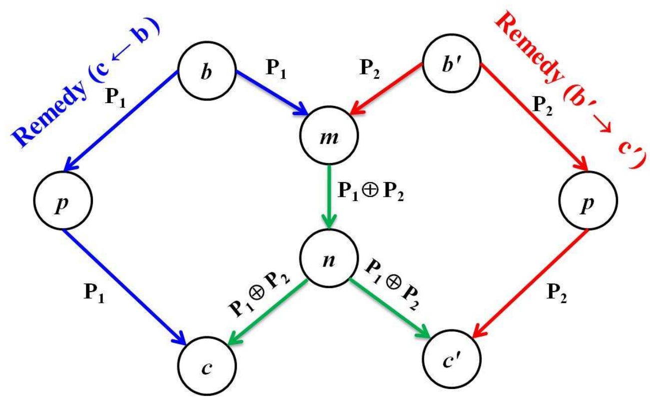


Fig. 7. The multi-hop wireless butterfly. Flows $f$ and $g$ begin at nodes $s(f) = b$ and $s(g) = b'$, respectively, and end at nodes $d(f) = c'$ and $d(g) = c$, respectively. Coding occurs at node $m$ and coded packets are broadcasted by node $n$ to nodes $c$ and $c'$

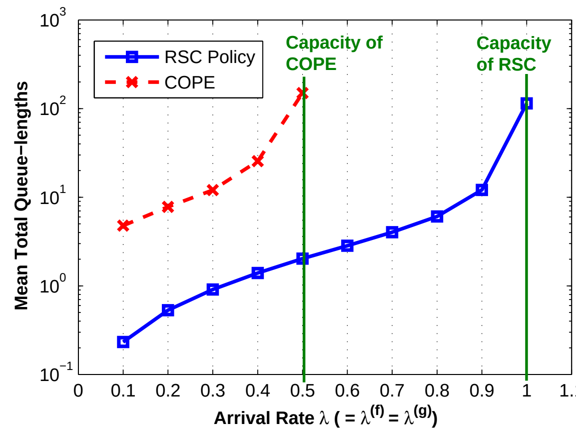


Fig. 8. Mean total queue-length under RSC and COPE policies for the multi-hop wireless butterfly network with symmetric arrivals

*c) Error-free Skeleton of the Butterfly:* Next, we consider the *skeleton of the butterfly network* where the side-links $(b, c)$ and $(b', c')$ in Figure 1 are completely disconnected, i.e. $\gamma_{(b,c)} = \gamma_{(b',c')} = 0$, while all the other links of the butterfly network are fully connected. This is an extreme scenario where the capacity regions of routing and coding are equal. Thus, network coding should not be performed at node $m$, since there exist no side links to convey the remedy packets necessary for decoding. Therefore, RSC is expected to adaptively stop coding and limit its transmissions to uncoded packets. This is exactly what we observe in Figure 9, which depicts the decision rule for the skeleton of the buttefly with varying symmetric arrival rates. We observe that node $m$ does not perform any coding operations and RSC achieves stability for rates inside the capacity region of the skeleton of the butterfly. A similar example can be easily constructed in the wireless butterfly scenario with similar performance.

*d) Asymmetric Butterfly with errors:* Up to now, we considered purely ON or OFF link qualities. In the following simulation, we consider $\gamma$ values that lie between $0$ and $1$. An interesting scenario is when link qualities and arrival rates are asymmetric. To that end, we assume the following link rates for the butterfly network of Figure 1: $\gamma_{(b',m)} = \gamma_{(n,c)} = 3/4$, $\gamma_{(b,m)} = \gamma_{(n,c')} = 1/2$, $\gamma_{(b,c)} = 3/4$, $\gamma_{(b',c')} = 1/3$, and $\gamma_{(m,n)} = 2/3$. Notice that the coded flow rate over link $(m, n)$ is limited by the link rate $\gamma_{(b',c')} = 1/3$. For this set of mean link rates, the capacity regions of routing and coding as described in Definitions 1 and 2 are depicted in Figure 10.

For this asymmetric network, we let $(\lambda^{(f)}, \lambda^{(g)}) = (1/3, 1/2)$, which is a rate that cannot be supported by mere routing. Figure 11 depicts the running average of the total queue-length levels in the system when our RSC algorithm is active, which shows that RSC quickly stabilizes the network queues. More interestingly, Figure 12 depicts the

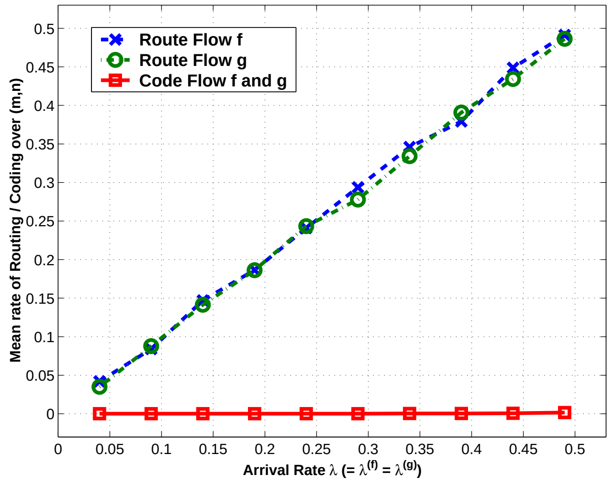


Fig. 9. Mean rate of flow allocated to routing and coding decisions for the skeleton of the buttefly at node $m$ in Figure 1.

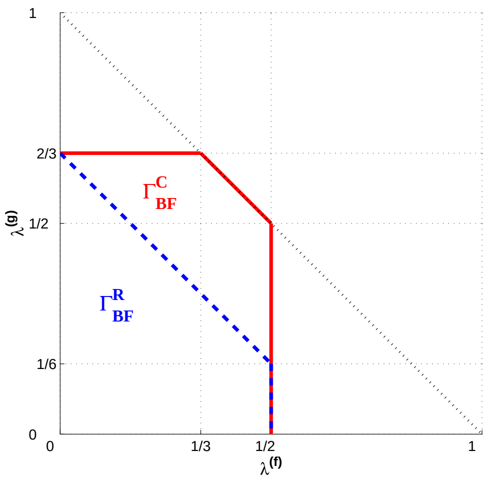


Fig. 10. Capacity regions under routing and coding strategies for the described asymmetric BF network.

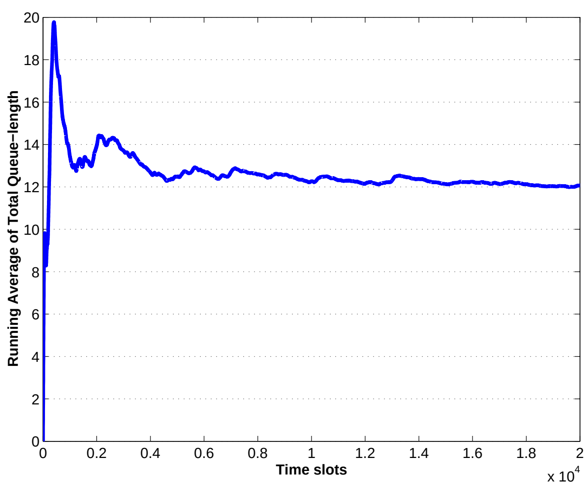


Fig. 11. Running average of the total queue-length level for the asymmetric BF network.

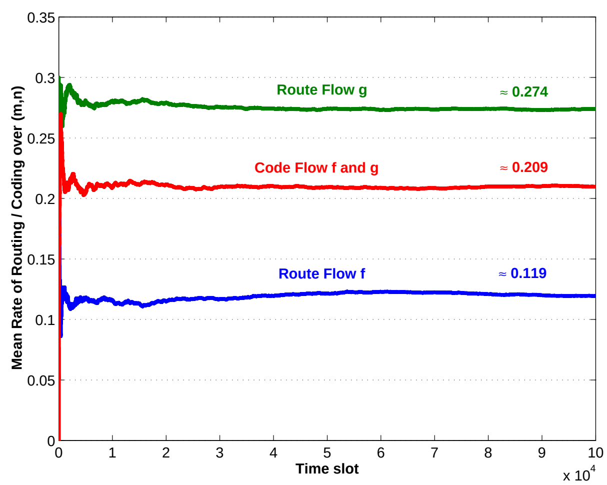


Fig. 12. Running average of the service rate provided to routing and coding over link $(m, n)$ for the asymmetric BF network.

running average of the rate of service provided to routing of the individual flows and the coded packets. We see that the RSC algorithm automatically converges to the routing/coding rates necessary to support the incoming traffic. This confirms the adaptive nature of the algorithm.

## VIII. Conclusions and Discussions

In this paper, we have introduced a dynamic routing-scheduling-coding strategy for inter-session network coding, which can be seen both as a generalization of dynamic routing-scheduling strategies based on differential backlog (e.g. [16], [20], [19]) and as an extension of COPE [9]. Our strategy decides whether independent flows should be coded together at a node and, if so, where and how. We did not consider allowing for coding operations that involve more than two flows at a time, but generalizing the ideas of this paper to allow for such coding operations is, at least conceptually, not difficult.

Our main result was to show that this strategy, called the RSC algorithm, stably supports any throughput that lies strictly within a known achievable rate region, $\Lambda$. Uncharacterized rate regions may simply have to be accepted to proceed with inter-session coding in a meaningful way since the general rate region for inter-session coding is shown to be very difficult to characterize [8], [44], and attempts to describe rate regions, such as $\Lambda$, have not yielded the gains observed in empirical studies (e.g., [9]). In this work, we have described a policy rather than a region, and it may be that any accurate characterization of achievable rates—especially in scenarios pertinent to practice—will have to come from practical implementations. The RSC algorithm, at any rate, is grounded in a solid theoretical framework and generalizes empirically studied strategies, such as COPE.

This work opens up numerous interesting avenues for future research. One immediate extension concerns the availability of multiple routes between the unicast source-destination pairs, which is a relatively easy extension. Another direction is the possibility of multi-cast sessions rather than unicast sessions. It is not difficult to see that our RSC algorithm can be easily extended to allow such multicasting. The performance analysis can also be performed along the same lines of argument once the achievable rate region $\Lambda$ is characterized for this scenario. In the backpressure literature, a direct connection has been identified (e.g. [21], [29]) between queue-lengths and congestion prices (or Lagrange multipliers of an associated optimization problem). It is of interest to identify such a connection between the extended version of queues that we introduced in this work and appropriately defined prices.

## Acknowledgments

The authors would like to thank Ralf Koetter, Muriel Médard, Asuman Ozdaglar, and Edmund Yeh for helpful discussions and comments.

## Appendix I
## Achievable Rate Region for General Wireline Networks

An achievable rate region for serving unicast traffic in a general network topology when inter-session linear coding is allowed has been described by Traskov et al. [11]. This region is obtained by exploiting coding opportunities of the "butterfly" type discussed in the previous section. The achievable rate region $\Lambda$ for wireline networks is as follows.

*Definition 5 ($\Lambda$ for wireline networks):* Assume that we are given a wireline network $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, a set of average link capacities $\{\gamma_{(i,j)}\}_{(i,j)\in\mathcal{E}}$, and a set $\mathcal{F}$ of flows with a beginning node $s(f) \in \mathcal{N}$ and a destination node $d(f) \in \mathcal{N}$ for flow $f \in \mathcal{F}$. We assign an arbitrary ordering to the flows in $\mathcal{F}$. Then, $\Lambda$ is the set of unicast flow rates $\{\lambda^{(f)}\}_{f\in\mathcal{F}}$

satisfying the following constraints for some $\{x_{(i,j)}^{(f)}\}$, $\{p_{(i,j)}^{(f\to g,l)}\}$, $\{r_{(i,j)}^{(f\to g,l)}\}$, and $\{v_{(i,j)}^{(f\to g,l)}\}$ [11]:

$$\sum_{j:(i,j)\in\mathcal{E}} x_{(i,j)}^{(f)} - \sum_{j:(j,i)\in\mathcal{E}} x_{(j,i)}^{(f)} = \begin{cases} \lambda^{(f)} & \text{if } i = s(f), \\ -\lambda^{(f)} & \text{if } i = d(f), \\ 0 & \text{otherwise}, \end{cases} \qquad \forall\ i \in \mathcal{N},\ f \in \mathcal{F}, \tag{28}$$

$$\begin{aligned} &\sum_{i:(i,j)\in\mathcal{E}} (p_{(i,j)}^{(f\to g,l)} + r_{(i,j)}^{(f\to g,l)} + v_{(i,j)}^{(f\to g,l)}) \\ &\qquad = \sum_{i:(j,i)\in\mathcal{E}} (p_{(j,i)}^{(f\to g,l)} + r_{(j,i)}^{(f\to g,l)} + v_{(j,i)}^{(f\to g,l)}), \qquad \forall\ j,l \in \mathcal{N},\ f,g \in \mathcal{F}, \end{aligned} \tag{29}$$

$$\sum_{i:(i,j)\in\mathcal{E}} p_{(i,j)}^{(f\to g,l)} - \sum_{i:(j,i)\in\mathcal{E}} p_{(j,i)}^{(f\to g,l)} \begin{cases} \geq 0 & \text{if } j = l, \\ \leq 0 & \text{otherwise}, \end{cases} \qquad \forall\ j,l \in \mathcal{N},\ f,g \in \mathcal{F}, \tag{30}$$

$$\sum_{i:(i,j)\in\mathcal{E}} v_{(i,j)}^{(f\to g,l)} - \sum_{i:(j,i)\in\mathcal{E}} v_{(j,i)}^{(f\to g,l)} \begin{cases} \leq 0 & \text{if } j = l, \\ \geq 0 & \text{otherwise}, \end{cases} \qquad \forall\ j,l \in \mathcal{N},\ f,g \in \mathcal{F}, \tag{31}$$

$$p_{(m,n)}^{(g\to f,m)} = p_{(m,n)}^{(f\to g,m)}, \qquad \forall\ (m,n) \in \mathcal{E},\ f,g \in \mathcal{F}, \tag{32}$$

$$\sum_{f\in\mathcal{F}} \left\{ x_{(i,j)}^{(f)} + \sum_{l\in\mathcal{N}} \sum_{g>f} p_{(i,j)}^{\max}(f,g,l) + \sum_{l\in\mathcal{N}} \sum_{g\neq f} r_{(i,j)}^{(f\to g,l)} \right\} \leq \gamma_{(i,j)}, \qquad \forall\ (i,j) \in \mathcal{E}, \tag{33}$$

$$x_{(i,j)}^{(g)} + \sum_{l\in\mathcal{N}} \sum_{f\neq g} (p_{(i,j)}^{(f\to g,l)} + v_{(i,j)}^{(g\to f,l)}) \geq 0, \qquad \forall\ (i,j) \in \mathcal{E},\ g \in \mathcal{F}, \tag{34}$$

$$\mathbf{p} \leq 0, \quad \mathbf{r} \geq 0, \quad \mathbf{v} \leq 0, \quad \mathbf{x} \geq 0,$$

where $p_{(i,j)}^{\max}(f,g,l) \triangleq \max(p_{(i,j)}^{(f\to g,l)}, p_{(i,j)}^{(g\to f,l)})$. $\diamond$

In this definition, the variables $\{x_{(i,j)}^{(f)}\}_{(i,j)\in\mathcal{E}}$ define flow $f$, which is a flow of size $\lambda^{(f)}$ going from $s(f)$ to $d(f)$. For any two flows $f$ and $g$, $\{p_{(i,j)}^{(f\to g,l)}\}_{(i,j)\in\mathcal{E}}$ is the *poison flow* of $f$ acting on $g$, which flows from where flow $g$ is remedied, or decoded, to the coding node $l$; $\{r_{(i,j)}^{(f\to g,l)}\}_{(i,j)\in\mathcal{E}}$ is the *remedy flow* of $f$ acting on $g$, which flows from where remedy packets are provided to where flow $g$ is remedied; and $\{v_{(i,j)}^{(f\to g,l)}\}_{(i,j)\in\mathcal{E}}$ is the *remedy request flow* of $f$ acting on $g$, which flows from the coding node $l$ to where remedy packets are provided. In Figure 13, we show these flows for the butterfly network of Figure 1.

Equations (28)–(29) ensure that $\mathbf{x}$, $\mathbf{p}$, $\mathbf{r}$, $\mathbf{v}$, indeed define these flows. Equations (30)–(31) ensure that $\mathbf{p}$ and $\mathbf{v}$ flow in and out, respectively, of the appropriate coding nodes. Equation (32) ensures that, at the point of coding, two flows being coded together introduce the same amount of poison to each other. When performing inter-session coding, the poison flow has the effect of removing flow from $g$ from the network, while the remedy flow adds additional flow to the network. Equation (33) ensures that there is sufficient capacity in the network given the effect of poision and remedy flows. Lastly, equation (34) ensures that poison and request flows follow the appropriate uncoded flows from which traffic is removed and remedy packets are provided. Readers interested in more detail are referred to [11].

## Appendix II
## Achievable Rate Region for General Wireless Networks

Using the insights developed in [45], it is not difficult to see that $\Lambda$ extends to the wireless case as follows. The principal differences between the wireless and wireline cases are, first, that the broadcast capability of links is captured using hyperarcs, and second, that possible interference among links is captured by assuming a more complicated set of feasible link capacities $\gamma$. The interpretation of the variables defining the achievable rate region is the same as in Appendix I.

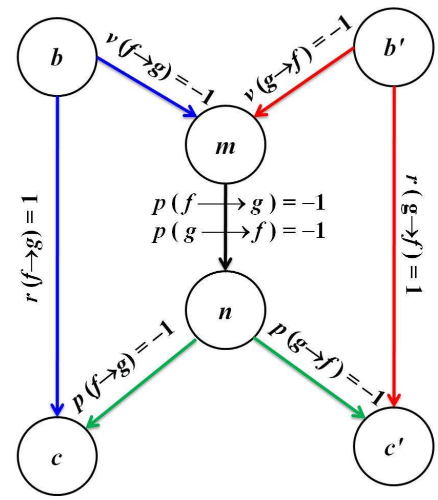


Fig. 13. Flow formulation of Traskov et al. [11] for the butterfly topology in Figure 1.

*Definition 6 ($\Lambda$ for wireless networks):* $\Lambda$ for a wireless network $\mathcal{H} = (\mathcal{N}, \mathcal{E})$ is the set of flow rates $\{\lambda^{(f)}\}$ satisfying the following constraints for some $\{x^{(f)}_{(m,N,n)}\}$, $\{p^{(f\to g,l)}_{(m,N,n)}\}$, $\{r^{(f\to g,l)}_{(m,N,n)}\}$, and $\{s^{(f\to g,l)}_{(m,N,n)}\}$:

$$\sum_{M:(n,M)\in\mathcal{E}} \sum_{m\in M} x^{(f)}_{(n,M,m)} - \sum_{m:(m,N)\in\mathcal{E}} \sum_{n\in N} x^{(f)}_{(m,N,n)} = \begin{cases} \lambda^{(f)} & \text{if } n = s(f), \\ -\lambda^{(f)} & \text{if } n = D(f), \\ 0 & \text{otherwise,} \end{cases} \qquad \forall\, n \in \mathcal{N},\, f \in \mathcal{F},$$

$$\sum_{m:(m,N)\in\mathcal{E}} \sum_{N\ni n} (p^{(f\to g,l)}_{(m,N,n)} + r^{(f\to g,l)}_{(m,N,n)} + s^{(f\to g,l)}_{(m,N,n)})$$
$$= \sum_{M:(n,M)\in\mathcal{E}} \sum_{m\in M} (p^{(f\to g,l)}_{(n,M,m)} + r^{(f\to g,l)}_{(n,M,m)} + s^{(f\to g,l)}_{(n,M,m)}), \qquad \forall\, l, n \in \mathcal{N},\, f, g \in \mathcal{F},$$

$$\sum_{m:(m,N)\in\mathcal{E}} \sum_{N\ni n} p^{(f\to g,l)}_{(m,N,n)} - \sum_{M:(n,M)\in\mathcal{E}} \sum_{m\in M} p^{(f\to g,l)}_{(n,M,m)} \begin{cases} \geq 0 & \text{if } n = l, \\ \leq 0 & \text{otherwise,} \end{cases} \qquad \forall\, l, n \in \mathcal{N},\, f, g \in \mathcal{F},$$

$$\sum_{m:(m,N)\in\mathcal{E}} \sum_{N\ni n} s^{(f\to g,l)}_{(m,N,n)} - \sum_{M:(n,M)\in\mathcal{E}} \sum_{m\in M} s^{(f\to g,l)}_{(n,M,m)} \begin{cases} \leq 0 & \text{if } n = l, \\ \geq 0 & \text{otherwise,} \end{cases} \qquad \forall\, l, n \in \mathcal{N},\, f, g \in \mathcal{F},$$

$$p^{(g\to f,m)}_{(m,N,n)} = p^{(f\to g,m)}_{(m,N,n)}, \qquad \forall\, (m, N) \in \mathcal{E},\, n \in N,\, f, g \in \mathcal{F},$$

$$\sum_{f\in\mathcal{F}} \left\{ \sum_{n\in N} x^{(f)}_{(m,N,n)} + \sum_{l} \sum_{g>f} p^{\max}_{(m,N)}(f, g, l) + \sum_{l} \sum_{f\neq g} \sum_{n\in N} r^{(f\to g,l)}_{(m,N,n)} \right\} \leq \gamma_{(m,N)}, \qquad \forall\, (m, N) \in \mathcal{E}, \tag{35}$$

$$x^{(g)}_{(m,N,n)} + \sum_{l} \sum_{f} (p^{(f\to g,l)}_{(m,N,n)} + s^{(g\to f,l)}_{(m,N,n)}) \geq 0, \qquad \forall\, (m, N) \in \mathcal{E},\, n \in N,\, g \in \mathcal{F}, \tag{36}$$

$$\mathbf{p} \leq 0, \quad \mathbf{r} \geq 0, \quad \mathbf{s} \leq 0, \quad \mathbf{x} \geq 0, \quad \gamma \in \text{Convex Hull}(\Theta), \tag{37}$$

where $p^{\max}_{(m,N)}(f, g, l) \triangleq \max\left( \sum_{n\in N} p^{(f\to g,l)}_{(m,N,n)}, \sum_{n\in N} p^{(g\to f,l)}_{(m,N,n)} \right)$. ⋄

## Appendix III
## Proof of Theorem 1

*Proof:* Our proof is based on stochastic stability methods that are commonly utilized in dynamic algorithm analysis (e.g. [43], [18], [19], [34]), whereby the mean drift of an appropriate Lyapunov function is studied and is shown to be negative except for a bounded state space.

To that end, we recall the evolutions of the queue-lengths given by (16) - (18) and note that the queue-lengths can be upper-bounded in terms of offered services $M_{\cdot}^{(\cdot)}[t]$ as follows:

$$\begin{aligned} Q_j^{(f)}[t+1] &\leq \left(Q_j^{(f)}[t] - M_{out(j)}^{(f)}[t]\right)^+ + A_{in(j)}^{(f)}[t] + Y_{in(j)}^{(f)}[t] + M_{in(j)}^{(f)}[t], \quad \forall j \in \mathcal{N}_f \backslash d(f) && (38)\\ Q_j^{rem(c)}[t+1] &\leq \left(Q_j^{rem(c)}[t] - M_{out(j)}^{rem(c)}[t]\right)^+ + X_{in(j)}^{rem(c)}[t] + M_{in(j)}^{rem(c)}[t], \quad \forall j \neq c && (39)\\ Q_j^{(c,(f,g),(c,c'))}[t+1] &\leq \left(Q_j^{(c,(f,g),(c,c'))}[t] - M_{out(j)}^{(c,(f,g),(c,c'))}[t]\right)^+ + M_{in(j)}^{(c,(f,g),(c,c'))}[t], \quad \forall f,g \in \mathcal{F}, && (40)\\ & \qquad \forall c \in \mathcal{N}_g, c' \in \mathcal{N}_f. \end{aligned}$$

Similarly, $Y_{in(j)}^{(f)}[t]$ and $X_{in(j)}^{rem(c)}[t]$ can be upper bounded in terms of offered services vector $\mathbf{M}[t]$. The definitions of the terms $M_{in(j)}^{(f)}[t]$, $M_{in(j)}^{rem(c)}[t]$, $M_{in(j)}^{(c,(f,g),(c,c'))}[t]$, $M_{out(j)}^{(f)}[t]$, $M_{out(j)}^{rem(c)}[t]$, and $M_{out(j)}^{(c,(f,g),(c,c'))}[t]$ is similar to that of their actual services counterparts in Section V-A.

We consider the quadratic Lyapunov function:

$$V(\mathbf{Q}) = \frac{1}{2}\left( \sum_{f\in\mathcal{F}} \sum_{j\in\mathcal{N}_f} (Q_j^{(f)})^2 + \sum_{f\in\mathcal{F}} \sum_{c\in\mathcal{N}_f} \sum_{j\in\mathcal{N}_f} (Q_j^{(rem(c))})^2 + \sum_{f\in\mathcal{F}} \sum_{g\in\mathcal{F}} \sum_{c\in\mathcal{N}_f} \sum_{c'\in\mathcal{N}_g} \sum_{j\in\mathcal{N}_f\cap\mathcal{N}_g} (Q_j^{(c,(f,g),(c,c'))})^2 \right)$$

We let $\Delta V(\mathbf{Q}) \triangleq E[V(\mathbf{Q}[t+1]) - V(\mathbf{Q}[t])|\mathbf{Q}[t] = \mathbf{Q}]$ to denote its conditional mean drift when the queue-length levels are $\mathbf{Q}$. Before we study the mean drift of the queue lengths for this Lyapunov function, we provide a bound that will be used in the analysis:

*Lemma 1:* Consider the queue-length evolution given by $Q[t+1] = (Q[t] - M[t])^+ + A[t] + Y[t]$ where $A[t]$ is a non-negative random variable with finite mean and finite second moment for every time-slot $t$. Also, $M[t]$ and $Y[t]$ are bounded non-negative random variables. Assume that, for every $t$, $M[t]$, $A[t]$, and $Y[t]$ are independent given $Q[t]$ and $A[t]$ is independent of $Q[t]$. Then the Lyapunov drift is bounded above as follows:

$$\Delta V(Q) \leq B_1 + Q\left(E[A[t] + Y[t] - M[t]]|Q[t] = Q]\right) \tag{41}$$

for some constant $B_1$.

*Proof:*

$$\begin{aligned} (Q[t+1])^2 - (Q[t])^2 &\leq \left((Q[t] - M[t])^+ + A[t] + Y[t]\right)^2 - (Q[t])^2 \\ &\leq (Q[t] - M[t] + U[t] + A[t] + Y[t])^2 - (Q[t])^2 \end{aligned}$$

where $U[t] = M[t] - Q[t]$ if $M[t] > Q[t]$ or $U[t] = 0$ otherwise. Note that, $U[t]$ is non-negative and can be bounded above by some constant $\hat{\eta}$. Therefore,

$$\begin{aligned} \Delta V(Q) &\leq (1/2)E\left[(Q[t] - M[t] + A[t] + Y[t])^2 - (Q[t])^2 | Q[t] = Q\right] + \hat{\eta}E\left[A[t] + Y[t]|Q[t] = Q\right] \\ &\leq \hat{\eta}E\left[A[t] + Y[t]|Q[t] = Q\right] + (1/2)E\left[(A[t] + Y[t] - M[t])^2|Q[t] = Q\right] + QE\left[A[t] + Y[t] - M[t]|Q[t] = Q\right] \\ &\leq \hat{\eta}E\left[A[t]\right] + \hat{\eta}E\left[Y[t]|Q[t] = Q\right] + (1/2)E\left[(A[t])^2\right] + (1/2)E\left[(Y[t] - M[t])^2|Q[t] = Q\right] \\ &\qquad + E\left[A[t]\right]E\left[(Y[t] - M[t])|Q[t] = Q\right] + Q[t]E\left[A[t] + Y[t] - M[t]|Q[t] = Q\right] \\ &\leq B_1 + QE\left[A[t] + Y[t] - M[t]|Q[t] = Q\right] \end{aligned}$$

where $B_1$ is a constant. The last inequality follows from the assumptions in the Lemma. ∎

Note that the terms in queue-length evolutions given in (38) - (40) satisfy the conditions of the above lemma and hence the drift can be bounded as in the lemma. For notational convenience, we drop the term $[t]$ in our derivations. Define $\mu_{(m,n)}^{(.)} = E[M_{(m,n)}^{(.)}|\mathbf{Q}]$. Then using the above lemma, the Lyapunov drift can be bounded above as follows:

$$\begin{aligned}\Delta V(\mathbf{Q}) \;\leq\; & B + \sum_{f\in\mathcal{F}} \lambda^{(f)} Q_{s(f)}^{(f)} - \sum_{(m,n)\in\mathcal{E}} \Bigg[ \sum_f \mu_{(m,n)}^{(f)} \left(Q_m^{(f)} - Q_n^{(f)}\right) + \sum_c \mu_{(m,n)}^{rem(c)} \left(Q_m^{rem(c)} - Q_n^{rem(c)}\right) \\ & + \sum_{(f,g),(c,c')} \mu_{(m,n)}^{((f,g),(c,c'))} \Bigg( \left(Q_m^{(c,(f,g),(c,c'))} - (Q_n^{(c,(f,g),(c,c'))} + Q_c^{(g)})\right) \\ & \qquad + \left(Q_m^{(c',(f,g),(c,c'))} - (Q_n^{(c',(f,g),(c,c'))} + Q_{c'}^{(f)})\right) \Bigg) \\ & + \sum_{(f,g),(b,b'),(c,c')} \mu_{(m,n)}^{((f,g),(b,b'),(c,c'))} \Bigg( Q_m^{(f)} - (Q_n^{(c',(f,g),(c,c'))} + Q_{b'}^{rem(c')}) \\ & \qquad + Q_m^{(g)} - (Q_n^{(c,(f,g),(c,c'))} + Q_b^{rem(c)}) \Bigg) \Bigg]\end{aligned}$$

where $B$ is a constant. The last equation follows from a substitution of the terms of the queue-length evolution as provided above, and the Lemma 1.

Notice that for any $(m,n) \in \mathcal{E}$, the instantaneous link capacity constraints ( Equation 15) force us to select the different service rates $\left(M_{(m,n)}^{(f)}, M_{(m,n)}^{rem(c)}, M_{(m,n)}^{((f,g),(c,c'))}, M_{(m,n)}^{((f,g),(b,b'),(c,c'))}\right)$ such that

$$\sum_f \mu_{(m,n)}^{(f)} + \sum_c \mu_{(m,n)}^{rem(c)} + \sum_{(f,g),(c,c')} \mu_{(m,n)}^{((f,g),(c,c'))} + \sum_{(f,g),(b,b'),(c,c')} \mu_{(m,n)}^{((f,g),(b,b'),(c,c'))} \leq \gamma_{(m,n)}. \tag{42}$$

It is not difficult to see that our RSC algorithm is designed to pick feasible $\mathbf{M}$ dynamically and distributively such that $\Delta V(\mathbf{Q})$ is minimized for each $\mathbf{Q}$.

Since $\{\lambda^{(f)} + \varepsilon\}$ lies in $\Lambda$, we have $\{x_{(m,n)}^{(f)}\}$, $\{p_{(m,n)}^{(f\to g,l)}\}$, $\{r_{(m,n)}^{(f\to g,l)}\}$, $\{s_{(m,n)}^{(f\to g,l)}\}$, and $\{\gamma_{(m,n)}\}$, which, together with $\{\lambda^{(f)} + \varepsilon\}$, satisfy the constraints provided in Definition 5 given in the section I. In general, $p^{(f\to g,l)}$, $r^{(f\to g,l)}$, and $s^{(f\to g,l)}$ form a flow that starts with $s$ at node $l$, changes to $r$, and ends with $p$ at node $l$ [11]. Using the conformal realization theorem (see, for example, [46, Proposition 1.1]), such a flow can be decomposed into the sum of a finite number of simple cycle flows. We assume, without loss of generality, that the flow formed by $p^{(f\to g,l)}$, $r^{(f\to g,l)}$, and $s^{(f\to g,l)}$ is a simple cycle flow; more general flows result from superpositions of this case. Specifically, we assume that $s^{(f\to g,l)}$ is a simple path flow from $l$ to $b^{(f\to g,l)}$, that $r^{(f\to g,l)}$ is a simple path flow from $b^{(f\to g,l)}$ to $D^{(f\to g,l)}$, and that $p^{(f\to g,l)}$ is a simple path flow from $D^{(f\to g,l)}$ to $l$. We denote the magnitude of these flows by $\lambda^{(f\to g,l)}$ and the node immediately preceding $l$ on simple path flow $p^{(f\to g,l)}$ by $k^{(f\to g,l)}$. For a pictorial demonstration, see Figure 13 in the section I, where we have $b^{(f\to g,l)} = b$, $D^{(f\to g,l)} = c$, $D^{(g\to f,l)} = c'$, and $k^{(g\to f,l)} = k^{(f\to g,l)} = n$ and $l = m$.

Let

$$\begin{aligned} W_{(m,n)} \triangleq \max \Bigg( & \max_f \left(Q_m^{(f)} - Q_n^{(f)}\right), \max_c \left(Q_m^{rem(c)} - Q_n^{rem(c)}\right), \\ & \max_{(f,g),(c,c')} \left(Q_m^{(c,(f,g),(c,c'))} - (Q_n^{(c,(f,g),(c,c'))} + Q_n^{(g)}\mathcal{I}_{n=c})\right) + \left(Q_m^{(c',(f,g),(c,c'))} - (Q_n^{(c',(f,g),(c,c'))} + Q_n^{(f)}\mathcal{I}_{n=c'})\right), \\ & \max_{(f,g),(b,b'),(c,c')} \left(Q_m^{(f)} - (Q_n^{(c',(f,g),(c,c'))} + Q_{b'}^{rem(c')}) + Q_m^{(g)} - (Q_n^{(c,(f,g),(c,c'))} + Q_b^{rem(c)})\right) \Bigg) \end{aligned}$$

be the weight associated with link $(m,n)$. Then, under the RSC algorithm, we have

$$\Delta V(\mathbf{Q}) \;\leq\; B + \sum_{f\in\mathcal{F}} \lambda^{(f)} Q_{s(f)}^{(f)} - \sum_{(m,n)\in\mathcal{E}} \gamma_{(m,n)} W_{(m,n)}, \tag{43}$$

which follows from the link rate constraint (42).

Next, we consider the term $\sum_{(m,n)\in\mathcal{E}} \gamma_{(m,n)} W_{(m,n)}$: using equation (33), we obtain

$$\sum_{(m,n)\in\mathcal{E}} \gamma_{(m,n)} W_{(m,n)} \quad \geq \quad \sum_{(m,n)\in\mathcal{E}} \sum_{f} \left\{ x_{(m,n)}^{(f)} + \sum_{l} \sum_{g>f} p_{(m,n)}^{\max}(f,g,l) + \sum_{l} \sum_{g\neq f} r_{(m,n)}^{(f\to g,l)} \right\} W_{(m,n)},$$

which then yields

$$\begin{aligned} \sum_{(m,n)\in\mathcal{E}} \gamma_{(m,n)} W_{(m,n)} \quad \geq \quad & \sum_{(m,n)\in\mathcal{E}} \sum_{f} x_{(m,n)}^{(f)} W_{(m,n)} & (44) \\ & + \sum_{(m,n)\in\mathcal{E}} \sum_{f} \sum_{l} \sum_{g\neq f} p_{(m,n)}^{(f\to g,l)} W_{(m,n)} & (45) \\ & - \sum_{(m,n)\in\mathcal{E}} \sum_{f} \sum_{l} \sum_{g>f} p_{(m,n)}^{\min}(f,g,l) W_{(m,n)} & (46) \\ & + \sum_{(m,n)\in\mathcal{E}} \sum_{f} \sum_{l} \sum_{g\neq f} r_{(m,n)}^{(f\to g,l)} W_{(m,n)}, & (47) \end{aligned}$$

where $p_{(m,n)}^{\min}(f,g,l) \triangleq \min(p_{(m,n)}^{(f\to g,l)}, p_{(m,n)}^{(g\to f,l)})$. In the last expansion, (45) and (46) follows from the fact that $p_{(m,n)}^{\max}(f,g,l) = p_{(m,n)}^{(f\to g,l)} + p_{(m,n)}^{(g\to f,l)} - p_{(m,n)}^{\min}(f,g,l)$.

Next, we study the terms (44)-(47): First, we consider $(44) + (45)$:

$$\begin{aligned} (44) + (45) \quad = \quad & \sum_{(m,n)\in\mathcal{E}} \sum_{f} \left( x_{(m,n)}^{(f)} + \sum_{l} \sum_{g\neq f} p_{(m,n)}^{(f\to g,l)} \right) W_{(m,n)} \\ \overset{(a)}{\geq} \quad & \sum_{(m,n)\in\mathcal{E}} \sum_{f} \left( x_{(m,n)}^{(f)} + \sum_{l} \sum_{g\neq f} p_{(m,n)}^{(f\to g,l)} \right) \left( \max_{f} \left( Q_m^{(f)} - Q_n^{(f)} \right) \right) \\ = \quad & \sum_{(m,n)\in\mathcal{E}} \sum_{f} x_{(m,n)}^{(f)} \left( Q_m^{(f)} - Q_n^{(f)} \right) & (48) \\ & + \sum_{(m,n)\in\mathcal{E}} \sum_{f} \sum_{l} \sum_{g\neq f} p_{(m,n)}^{(f\to g,l)} \left( Q_m^{(f)} - Q_n^{(f)} \right), & (49) \end{aligned}$$

where the inequality $(a)$ follows from the definition of $W_{(m,n)}$ and the fact that $x_{(m,n)}^{(f)} + \sum_l \sum_{g\neq f} p_{(m,n)}^{(f\to g,l)} \geq 0$ for all $(m,n) \in \mathcal{E}$ and $f \in \mathcal{F}$ since (34) holds and $\mathbf{s} \leq 0$.

Next, let us lower-bound the terms in (48) and (49). By equation (28) and the fact that $Q_{d(f)}^{(f)} = 0$ for all $f \in \mathcal{F}$, we have

$$\sum_{(m,n)\in\mathcal{E}} x_{(m,n)}^{(f)} \left( Q_m^{(f)} - Q_n^{(f)} \right) \quad = \quad (\lambda^{(f)} + \varepsilon) Q_{s(f)}^{(f)}. \tag{50}$$

Also, by the assumption that $p^{(f\to g,l)}$ is a simple path flow from $l$ to $D^{(f\to g,l)}$ of rate $-\lambda^{(f\to g,l)}$, we have

$$\begin{aligned} \sum_{(m,n)\in\mathcal{E}} p_{(m,n)}^{(f\to g,l)} \left( Q_m^{(f)} - Q_n^{(f)} \right) \quad &= \quad -\lambda^{(f\to g,l)} \left( Q_l^{(f)} - Q_{D^{(f\to g,l)}}^{(f)} \right) \\ &= \quad \lambda^{(f\to g,l)} \left( Q_{D^{(f\to g,l)}}^{(f)} - Q_l^{(f)} \right). & (51) \end{aligned}$$

Thus, by using the lower bounds in (50) and (51) in (48) and (49), respectively, we obtain

$$(44) + (45) \quad \geq \quad \sum_{f} (\lambda^{(f)} + \varepsilon) Q_{s(f)}^{(f)} + \sum_{f} \sum_{l} \sum_{g\neq f} \lambda^{(f\to g,l)} \left( Q_{D^{(f\to g,l)}}^{(f)} - Q_l^{(f)} \right). \tag{52}$$

We now move to the third summation (46). We have, using equation (32),

$$-\sum_{(m,n)\in\mathcal{E}} p^{\min}_{(m,n)}(f,g,l)W_{(m,n)} \quad = \quad \lambda^{(f\to g,l)}W_{(l,k^{(f\to g,l)})} \; - \sum_{(m,n)\in\mathcal{E},m\neq l} p^{\min}_{(m,n)}(f,g,l)W_{(m,n)}. \tag{53}$$

Now, using the definition of $W_{(m,n)}$ we can write

$$\begin{aligned}
&\lambda^{(f\to g,l)}W_{(l,k^{(f\to g,l)})}\\
&\quad \geq \lambda^{(f\to g,l)} \max_{(f,g),(b,b'),(c,c')} \left\{ Q_l^{(f)} + Q_l^{(g)} - Q_b^{rem(c)} - Q_{b'}^{rem(c')} - Q_{k^{(f\to g,l)}}^{(c,(f,g),(c,c'))} - Q_{k^{(g\to f,l)}}^{(c',(f,g),(c,c'))} \right\}\\
&\quad \overset{(a)}{\geq} \lambda^{(f\to g,l)} \Big( Q_l^{(f)} + Q_l^{(g)} - Q_{b^{(f\to g,l)}}^{rem(D^{(f\to g,l)})} - Q_{b^{(g\to f,l)}}^{rem(D^{(g\to f,l)})}\\
&\qquad\qquad - Q_{k^{(f\to g,l)}}^{(D^{(f\to g,l)},(f,g),(D^{(f\to g,l)},D^{(g\to f,l)}))} - Q_{k^{(g\to f,l)}}^{(D^{(g\to f,l)},(f,g),(D^{(f\to g,l)},D^{(g\to f,l)}))} \Big),
\end{aligned} \tag{54}$$

where the inequality $(a)$ is obtained by setting $(c,c')$ in the previous maximization to $(D^{(f\to g,l)}, D^{(g\to f,l)})$.

Similarly, noting that $\mathbf{p}\leq 0$ and using the definition of $W_{(m,n)}$, we can write

$$\begin{aligned}
&-p^{\min}_{(m,n)}(f,g,l)W_{(m,n)}\\
&\quad \geq -p^{\min}_{(m,n)}(f,g,l) \max_{(f,g),(c,c')} \Big( Q_m^{(c,(f,g),(c,c'))} - (Q_n^{(c,(f,g),(c,c'))} + Q_n^{(g)}\mathcal{I}_{n=c})\\
&\qquad\qquad + Q_m^{(c',(f,g),(c,c'))} - (Q_n^{(c',(f,g),(c,c'))} + Q_n^{(f)}\mathcal{I}_{n=c'}) \Big)\\
&\quad \geq -p^{\min}_{(m,n)}(f,g,l) \Big( Q_m^{(D^{(f\to g,l)},(f,g),(D^{(f\to g,l)},D^{(g\to f,l)}))} - Q_n^{(D^{(f\to g,l)},(f,g),(D^{(f\to g,l)},D^{(g\to f,l)}))} - Q_n^{(f)}\mathcal{I}_{\{n=D^{(f\to g,l)}\}}\\
&\qquad\qquad + Q_m^{(D^{(g\to f,l)},(f,g),D^{(f\to g,l)},D^{(g\to f,l)}))} - Q_n^{(D^{(g\to f,l)},(f,g),(D^{(f\to g,l)},D^{(g\to f,l)}))} - Q_n^{(g)}\mathcal{I}_{\{n=D^{(g\to f,l)}\}} \Big),
\end{aligned} \tag{55}$$

where the last inequality is obtained by setting $(c,c') = (D^{(f\to g,l)}, D^{(g\to f,l)})$ in the previous maximization. Now, $p^{\min}_{(m,n)}(f,g,l)W_{(m,n)}$ is a flow of rate $-\lambda^{(f\to g,l)}$ from $l$ on to the path $\mathcal{P} = \{$path from $k^{(f\to g,l)}$ to $D^{(f\to g,l)}\} \cup \{$path from $k^{(g\to f,l)}$ to $D^{(g\to f,l)}\}$.Therefore, we can write from (55)

$$\begin{aligned}
&-\textstyle\sum_{(m,n)\in\mathcal{E},m\neq l} p^{\min}_{(m,n)}(f,g,l)W_{(m,n)}\\
&\quad \geq \sum_{(m,n)\in\mathcal{P}} \lambda^{(f\to g,l)}W_{(m,n)}\\
&\quad \geq \lambda^{(f\to g,l)} \Big( Q_{k^{(f\to g,l)}}^{(D^{(f\to g,l)},(f,g),(D^{(f\to g,l)},D^{(g\to f,l)}))} - Q_{D^{(f\to g,l)}}^{(f)} + Q_{k^{(g\to f,l)}}^{(D^{(g\to f,l)},(f,g),(D^{(f\to g,l)},D^{(g\to f,l)}))} - Q_{D^{(g\to f,l)}}^{(g)} \Big),
\end{aligned} \tag{56}$$

Hence, by combining (53), (54), and (56), we obtain

$$\begin{aligned}
&-\sum_{(m,n)\in\mathcal{E}}\sum_f\sum_l\sum_{g>f} p^{\min}_{(m,n)}(f,g,l)W_{(m,n)}\\
&\qquad \geq \sum_f\sum_l\sum_{g>f} \lambda^{(f\to g,l)} \Big( Q_l^{(f)} - Q_{D^{(f\to g,l)}}^{(f)}\\
&\qquad\qquad + Q_l^{(g)} - Q_{D^{(g\to f,l)}}^{(g)} - Q_{b^{(f\to g,l)}}^{rem(D^{(f\to g,l)})} - Q_{b^{(g\to f,l)}}^{rem(D^{(g\to f,l)})} \Big)\\
&\qquad = \sum_f\sum_l\sum_{g\neq f} \lambda^{(f\to g,l)} \Big( Q_l^{(f)} - Q_{D^{(f\to g,l)}}^{(f)} - Q_{b^{(f\to g,l)}}^{rem(D^{(f\to g,l)})} \Big).
\end{aligned} \tag{57}$$

Finally, we move to the fourth summation term (47). We have

$$\begin{aligned}
\sum_{(m,n)\in\mathcal{E}} r^{(f\to g,l)}_{(m,n)} W_{(m,n)} &\geq \sum_{(m,n)\in\mathcal{E}} r^{(f\to g,l)}_{(m,n)} \Big( Q_m^{rem(D^{(f\to g,l)})} - Q_n^{rem(D^{(f\to g,l)})} \Big)\\
&= \lambda^{(f\to g,l)} Q_{b^{(f\to g,l)}}^{rem(D^{(f\to g,l)})},
\end{aligned}$$

where we have used the assumption that $r^{(f\rightarrow g,l)}$ is a simple path flow from $b^{(f\rightarrow g,l)}$ to $D^{(f\rightarrow g,l)}$ of magnitude $\lambda^{(f\rightarrow g,l)}$. Thus,

$$\sum_{(m,n)\in\mathcal{E}} \sum_{f} \sum_{l} \sum_{g\neq f} r^{(f\rightarrow g,l)}_{(m,n)} W_{(m,n)} \geq \sum_{f} \sum_{l} \sum_{g\neq f} \lambda^{(f\rightarrow g,l)} Q^{rem(D^{(f\rightarrow g,l)})}_{b^{(f\rightarrow g,l)}}. \tag{58}$$

Bounding the summations (44)-(47), by (52), (57), and (58), and then canceling common terms yields

$$\sum_{(m,n)\in\mathcal{E}} \gamma_{(m,n)} W_{(m,n)} \geq \sum_{f\in\mathcal{F}} (\lambda^{(f)} + \varepsilon) Q^{(f)}_{s(f)},$$

which, when substituted into equation (43), proves that $\Delta V(\mathbf{Q}[t]) \leq B - \varepsilon \sum_{f\in\mathcal{F}} Q^{(f)}_{s(f)}[t]$.

We take the expectation of both sides with respect to distribution of $\mathbf{Q}[t]$ for all $t$ between $0$ to $T-1$. Then, add both sides of the resulting $T$ inequalities and cancel common terms to get

$$\mathbb{E}[V(\mathbf{Q}[T-1]) - V(\mathbf{Q}[0])] \quad \leq \quad BT - \epsilon \sum_{t=0}^{T-1} \sum_{f\in\mathcal{F}} \mathbb{E}[Q^{(f)}_{s(f)}[t]].$$

After re-arranging terms, noting the non-negativity of $V(\mathbf{Q})$ for any feasible queue-length vector, dividing both sides by $T$, and letting $T$ go to infinity, we obtain

$$\limsup_{T\rightarrow\infty} \frac{1}{T} \sum_{t=0}^{T-1} \sum_{f\in\mathcal{F}} \mathbb{E}[Q^{(f)}_{s(f)}[t]] \quad \leq \quad \frac{B}{\epsilon} < \infty,$$

which establishes the strong stability of all entry-level queues. Then, we can recursively argue that each downstream queue must also be stable since, under our proposed RSC policy, packets are forwarded in the direction of smaller queue-lengths. Thus, the entry-level queues are statistically largest, and their stability implies the stability of the remaining queues. ■

## References


[1] T. Ho and D. S. Lun, *Network Coding: An Introduction*. Cambridge: Cambridge University Press, 2008.

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

[3] R. W. Yeung, "Multilevel diversity coding with distortion," *IEEE Trans. Inform. Theory*, vol. 41, no. 2, pp. 412–422, Mar. 1995.

[4] R. Koetter and M. Médard, "An algebraic approach to network coding," *IEEE/ACM Trans. Networking*, vol. 11, no. 5, pp. 782–795, Oct. 2003.

[5] S. Deb, M. Effros, T. Ho, D. R. Karger, R. Koetter, D. S. Lun, M. Médard, and N. Ratnakar, "Network coding for wireless applications: A brief tutorial," in *Proc. International Workshop on Wireless Ad-hoc Networks (IWWAN) 2005*, May 2005, invited paper.

[6] A. F. Dana, R. Gowaikar, R. Palanki, B. Hassibi, and M. Effros, "Capacity of wireless erasure networks," *IEEE Trans. Inform. Theory*, vol. 52, no. 3, pp. 789–804, Mar. 2006.

[7] D. S. Lun, M. Médard, R. Koetter, and M. Effros, "On coding for reliable communication over packet networks," *Physical Communication*, vol. 1, no. 1, pp. 3–20, Mar. 2008.

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

[9] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, "XORs in the air: Practical wireless network coding," *IEEE/ACM Trans. Networking*, vol. 16, no. 3, pp. 497–510, June 2008.

[10] Y. Wu, P. A. Chou, and S.-Y. Kung, "Information exchange in wireless networks with network coding and physical-layer broadcast," in *Proc. Conference on Information Sciences and Systems (CISS)*, 2005.

[11] 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. IEEE International Symposium on Information Theory (ISIT)*, July 2006, pp. 1758–1762.

[12] A. Khreishah, C. Wang, and N. Shroff, "Rate control with pairwise inter-session network coding," *IEEE/ACM Trans. Networking*, 2010, to Appear.

[13] D. S. Lun, M. Médard, T. Ho, and R. Koetter, "Network coding with a cost criterion," in *Proc. International Symposium on Information Theory and its Applications (ISITA)*, Oct. 2004, pp. 1232–1237.

[14] N. Ratnakar, D. Traskov, and R. Koetter, "Approaches to network coding for multiple unicasts," in *Proc. International Zurich Seminar on Communications (IZS)*, 2006, pp. 70–73, invited paper.

[15] T. Ho, Y.-H. Chang, and K. Han, "On constructive network coding for multiple unicasts," in *Proc. 44th Annual Allerton Conference on Communication, Control, and Computing*, Sept. 2006.

[16] L. Tassiulas and A. F. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE Trans. Automat. Contr.*, vol. 37, no. 12, pp. 1936–1948, Dec. 1992.

[17] M. Andrews, K. Kumaran, K.Ramanan, A. Stolyar, R. Vijayakumar, and P. Whiting, "Providing quality of service over a shared wireless link," February 2001.

[18] S. Shakkottai and A. Stolyar, "Scheduling for multiple flows sharing a time-varying channel: The exponential rule," *Translations of the AMS*, A volume in memory of F. Karpelevich, vol. 207, pp. 185–202, 2002.

[19] A. Eryilmaz, R. Srikant, and J. R. Perkins, "Stable scheduling policies for fading wireless channels," *IEEE/ACM Trans. Networking*, vol. 13, no. 2, pp. 411–424, Apr. 2005.

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

[21] X. Lin and N. Shroff, "The impact of imperfect scheduling on cross-layer rate control in multihop wireless networks," in *Proceedings of IEEE Infocom*, Miami, FL, March 2005.

[22] T. Ho and H. Viswanathan, "Dynamic algorithms for multicast with intra-session network coding," in *Proc. 43rd Annual Allerton Conference on Communication, Control, and Computing*, Sept. 2005.

[23] H. Seferoglu, A. Markopoulou, and U. Kozat, "Network coding-aware rate control and scheduling in wireless networks," in *Proc. International Conference on Multimedia and Expo (ICME)*, 2009.

[24] H. Seferoglu and A. Markopoulou, "Improving the performance of tcp over coded wireless networks," *CoRR*, vol. abs/1002.4885, 2010.

[25] ——, "Network coding-aware queue management for unicast flows over coded wireless networks," in *Proc. IEEE International Symposium on Network Coding (NetCod)*, Toronto,Canada, 2010.

[26] T. Cui, L. Chen, and T. Ho, "Energy efficient opportunistic network coding for wireless networks," in *INFOCOM*, 2008, pp. 361–365.

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

[28] M. Neely, E. Modiano, and C. Rohrs, "Dynamic power allocation and routing for time varying wireless networks," in *Proceedings of IEEE Infocom*, April 2003, pp. 745–755.

[29] A. Eryilmaz and R. Srikant, "Resource allocation of multi-hop wireless networks," in *Proc. International Zurich Seminar on Communications (IZS)*, Feb. 2006.

[30] A. Stolyar, "Maximizing queueing network utility subject to stability: Greedy primal-dual algorithm," *Queueing Systems*, vol. 50, no. 4, pp. 401–457, 2005.

[31] L. Chen, S. H. Low, M. Chiang, and J. C. Doyle, "Jointly optimal congestion control, routing, and scheduling for wireless ad hoc networks," in *Proceedings of IEEE Infocom*, Barcelona, Spain, April 2006.

[32] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, "Network coding made practical," MIT CSAIL, Technical Report 2006-009, Feb. 2006.

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

[34] L. Georgiadis, M. Neely, and L. Tassiulas, "Resource allocation and cross-layer control in wireless networks," *Foundations in Networking*, vol. 1, no. 1, 2006.

[35] H. Xiong, R. Li, A. Eryilmaz, and E. Ekici, "Delay performance improvement of throughput-optimal resource allocation algorithms in multi-hop networks," in *Information Theory and Applications (ITA) Workshop*, 2009.

[36] L. Ying, S. Shakkottai, and A. Reddy, "On combining shortest-path and back-pressure routing over multihop wireless networks," in *INFOCOM*, 2009.

[37] P. Chaporkar, K. Kar, and S. Sarkar, "Throughput guarantees through maximal scheduling in wireless networks," in *Proceedings of the Allerton Conference on Control, Communications and Computing*, 2005.

[38] X. Wu and R. Srikant, "Regulated maximal matching: A distributed scheduling algorithm for multi-hop wireless networks with node-exclusive spectrum sharing," in *Proceedings of* IEEE Conference on Decision and Control., 2005.

[39] L. Bui, A. Eryilmaz, R. Srikant, and X. Wu, "Joint asynchronous congestion control and distributed scheduling for wireless networks," in *Proc. IEEE Infocom*, 2006.

[40] E. Modiano, D. Shah, and G. Zussman, "Maximizing throughput in wireless networks via gossiping," in *ACM SIGMETRICS/IFIP Performance*, 2006.

[41] A. Eryilmaz, E. Modiano, and A. Ozdaglar, "Randomized algorithms for throughput-optimality and fairness in wireless networks," in *Proceedings of IEEE Conference on Decision and Control*, San Diego, CA, December 2006.

[42] S. Sanghavi, L. Bui, and R. Srikant, "Distributed link scheduling with constant overhead," 2007, technical Report.

[43] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE Transactions on Automatic Control*, vol. 36, pp. 1936–1948, December 1992.

[44] R. Dougherty, C. Freiling, and K. Zeger, "The Vámos network," in *Proc. 4th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt)*, 2006.

[45] 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. Inform. Theory*, vol. 52, no. 6, pp. 2608–2623, June 2006.

[46] D. P. Bertsekas, *Network Optimization: Continuous and Discrete Models*. Belmont, MA: Athena Scientific, 1998.