

Joint Scheduling and Instantaneously Decodable Network Coding

Danail Traskov*, Muriel Médard[‡], Parastoo Sadeghi[§], and Ralf Koetter

* Institute for Communications Engineering, Technical University Munich, GERMANY

[‡] Research Laboratory of Electronics, Massachusetts Institute of Technology, USA

[§] Research School of Information Sciences and Engineering, The Australian National University, AUSTRALIA

Emails: danail.traskov@tum.de, medard@mit.edu, parastoo.sadeghi@anu.edu.au

Abstract—We consider a wireless multi-hop network and design an algorithm for jointly optimal scheduling of packet transmissions and network coding. We consider network coding across different users, however with the restriction that packets have to be decoded after one hop. We compute the stability region of this scheme and propose an online algorithm that stabilizes every arrival rate vector within the stability region. The online algorithm requires computation of stable sets in an appropriately defined conflict graph. We show by means of simulations that this inherently hard problem is tractable for some instances and that network coding extends the stability region over routing and leads, on average, to a smaller backlog.

I. INTRODUCTION

Applying network coding to wireless, mobile ad-hoc networks has been shown to dramatically improve their performance, e.g. to reduce energy consumption [1], improve bandwidth efficiency [2] or increase throughput by enhancing the Medium-Access (MAC) layer [3]. However, the vast majority of applications assume so called *intra*-session network coding, i.e. only packets belonging to the same user are allowed to mix. The reason is that allowing data of different users to mix, leading to *inter*-session network coding is a difficult problem [4], and in fact may even require complicated non-linear processing [5].

On the other hand, approaches to inter-session network coding that are not necessarily optimal, yet practical from an engineering point of view have demonstrated large performance gains; e.g. in [3], where the authors exploit the beneficial effect of network coding to reduce MAC-layer congestion. In a wireless network, due to broadcasting, nodes overhear packets (that are not intended for them) frequently. This additional “evidence” can be used to combine several packets in one transmission.

This is also the approach that we take, more precisely assume that a node decides to combine L packets with binary XOR and broadcasts them to its neighbors. Then, we require the transmission to be *instantaneously decodable* for all neighbors, which is achieved when all neighbors have

overheard already $L - 1$ of the packets. Every receiver can then cancel out all, but the one packet that is new to him. As we show, the instantaneous decodability condition can be formulated as a *conflict graph* model where valid packet combinations correspond to *stable sets*¹. This is, in general, an NP-hard combinatorial problem which is inherent in the instantaneous decodability condition. Our simulations indicate that for moderate size networks the optimal solution can be within reach.

In mobile wireless networks a large body of work suggests [2], [7] that to capitalize on significant gains, routing and network coding have to be optimized jointly with the MAC-layer. This is also the approach we take, by formulating a linear program that includes both scheduling and network coding. With this problem formulation we are able to compute the achievable rate region of our technique and to quantify the gains over routing. On the other hand, in most practical mobile networks a computationally lightweight, decentralized and online algorithm is preferable. We formulate such an algorithm based on ideas from [6], where authors derive a widely applicable class of online scheduling algorithms achieving optimal throughput. To include network coding, we introduce a system of *virtual queues* that can be served jointly subject to the constraints arising from the conflict graph model.

There are two lines of work that are related to our approach. In [3] the authors introduce COPE, a 802.11-based protocol that uses network coding to enhance the performance of the MAC-layer. There, the idea of combining packets locally, opportunistically and heuristically was developed and shown to yield significant performance gains. However, the decision which packets to combine is done by means of a sequential (essentially greedy) search heuristic, while we optimize over the set of network coding decisions and over the schedule. In [8], the authors analyze theoretically the performance of COPE-type network coding by means of formulating a linear program capturing the network coding, routing and scheduling constraints. Compared to their work, our approach optimizes over a larger set of network coding decisions and furthermore we present an online algorithm that stabilizes every point within the rate region.

The other line of work starts with [9] (see also [10]), where the authors consider a fixed network and relax the instantaneous decodability assumption to allow the mixing

¹Some authors, e.g. [6] prefer the terminology independent set.

This material is based upon work under subcontract #18870740-37362-C issued by Stanford University and supported by the Defense Advanced Research Projects Agency (DARPA). It was also supported by the United States Department of the Navy's Space and Naval Warfare Systems Command (SPAWAR) under Contract No. N66001-06-C-2020 through BAE Systems (DARPA CBMANET) and by the European Commission in the framework of the FP7 Network of Excellence in Wireless COMMunications NEWCOM++ (Contract no. 216715).

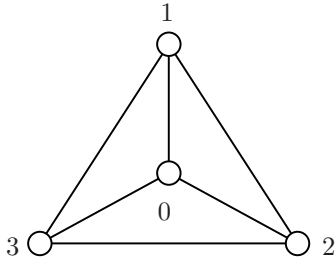


Fig. 1. The topology graph \mathcal{G}_t of a network with $n = 3$ nodes and a relay node 0. Each link stands for two directed links, going in opposite directions.

of packets only subject to being decodable eventually. This comes, however, at the price of allowing at most two packets to mix. The achievable region of this technique was later shown in [11] to be stabilizable with an online backpressure algorithm. These techniques, while optimizing over the entire network instead of just locally, apply only a restricted set of network coding operations, a strict subset of the operations we allow in our present approach. In summary, we

- find the optimal network coding solution within the class of instantaneously decodable codes,
- compute the rate region for jointly optimal network coding and scheduling,
- derive an online algorithm that stabilizes every point within the rate region,
- show the utility of our approach by means of simulations.

The rest of our work is organized as follows. In Section II, we introduce the network model and discuss the network coding framework. In Section III, we compute the stability region of the network and derive the online algorithm in the following Section IV. In Section V, we present simulation results. Section VI concludes the paper and gives an outlook on possible extensions and further work.

II. NETWORK MODEL

A. General model and assumptions

Consider a wireless network, the *topology* of which is represented as a directed graph $\mathcal{G}_t = (\mathcal{N}_t, \mathcal{A}_t)$ with node set $\mathcal{N}_t = \{0, 1, \dots, n\}$ and arc set $\mathcal{A}_t = \{(i, j) : 0 \leq i, j \leq n, i \neq j\}$. The case where $n = 3$ is depicted in Fig. 1. From the definition, the network is fully symmetric, however, we assume that node 0 is a special *relay* node with extended capabilities. This model can arise, for example, when the network consists of a number of ground nodes $1, \dots, n$ and one unmanned aerial vehicle (UAV), node 0, with extended range, power and a larger set of coding and modulation schemes. The network operates with constant-length packets and in slotted time, where the slot index t is an integer corresponding to the time interval $[t, t + 1)$.

We assume, for simplicity, that the relay serves solely the purpose of enhancing communication between the other nodes and does not inject individual packets. Exogenous packet arrivals at node i with destination j (resulting from processes at the application layer of node i) occur according to *admissible* stochastic processes $\mathbf{A}(t) = (A_i^j(t))$, for $1 \leq i, j \leq n, i \neq j$,

with average rates $\lambda_i^j = E[A_i^j]$. We use the same definition of admissible as the authors in [12, Definition 3.4]

Definition 1 A process $A(t)$ is admissible with rate λ if

- The time average expected arrival rate satisfies:

$$\lim_{t \rightarrow \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} E\{A(\tau)\} = \lambda. \quad (1)$$

- For all time slots t , we have $E\{A(t)^2 | \mathbf{H}(t)\} \leq A_{\max}^2$, where A_{\max} is a positive constant and $\mathbf{H}(t)$ represents the history up to time t , i.e. all events in slots $\tau \in \{0, \dots, t-1\}$.
- For any $\delta > 0$, there exists an interval size T such that for any initial time t_0 the following condition holds:

$$E \left\{ \frac{1}{T} \sum_{k=0}^{T-1} A(t_0 + k) | \mathbf{H}(t_0) \right\} \leq \lambda + \delta. \quad (2)$$

Assume that transmissions from and to the relay are always successful. On the other hand, any other link can be either ON, in which case it can support the transmission of one packet per slot or OFF, in which case no packet can be transmitted over this link. The topology state at time t is thus given by a binary vector $S(t) = (S_{ij}(t))$, for $i, j \in \{1, \dots, n\}, i \neq j$, with $S_{ij}(t) = 1$ indicating that the corresponding link is ON. Assume that the state $S(t)$ evolves according to a finite state, irreducible Markov chain with state space \mathcal{S} and let π_s denote the average fraction of time that the process spends in state $S(t) = s$. For such chains the time averages π_s are well defined and with probability 1 we have

$$\pi_s = \lim_{t \rightarrow \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} 1_{[S(\tau)=s]}, \quad \text{for all } s \in \mathcal{S}, \quad (3)$$

where $1_{[\cdot]}$ is the indicator function. Due to interference at most one node in the network can transmit per slot.

If node i transmits a packet designated for node j , and j receives it successfully, it is removed from the system. Otherwise, the following sequence of actions is carried out

- node i removes it from its queue,
- the relay (that by assumption receives every packet successfully) assumes responsibility for the packet and stores it for further transmission,
- all nodes that have overheard the packet store it until it has reached its destination (for the purpose of using it at a later stage for network coding).

This scheme requires a certain amount of perfect feedback in the following form. After any packet transmission from a non-relay node, every other node has to acknowledge (or negatively acknowledge) the reception to the relay. Note that feedback between non-relay nodes is not required, which is consistent with our assumption that these nodes have more limited capabilities than the relay. For our analysis we will use three different graphs, each of them describing a different aspect of the system. In addition to the topology graph \mathcal{G}_t , we will introduce the queuing network (directed) graph \mathcal{G}_q and the network coding conflict (undirected) graph \mathcal{G}_c , both to be precisely defined later.

B. Queuing model

Consider the two-stage *queuing network* \mathcal{G}_q in Fig. 2 consisting of the queues (R_i^j) and the virtual queues at the relay (X_Q^j) . In contrast to the topology \mathcal{G}_t , here we explicitly model the dynamic behavior in a queuing theoretic framework that allows us to accommodate scheduling and network coding. Packets that leave the system (the arrows with solid tips) are directed to an artificial node E , the *system exit*. This queuing network, in particular its stability region and an online stabilizing algorithm, is the focus of our analysis.

In the original network of Fig. 1, a packet broadcasted from node i can, depending on the state s , either reach its destination or it is overheard by the relay and possibly a subset of its neighbors. In the queuing model, correspondingly, it is either transferred to the system exit or to one of the virtual queues at the relay. That means that for a given topology state s , each queue R_i^j will have exactly one state-dependent outgoing link denoted by $(R_i^j, d(s))$, where we define

$$d(s) = \begin{cases} E & \text{if } s_{ij} = 1, \\ X_{Q'}^j & \text{if } s_{ij} = 0, \end{cases} \quad (4)$$

where $Q' = \{k | k \neq j, s_{ik} = 1\} \cup i$.

A queue with backlog $X(t)$ evolves according to the discrete-time dynamics $X(t+1) = \max(X(t) - \mu(t), 0) + A(t)$, where $A(t)$ is the arrival process, and $\mu(t)$ the service process. For queue stability, we use the following definition [12, Definition 3.1]

Definition 2 A queue is called (strongly) stable if

$$\limsup_{t \rightarrow \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} E\{X(\tau)\} < \infty. \quad (5)$$

A network of queues is strongly stable if all queues comprising the network are strongly stable. In our model, each node $i \in \{1, \dots, n\}$ has queues R_i^j , $j \in \{1, \dots, n\} \setminus i$, one for each possible packet destination. The relay, on the other hand, has a system of *virtual queues* in which it stores received packets (that failed to reach their designated destination) for the purpose of performing network coding. More precisely, the relay partitions all overheard packets in $n \cdot (2^{n-1} - 1)$ equivalence classes, according to their next-hop j and the set of nodes $\mathcal{Q} \subset \{1, \dots, n\} \setminus j$, $\mathcal{Q} \neq \emptyset$ that have knowledge of them. The set \mathcal{Q} is never empty as there is always one node, the original sender i , that has the packet. The relay keeps track of a virtual queue for each such class of packets. Let $\mathbf{X}(t) = (X_Q^j(t))$ denote the queue length vector of all packet classes at time t .

C. Network coding

The relay node has network coding capabilities, in that it can combine several of its queued packets with binary XOR, subject to the constraint that the combination can be instantaneously decoded at all neighboring nodes [3]. This means that if the relay XORs L packets together, each of the intended receivers must have overheard already $L-1$ of them.

Every receiver can then cancel out all but the one packet that is new to him.

We can represent valid network coding combinations resulting from this condition by a graphical model. In this *conflict graph* approach, we construct an undirected graph with vertices corresponding to the queues. Two queues are connected with a link if they *cannot* be served jointly, i.e. packets from the two queues cannot be XORed together, because they violate the instantaneous decodability condition. This is made precise in the following definition.

Definition 3 For the system of queues (X_Q^j) , the *conflict graph* $\mathcal{G}_c = (\mathcal{V}, \mathcal{E})$ is an undirected graph with a one-to-one correspondence between vertices \mathcal{V} and queues. Two vertices $X_{Q_1}^i$ and $X_{Q_2}^j$ are **not** connected if

- $i \neq j$,
- $i \in Q_2$, and $j \in Q_1$,

otherwise they are connected with an undirected link.

The first condition guarantees that the packets in the two queues have different destinations and the second condition means that each destination has overheard the packet meant for the other destination node. We define a valid configuration of queues as a set of nodes in the conflict graph without any conflicting pair, i.e. a valid configuration is a stable set.

Definition 4 A *stable set* C of an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a set of nodes such that for any pair of nodes in C , there is no edge connecting them. Its incidence vector is a column vector of length $|\mathcal{V}|$, defined as

$$\chi_v^C = \begin{cases} 1 & \text{if } v \in C, \\ 0 & \text{otherwise.} \end{cases} \quad (6)$$

The collection of all stable sets is denoted by $STAB(\mathcal{G})$.

A *maximal* stable set is one that is not contained in any other stable set. A *maximum* stable set is a stable set of largest cardinality; finding such a stable set is in general NP-hard. The stable set polytope $P_{STAB}(\mathcal{G}_c)$ is the convex hull of the incidence vectors of all stable sets of \mathcal{G}_c .

For the network in Fig. 1, the corresponding conflict graph is depicted in Fig. 3.

D. Joint scheduling and network coding

We return to the queuing model (see Fig. 2) and give a precise definition

Definition 5 The *queuing network* $\mathcal{G}_q = (\mathcal{N}_q, \mathcal{A}_q)$ is a directed graph, with node set

$$\mathcal{N}_q = \left\{ (R_i^j) \cup (X_Q^j) \cup E \right\} \quad (7)$$

and arc set

$$\mathcal{A}_q = \left\{ (R_i^j, E) \right\} \quad \forall R_i^j \in \mathcal{N}_q \quad (8)$$

$$\cup \left\{ (R_i^j, X_Q^j) \right\} \quad \text{if } i \in \mathcal{Q} \quad (9)$$

$$\cup \left\{ (X_Q^j, E) \right\} \quad \forall X_Q^j \in \mathcal{N}_q. \quad (10)$$

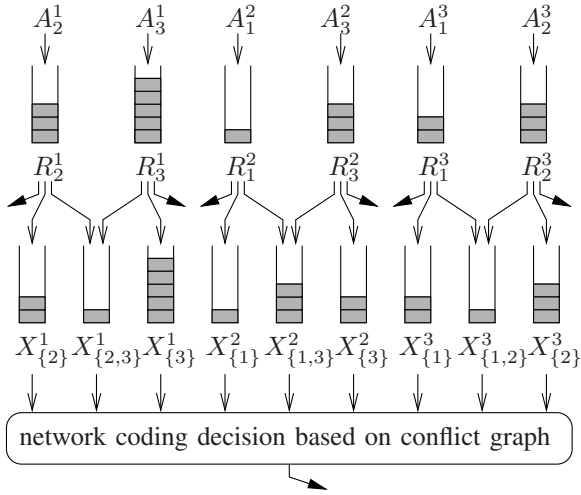


Fig. 2. The corresponding queuing network graph \mathcal{G}_q for the network in Fig. 1. Directed links indicate possible packet transitions; packets that leave the system (the arrows with solid tips) are directed to an artificial node E , the *system exit*. A subset of the virtual queues X_Q^j can be served jointly in one time slot if they correspond to a stable set in the conflict graph \mathcal{G}_c .

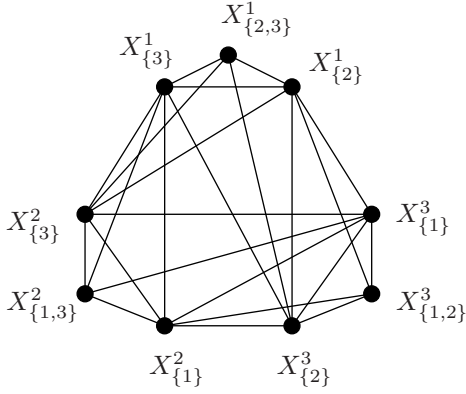


Fig. 3. The conflict graph \mathcal{G}_c corresponding to the virtual queues at the relay in Fig. 2.

Due to the interference constraints the control action in each time slot is to either serve one of the links $(R_i^j, d(s))$ or a valid configuration of the (X_Q^j, E) links subject to the network coding constraints. A *control input* $\mathbf{I}(t) = (I_{ab}(t))$ for the queuing network is a binary vector with $I_{ab}(t) = 1$ if link $(a, b) \in \mathcal{A}_q$ is activated in slot t .

The control space \mathcal{I}_s for a state s thus consists of

$$\begin{aligned} \mathcal{I}_s &= \mathcal{I}'_s \cup \mathcal{I} \\ &= \left\{ \left(R_i^j, d(s) \right) : i, j \in \{1, \dots, n\}, i \neq j \right\} \\ &\quad \cup \left\{ \left(X_{Q_i}^{j_i}, E \right) : \left(X_{Q_i}^{j_i} \right) \text{ is a stable set in } \mathcal{G}_c \right\}, \end{aligned} \quad (11)$$

where \mathcal{I}'_s denotes the state-dependent part, and \mathcal{I} the state-independent part of the control.

Let $\mathbf{c}(I(t), S(t)) = (c_{ab}(I(t), S(t)))$ denote the link capacity vector under control $I(t) \in \mathcal{I}_{S(t)}$ and state $S(t) \in \mathcal{S}$. Based on the previous discussion, the capacity of link (a, b) , measured in packets/slot is

$$c_{ab}(I(t), S(t)) = \begin{cases} 1 & \text{if } I_{ab}(t) = 1, \\ 0 & \text{otherwise.} \end{cases} \quad (12)$$

Consider the region defined by

$$\Gamma = \sum_{s \in \mathcal{S}} \pi_s \text{CH} \{ \mathbf{c}(I, s) : I \in \mathcal{I}_s \}, \quad (13)$$

where $\text{CH}(\cdot)$ denotes the convex hull and the different convex hulls are added using the usual set summation. Using the decomposition from Eqn. (11), we can rewrite the region Γ as follows, isolating the contribution of the stable set polytope of the conflict graph

$$\begin{aligned} \Gamma &= \sum_{s \in \mathcal{S}} \pi_s \mu_s \text{CH} \{ \mathbf{c}(I, s) : I \in \mathcal{I}'_s \} \\ &\quad + \left[\sum_{s \in \mathcal{S}} \pi_s (1 - \mu_s) \right] P_{STAB}(\mathcal{G}_c), \end{aligned} \quad (14)$$

where $\mu_s \in [0, 1], \forall s \in \mathcal{S}$. The significance of this region is that every vector (g_{ab}) of long-term link transmission rates that can be supported by the network has to lie in Γ [12]. For the introduced constrained queuing system, two questions naturally arise and we will address them next: the optimal service policy and its associated stability region.

III. STABILITY REGION

We begin by studying the stability region (or network layer capacity region, as opposed to the information theoretic notion of capacity) which is defined as follows [12]

Definition 6 *The stability region Λ is the closure of the set of all arrival rate matrices (λ_i^j) that can be stably supported by the network considering all possible policies for routing, scheduling and restricted network coding (i.e. instantaneous decodability and network coding only at the relay).*

The characterization of the stability region is given in the following theorem.

Theorem 1 *The stability region for the constrained queuing system in Fig. 2 is the set of all arrival rate vectors (λ_i^j) such that for all links $(a, b) \in \mathcal{A}_q$ there exists a non-negative flow vector $(f(a, b))$ and a transmission rate vector $(g(a, b)) \in \text{Cl}(\Gamma)$ satisfying² the flow conservation constraints*

$$\lambda_i^j \leq f(R_i^j, E) + \sum_Q f(R_i^j, X_Q^j), \quad \forall \lambda_i^j, \quad (15)$$

$$\sum_i f(R_i^j, X_Q^j) \leq f(X_Q^j, E), \quad \forall X_Q^j, \quad (16)$$

and the capacity constraints

$$f(a, b) \leq g(a, b), \quad \forall (a, b) \in \mathcal{A}_q. \quad (17)$$

Proof: This is a straightforward application of [12, Theorem 3.8] to the queuing network \mathcal{G}_q . ■

² $\text{Cl}(\cdot)$ denotes the closure of a set.

IV. ONLINE ALGORITHM

The stability region tells us that if the average arrival rates were fixed and known a priori, there exists a policy that stabilizes the network. However, it might not be causal, i.e. the decision at time t might depend on events occurring after time t . An *online* algorithm on the other hand, decides at time t solely based on the history up to this time and the current state of the network. As the authors in [6], [12] have shown, there exists a class of online algorithms, so called *differential backpressure* algorithms that stabilize every point in the interior of the stability region.

Consider the following three-step algorithm.

1. Computation of backpressure weights: In each time slot t , first observe the topology state variable $\bar{S}(t)$. Then compute for all links $(R_i^j, d(s))$ the differential backlogs $w_i^j(t)$ as follows

$$w_i^j(t) = \begin{cases} R_i^j(t) & \text{if } d(s) = E, \\ R_i^j(t) - X_{Q'}^j(t) & \text{if } d(s) = X_{Q'}^j. \end{cases}$$

Compute the maximum weighted stable set of \mathcal{G}_c with weights $X_{Q'}^j(t)$

$$\mathbf{c}^* = \arg \max_{\mathbf{c} \in \text{STAB}(\mathcal{G}_c)} \{ \mathbf{X}^T(t) \mathbf{c} \},$$

and denote the corresponding weight $w^*(t) = \mathbf{X}^T(t) \mathbf{c}^*$.

2. Scheduling: Select the maximum weight among $\{w^*(t), w_i^j(t)\}$, for $i, j = 1, \dots, n$. The queue scheduled for service is the relay if the maximum is $w^*(t)$, or otherwise the queue R_i^j corresponding to the maximum backpressure weight $w_i^j(t)$.

3. Network coding: If the relay is scheduled for transmission, identify the queues which are members of the stable set \mathbf{c}^* computed in the previous step, and serve them jointly. To that end, take the packets at the head of each queue, combine them with binary XOR and transmit the resulting combination.

The described algorithm stabilizes every arrival rate vector within the stability region. The following result, originally due to [6], is cited from [12][Theorem 4.5].

Theorem 2 *The backpressure algorithm stabilizes the network for an arrival rate vector λ if there exist a scalar $\epsilon > 0$ such that $\lambda + \epsilon \mathbf{1} \in \Lambda$, where $\mathbf{1}$ denotes the vector with all entries equal to 1.*

A remarkable consequence is that the algorithm stabilizes the system for all points in the interior of the stability region without even requiring knowledge of the stability region.

V. PERFORMANCE EVALUATION

We illustrate the performance of our scheme in three ways. Firstly, we illustrate the network coding gains by computing the volume of the stable set polytope $P_{STAB}(\mathcal{G}_c)$ and comparing with the volume of the constraint polytope when no network coding is allowed. This approach has been pursued in [13] in the context of network coding for switches with multicast capabilities. Secondly, we compute the stability region for network coding and for routing, and thirdly, we simulate the online scheduling and network coding algorithm.

A. Polytope volume computation

Consider the case $n = 3$ nodes and the 9 virtual queues which can be scheduled for joint service according to the conflict graph in Fig. 3. By inspection, the conflict graph contains one maximum stable set of cardinality 3, namely $\{X_{1,3}^2, X_{2,3}^1, X_{1,2}^3\}$, similarly nine maximal stable sets of cardinality 2 and nine stable sets corresponding to the individual vertices, so it can be written as the convex hull of these 19 points and the origin. Using the Multi-Parametric Toolbox for MATLAB [14], we have used this representation to compute its volume, which turns out to be $2.8660 \cdot 10^{-4}$. Without network coding, only one virtual queue can be served at a time, so the ‘‘conflict graph’’ when only routing is allowed is the complete graph K_9 . The volume of the resulting stable set polytope (which is a 9-dimensional standard simplex) is $(9!)^{-1} = 2.7557 \cdot 10^{-6}$. The ratio of the two volumes is $\text{Vol}(P_{STAB}(\mathcal{G}_c))/(9!)^{-1} = 104$.

B. Stability region

We compute the stability region as characterized in Theorem 1 for the special case when all injection rates are equal. Though this computation is not easier than the general case, it has the nice property that the network throughput is parameterized by a scalar $\lambda = \lambda_i^j$. We consider $n = 3$ and the state process is assumed to be i.i.d. across time and across links with each link being ON with probability 0.2 and OFF with probability 0.8. Routing, i.e. serving one virtual queue at a time, leads to a maximum symmetric rate λ_r and network coding to a rate λ_n which, due to the fact that network coding includes routing as a special case, is at least as large as λ_r . The maximum symmetric rates, $\lambda_r = 0.1448$ for routing and $\lambda_n = 0.1521$ for network coding, are shown in Fig. 4.

C. Online algorithm

To illustrate the performance of the online algorithm, we simulate its behavior for symmetric input rates which are close to the breaking points for routing and network coding, respectively. Consider $\lambda_1, \dots, \lambda_4$ as indicated in Fig. 4 and the corresponding sample paths in Fig. 5. For λ_1 , which is in the stability region of both policies, we see that routing leads on average to significantly more packets in the system. When we slightly increase the rate to λ_2 routing breaks down, while network coding is largely unaffected. Going further to λ_3 network coding is still stable, though at a higher average backlog. Finally, at λ_4 both systems operate beyond stability but network coding ‘‘degrades’’ more gracefully.

VI. CONCLUSION AND FURTHER WORK

We investigated the stability region as well as online stabilizing algorithms for instantaneously decodable network coding. It was shown that network coding can extend the stable operation regime of network and on average reduce the backlog in the system. Possible extensions are to allow every node, not just the relay, to perform network coding. Furthermore, one can investigate relaxations of the instantaneous decodability constraint and the associated trade-off between higher throughput and computational complexity/ coordination overhead.

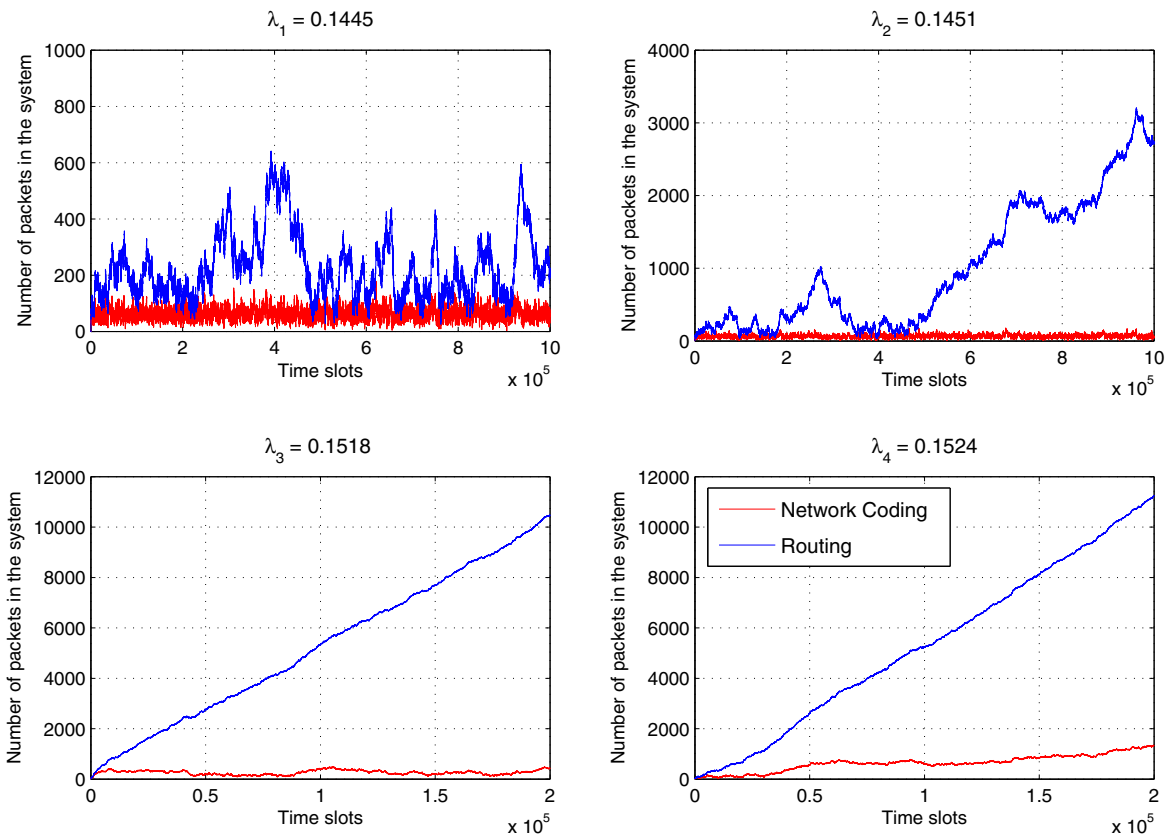


Fig. 5. The total number of queued packets in the system for different injection rates. In particular, for λ_1 both routing and network coding stabilize the system, for λ_2 and λ_3 only network coding stabilizes the system, while for λ_4 both policies result in an unstable system.

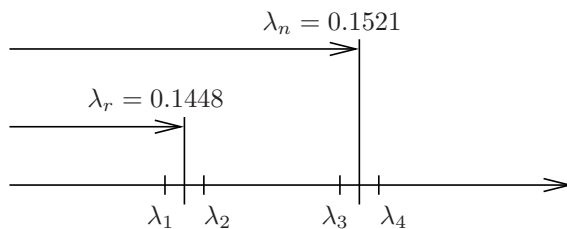


Fig. 4. Routing stabilizes the network for all rates smaller than $\lambda_r = 0.1448$, network coding extends the stability region to $\lambda_n = 0.1521$, which is an increase of about 5%. In Fig. 5 we simulate sample paths of the online backpressure algorithm for the points $\{\lambda_1, \lambda_2\} = \lambda_r \pm 0.0003$, and $\{\lambda_3, \lambda_4\} = \lambda_n \pm 0.0003$.

REFERENCES

- [1] D. Lun, N. Ratnakar, M. Médard, R. Koetter, D. Karger, T. Ho, and E. Ahmed, "Minimum-cost multicast over coded packet networks," *IEEE Trans. on Information Theory*, vol. 52(6), pp. 2608–2623, June 2006.
- [2] D. Traskov, M. Heindlmaier, M. Médard, R. Koetter, and D. Lun, "Scheduling for network coded multicast: A conflict graph formulation," in *GLOBECOM Workshops, 2008 IEEE*, New Orleans, LA, Dec. 2008.
- [3] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, "XORs in the air: Practical wireless network coding," in *In Proc. of ACM SIGCOMM*. ACM Press, 2006, pp. 243–254.
- [4] R. Koetter and M. Médard, "An algebraic approach to network coding," *IEEE/ACM Transactions on Networking*, vol. 11, pp. 782–795, 2003.
- [5] R. Dougherty, C. Freiling, and K. Zeger, "Insufficiency of linear coding in network information flow," *Information Theory, IEEE Transactions on*, vol. 51, no. 8, pp. 2745–2759, Aug. 2005.
- [6] 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, pp. 1936–1948, Dec. 1992.
- [7] Y. Sagduyu and A. Ephremides, "On joint MAC and network coding in wireless ad hoc networks," *Information Theory, IEEE Transactions on*, vol. 53, no. 10, pp. 3697–3713, Oct. 2007.
- [8] S. Sengupta, S. Rayanchu, and S. Banerjee, "An analysis of wireless network coding for unicast sessions: The case for coding-aware routing," in *IEEE INFOCOM 2007*, Anchorage, AK, 2007.
- [9] D. Traskov, N. Ratnakar, D. Lun, R. Koetter, and M. Médard, "Network coding for multiple unicasts: An approach based on linear optimization," in *ISIT 2006*, Seattle, WA, July 2006.
- [10] T. Ho, Y. Chang, and K. J. Han, "On constructive network coding for multiple unicasts," in *44th Allerton conference*, Monticello, IL, 2006.
- [11] A. Eryilmaz and D. S. Lun, "Control for inter-session network coding," in *Proc. of (NetCod)*, 2007.
- [12] L. Georgiadis, M. J. Neely, and L. Tassiulas, *Resource allocation and cross-layer control in wireless networks*. Foundations and Trends in Networking, 2006.
- [13] J. Sundararajan, M. Médard, M. Kim, A. Eryilmaz, D. Shah, and R. Koetter, "Network coding in a multicast switch," in *IEEE INFOCOM 2007*, Anchorage, AK, 2007.
- [14] M. Kvasnica, P. Grieder, and M. Baotic, "Multi-Parametric Toolbox (MPT)," [Online], Available: <http://control.ee.ethz.ch/mpt/>, 2004.