Optimal scheduling of pairwise XORs under statistical overhearing and feedback

Georgios S. Paschos*, Leonidas Georgiadis† and Leandros Tassiulas*
* University of Thessaly, † Aristotle University of Thessaloniki

***Abstract*—We study the problem of scheduling transmissions of packets belonging to several flows traversing a given node while making local encoding decisions. Practical wireless network coding solutions rely on knowledge of key knowledge which is obtained at the transmitter/scheduler either by acknowledgments or statistically. In the latter case, the knowledge of overhearing events for each packet improves progressively with feedback from the transmissions. We propose a virtual network mechanism in order to characterize the throughput region of such a system for the case where we allow only pairwise XORing. We also provide the policy which achieves the throughput region.**

I. Introduction

The strength of local XOR operations lies on the simplicity of the decoding functionality. Given that $N-1$ out of N native packets are known, a destination receiver can apply the XOR operation on the encoded packet to obtain the N^{th} packet. Despite the fact that the throughput gain from this method varies greatly with the topology, the aggregation of the effect throughout all network, along with the ability to implement the local encoding operations in a transparent way to the rest of the network layer stack, is what makes wireless NC important for practical implementations.

In order to increase the efficiency of the wireless NC scheme, COPE [1] proposes *opportunistic listening*, see Fig 1-a. This extra feature enables encoding of packets belonging to non-symmetric flows (two flows are called symmetric when one's source is the other's destination and vice versa) and thus extends the throughput benefits to wider topologies and flow scenarios. Collecting the required overhearing state information at the encoding node is not a trivial task though. The first approach in [1] was to deal with this issue by *explicitly* acknowledging all overheard packets, a policy reported by the authors to be sluggish and costly as the channel rate and number of neighbors increases. As an alternative lightweight approach, obtaining statistical information about the overhearing events was proposed. This comes of course at a loss of throughput, the so-called regret region where the scheduler regrets not having deterministic state information. In this paper we are interested in quantifying this loss, as well as propose an algorithm that stabilizes the network whenever the arrival conditions make it stabilizable.

Apart from [1], there exist other practical approaches that try to enable beyond-COPE practices in real systems using local NC, like CLONE [2] and ER [3]. The authors in [4] propose a simple practical scheme, XOR-SYM which allows XOR coding of symmetric sessions only and disregards opportunistic listening. Similar to [5] and [6], [4] considers dynamic strategies that do not require arrival rates as input. However, all these approaches assume that the scheduler/encoder (or the entity that makes the transmission decisions) has deterministic knowledge of what information each destination has. Our work builds on top of [1] and [4] addressing the problem of making encoding decisions based solely on statistical information about the overhearing state while being agnostic to arrival rates. Other approaches that consider multihop scheduling and NC problem and invoke dynamic backpressure algorithms can be found in [5], [6]. The on-line system is robust to dynamics because it reacts to present circumstances using the state of the queues. The difference in our work, lies on the fact that contrary to previous work, we do not assume complete knowledge for the system. Instead, we assume that the scheduler possesses only statistical information about the overhearing events, i.e. which destination holds which keys.

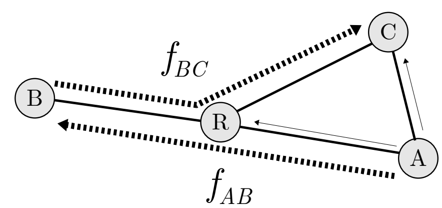

Fig. 1. Opportunistic listening; when Alice uploads her packet destined to Bob, Chloe overhears the transmission and stores the packet as a key. Later, the relay node may mix the flows f_{AB} and f_{BC} successfully. Bob decodes due to ownership and Chloe due to overhearing.

Scheduling in systems with uncertainty and feedback is a topic of recent interest, [7], [8]. In previous works regarding scheduling in systems with uncertainty and feedback, it is stressed that in the general case, such problems are well modeled by Markov decision processes which however are often intractable to solve. In our work, We make a connection between scheduling problems with feedback and the control theory of [9] which results in feasible algorithms for solving such problems optimally.

Driven by the applicative nature of this approach, we focus on wireless network coding and study the problem of scheduling packets and making encoding decisions jointly relying only on statistical overhearing information. We make a few assumptions similar to prior practical NC schemes that affect the throughput region:

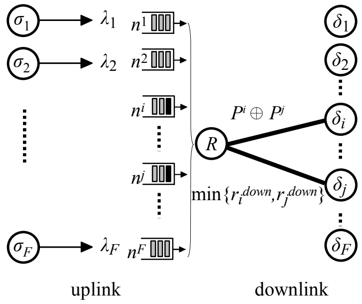

Fig. 2. Overview of our scheme; We study scheduling in the downlink but consider also the overhearing events that take place in the uplink.

1) Use XORs only for combining packets.
2) Terminals are not allowed to forward encoded packets (local NC), i.e. they must decode immediately the packet and then encode it again if appropriate at the next hop.
3) Terminals are not allowed to store encoded packets for future use.
4) The downlink does not have erasures.
5) We consider singlehop downlink only.

Assumptions 1-2 are clearly justified for practical NC implementations. We plan to relax assumptions 3-5 in future work. In this paper we make the following contribution:

- We propose a virtual network technique which categorizes the packets based on the belief of the scheduler about which receiver has which keys.
- Using this virtual network, we characterize the throughput region of the system in the case of F flows and allowing for pairwise encoding (i.e. encoding up to two packets).
- We also provide the throughput optimal policy for this case.

II. Problem Formulation

We assume the existence of 2–hop flows traversing a central node and we write $i \in \mathcal{F} = \{1, 2, \ldots, F\}$ such that $i : \sigma_i \rightarrow$ R $\rightarrow \delta_i$ where σ_i denotes the source node of flow i, δ_i denotes its destination node and R is the central node called the relay. The relay maintains a queue for each flow, call it n^i for flow i and let x^i be its corresponding backlog. See Fig. 1-b for a visual example.

In the **uplink phase**, the flow sources transmit native packets towards the relay at rates r_i^{up}. We denote the packets of flow i arriving at the relay with P_k^i, where $k = 1, 2, \ldots$ orders the packets based on arrival time (we may omit the subscript k when we speak of an arbitrary packet). Each arriving packet P_k^i is associated with an overhearing state vector $\mathbf{s}_k^i$ that characterizes the destinations having overheard this packet. $\mathbf{s}_k^i$ is a binary vector taking values in $\mathcal{S} = \{0, 1\}^F$ with $s_k^i(j) = 1$ indicating that node δ_j (the destination of flow j) has (by overhearing or ownership) the packet P_k^i in its buffer and $s_k^i(j) = 0$ indicating the complement. We assume that $\mathbf{s}_k^i$ is random and the randomness is stirred by channel fading, mobility or spatially differentiated collisions. The relay obtains statistical information about the overhearing events through a mechanism that operates independently from the scheduler and at a larger time-scale. Thus, the relay initially knows the probability of the event $s^i(j) = 1$ denoted as q_{ij}.

In the **downlink phase**, the relay may select a set of packets $\mathcal{P}$ chosen from different flows, encode them in a single packet and transmit the encoded packet at the minimum rate $\min_i\{r_i^{down}\}$ where the minimization is over the set of flows to which the packets in $\mathcal{P}$ belong. In this paper we limit ourselves to the case where $|\mathcal{P}| \leq 2$.

In the **feedback phase**, each intended destination δ_i attempts to decode the encoded packet and obtain the corresponding P_k^i, and in case of success returns an *acknowledgment message* (ACK). Once the ACK is received, the relay removes P_k^i from queue n^i which is now considered served. In case of an encoded packet, we have four distinct cases which convey complete feedback information to the relay about the decoding of the receivers as well as the overhearing events of the transmitted packets which were not decoded successfully. Say we transmit $P_{k_1}^i \oplus P_{k_2}^j$, we have the following cases:

1) Both packets are ACKed in which case they both leave the system.
2) Packet $P_{k_1}^i$ is ACKed and packet $P_{k_2}^j$ is not ACKed. In this case, $P_{k_1}^i$ leaves the system and $P_{k_2}^j$ stays in the system while the relay learns that δ_i has $P_{k_2}^j$ (i.e. it learns that $s_{k_2}^j(i) = 1$) and it may use this information in the future.
3) Packet $P_{k_1}^i$ is not ACKed and packet $P_{k_2}^j$ is ACKed. By symmetry, $P_{k_1}^i$ stays in the system and the relay learns that δ_j has $P_{k_1}^i$ and $P_{k_2}^j$ leaves the system.
4) Both packets are not ACKed in which case they both stay in the system while the relay learns that δ_j does not have $P_{k_1}^i$ and δ_i does not have $P_{k_2}^j$.

The above described feedback mechanism is the typical handshaking procedure used in IEEE 802.11 networks and thus does not incur the same problems as the one used for acknowledging overhearing events. The acknowledgment of overhearing delays the encoding decisions while the acknowledgment of decoding takes place in a lower layer and does not hinder the relay operation.

Utilizing feedback information, the relay may improve the state knowledge for a packet each time a decoding failure occurs. This implies that the expected service rate obtained by serving a given queue is not constant, a fact that gives opportunities for performance improvements but complicates the scheduling decisions. Below, we motivate further our work.

A. Encoding decision motivation

Consider the simple network of Figure 3 where two flows exist, A→R→C and B→R→D. D overhears A with probability q_{AD} and C overhears B with probability q_{BC} and let $q_{\text{AD}} = 1$, $q_{\text{BC}} = q$, hence after transmission of an encoded packet $P_{AC} \oplus P_{BD}$, destination D always decodes packet P_{BD} while destination C may or may not decode packet P_{AC}. Consider the downlink rates $r_C^{down} \doteq r_2$ and $r_D^{down} \doteq r_1$ and the uplink rates $r_A^{up} = r_B^{up} \doteq r_1$, and assume equal

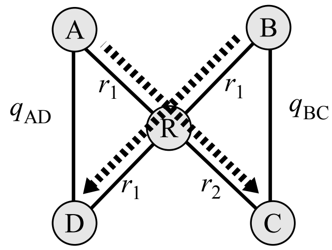

Fig. 3. An example network with two flows, A→R→C and B→R→D. D overhears A with probability q_{AD} and C overhears B with probability q_{BC}.

arrival rates $\lambda_1 = \lambda_2 = \lambda$. Under strategy 1: "Transmit only native packets" the maximum λ achieved is $\frac{2r_1r_2}{3r_2+r_1}$ while under strategy 2 "Transmit the encoded packet $P_{AC} \oplus P_{BD}$ and retransmit as native all undecoded P_{AC} packets" it is $\frac{2r_1r_2\min\{r_1,r_2\}}{2r_2\min\{r_1,r_2\}+r_1r_2+(1-q)r_1\min\{r_1,r_2\}}$. By setting these two equal, the threshold q_{thr} is obtained, such that for $q \geq q_{thr}$ NC is beneficial. Solving, we get

$$q_{thr} = \begin{cases} 0 & r_1 \leq r_2 \\ \frac{r_1-r_2}{r_1} & r_1 > r_2, \end{cases}$$

indicating that the strategy "encode as many packets as possible" used in [1] (where equal rates are assumed) is not optimal when $r_1 > r_2$. Also, a fixed threshold policy is bound to fail in a variable rate scenario. The phenomenon becomes non-trivial to visualize and solve if one considers several flows with some of them having common sources or destinations.

B. Scheduling with feedback motivation

To see the impact of feedback, consider the cross topology of Figure 3 with $q_{AD} + q_{BC} \geq 1$ and $q_{AD} < 1$, $q_{BC} < 1$ and $r_1 = r_2$. Assume the use of a reasonable scheduling policy which selects the packet $P_{AC} \oplus P_{BD}$ iff $q_{AD} + q_{BC} \geq 1$, which corresponds to decoding of at least one packet on the average and otherwise it selects the transmission of a native packet. Now assume that we only have two packets to send, one for each flow, and it happened that these packets were both not overheard at uplink time. The scheduler, oblivious of this unhappy occasion, will encode the two packets (since $q_{AD} + q_{BC} \geq 1$) and transmit the coded packet. In this case no ACK will be received since it is impossible for both destinations to decode. A scheduler that disregards feedback information will keep on sending the encoded combination of the same packets and the system will never escape the deadlock. Note, that we may let the above packets mix in the queues with other packets and thus reduce the probability of a deadlock. However, the throughput will decrease in any case. The reason this happens is because the scheduler actually has in its possession new information which is not taken into account in future scheduling decisions. Indeed, once no ACK messages were received, the scheduler can deduce the information that none of the two packets is overheard. Then it should append this information to the particular packets and treat them differently.

III. Problem Formulation

Consider the downlink of a network with F flows and F corresponding queues, similar to the one in Figure 2. Let $\tau = 1, 2, \ldots$ be time instances at which the relay is ready to transmit, also called decision slots. We assume a random process $A_i(\tau)$, with $\mathbb{E}[A_i(\tau)] = \lambda_i$, of packets of flow i arriving at the corresponding queue at the relay node along with a randomly chosen state vector $\mathbf{s}_k^i$ for each such packet.

At each decision slot, the scheduler located at the relay chooses one control I from a set of controls $\mathcal{I}$. Each control consists of activating either a single queue, or a pair of queues. A control having only one queue activated corresponds to a native packet transmission. A control with two activated queues corresponds to the transmission of the XOR of two packets. Although the chosen control dictates which packets are transmitted, the service (successful transmission) of a packet in a queue is determined also by the vector $\mathbf{s}_k^i$ which is uncontrollable. We use the definitions of queue network stability from [9].

Definition 1 (Queue stability): The queue n^i of flow i is called stable iff

$$\limsup_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\big[x^i(\tau)\big] < \infty.$$

Definition 2 (Network stability): The network is stable iff all queues are stable.

We seek to find a policy π which selects an appropriate control at each decision slot such that the downlink of the network is stabilized. The *stability region* of a policy π, Λ_π is the closure of the set of arrival rates for which the network is stable when π is in use. The *throughput region* Λ is the closure of the set of all arrival vectors $\boldsymbol{\lambda} = \{\lambda_i\}$ stabilizable by any policy. A control policy π^* is called *throughput optimal* if its stability region is Λ, see [9], [10] for implications of these definitions.

IV. The virtual network mechanism

A. Description of the virtual network for two flows

We begin by illustrating the virtual network mechanism for the case of two flows ($F = 2$), thus the allowable controls are either to schedule a native packet from one of the two flows, or the XOR combination of the pair. A description for $F > 2$ can be found in the following subsection.

We will use the notation $i, \bar{i}$ with $\bar{1} \doteq 2$ and $\bar{2} \doteq 1$. Any particular packet P_k^i of flow i can be categorized by the scheduler according to the state estimation of the event $\mathbb{1}_{\bar{i}}\{P_k^i\}$: "the destination of flow $\bar{i}$ possesses P_k^i" as follows:

- Unknown state (u): In this case the scheduler does not know $\mathbb{1}_{\bar{i}}\{P_k^i\}$ deterministically. It possesses, nevertheless, the statistical information $\mathbb{E}\big[\mathbb{1}_{\bar{i}}\{P_k^i\}\big] \doteq q_{i\bar{i}}$.
- Good state (g): In this case the scheduler knows that $\mathbb{1}_{\bar{i}}\{P_k^i\} = 1$.
- Bad state (b): In this case the scheduler knows that $\mathbb{1}_{\bar{i}}\{P_k^i\}$=0.

In order to group packets with the same properties together, we define for each flow i a *directional subnetwork* $\mathcal{G}_i =$

I	w^1_{ud}	w^1_{ug}	w^1_{ub}	w^1_{gd}	w^1_{gg}	w^1_{bd}	w^1_{bb}
$\{n^1_u, n^2_u\}$	q_{21}	$(1-q_{21})q_{12}$	$(1-q_{21})(1-q_{12})$	0	0	0	0
$\{n^1_u, n^2_g\}$	1	0	0	0	0	0	0
$\{n^1_u, n^2_b\}$	0	q_{12}	$1-q_{12}$	0	0	0	0
$\{n^1_g, n^2_u\}$	0	0	0	q_{21}	$1-q_{21}$	0	0
$\{n^1_g, n^2_g\}$	0	0	0	1	0	0	0
$\{n^1_g, n^2_b\}$	0	0	0	0	1	0	0
$\{n^1_b, n^2_u\}$	0	0	0	0	0	q_{21}	$1-q_{21}$
$\{n^1_b, n^2_g\}$	0	0	0	0	0	1	0
$\{n^1_b, n^2_b\}$	0	0	0	0	0	0	1

TABLE I
TRANSITION WEIGHTS IN THE SUBNETWORK $\mathcal{G}_1$ FOR ALL PAIR CONTROLS.

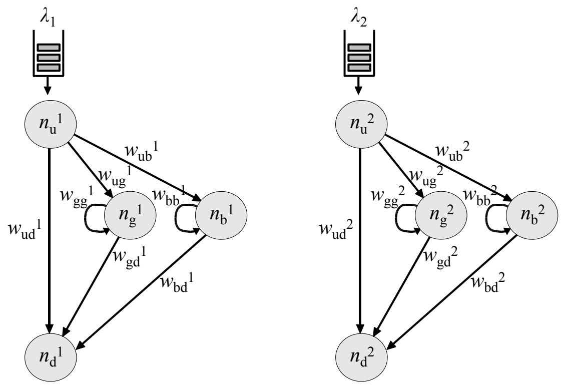

Fig. 4. The virtual network for the case of 2 flows consists of two subnetworks, one for each flow.

$(\mathcal{N}_i, \mathcal{E}_i)$ having one node for each possible overhearing state and one for the destination. We write $\mathcal{N}_i = \{n^i_u, n^i_g, n^i_b, n^i_d\}$, $i = 1, 2$ for the virtual node set. In order to capture the possible state transitions, we define the virtual link set to consist of the ordered pairs $\mathcal{E}_i = \{e^i_{ud}, e^i_{gd}, e^i_{bd}, e^i_{ug}, e^i_{ub}, e^i_{gg}, e^i_{bb}\}$, with $e^i_{km} \doteq (n^i_k, n^i_m)$ and $i = 1, 2$. The virtual network is the union of the two subnetworks for $i = 1, 2$, see Fig. 4.

We associate each node of the virtual network with a queue backlog (i.e. node/queue n^i_u has a backlog x^i_u). The packets of flow i enter the virtual network at the node n^i_u, they are routed inside the network and eventually leave the system when they reach node n^i_d -thus the destination node queue is always empty. At each time slot the scheduler selects a control which corresponds to activating either a) one node from the virtual network e.g. $I = \{n^1_b\}$ or b) two nodes from two different subnetworks, excluding destination nodes, e.g. $I = \{n^1_u, n^2_g\}$. Once a control is taken, the first packet in each selected node (head-of-line, HOL, packet) is transmitted. The node to which a packet is transferred after a transmission is random and its probability law depends on the chosen control. While the scheduler knows the probability law, it does not know the actual destination - unless the probability of transition to a particular destination is 1. To reflect this, each link (k, m) of subnetwork $\mathcal{G}_i$ is associated with a probability weight $w^i_{km}(I)$ that depends on the taken control (sometimes and if there is no possibility for confusion we may omit superscript i for simplicity). For any given control I and any node k we have $\sum_{m \in \mathcal{N}^k} w_{km}(I) = 1$ where $\mathcal{N}^k$ is the set of neighbors of k.

Next we describe all possible transitions of a packet of flow 1 when the control $I = \{n^1_u, n^2_u\}$ is taken. We write w^1_{ud} to represent $w^1_{(n^1_u, n^1_d)}$. To determine the transitions of the packet of flow 1, we check the state of the packet of flow 2 which in this case it is unknown. We can calculate the probability of correct decoding for destination δ_1, which is q_{21}. If the decoding fails (it happens with probability $1 - q_{21}$), then the scheduler will learn the state of the packet of flow 1, based on the received feedback from the two destinations. It can be the good state with probability q_{12} (thus moves to the node n^1_g with probability $q_{12}(1 - q_{21})$) or the bad state with probability $1 - q_{12}$ (in which case it moves to the node n^1_b with probability $(1 - q_{12})(1 - q_{21})$). Thus for the chosen control we get $w^1_{ud}(I) = q_{21}$, $w^1_{ug}(I) = q_{12}(1 - q_{21})$ and $w^1_{ub}(I) = (1 - q_{12})(1 - q_{21})$. Similarly we develop the link weights for subnetwork $\mathcal{G}_2$ by exchanging 1 and 2.

If the scheduler selects the control $I = \{n^1_u, n^2_g\}$ then the packet of flow 1 will definitely leave the system. Thus now we have $w^1_{ud}(I) = 1$ and the rest weights are zero. Instead if the scheduler selects the control $I = \{n^1_u, n^2_b\}$, we obtain $w^1_{ud}(I) = 0$, $w^1_{ug}(I) = q_{12}$ and $w^1_{ub}(I) = 1 - q_{12}$. Also, given the controls $\{n^1_g, n^2_u\}$ or $\{n^1_b, n^2_u\}$, note that the packet of flow 1 will be either sent to destination or stay at the same node since the packet state is known to the scheduler and cannot change. The weights of the subnetwork $\mathcal{G}_1$ for all controls that activate pairs of queues are given in the table I.

1) Virtual network for F flows and pairwise XOR: The extension of the virtual network mechanism to the case of F flows using pairwise XOR is natural due to the complete feedback information provided. Each packet state is now characterized by the scheduler knowledge about the state vector that contains information about packet knowledge in all $F - 1$ receivers. Thus, instead of having three states as before, now we have 3^{F-1} states. Correspondingly, each of the F independently created subnetworks will have $3^{F-1} + 1$ nodes, one for each state and one for the destination. We use the generalized notation for the nodes $n^i_{\mathbf{c}}$, where $i \in \mathcal{F}$ indicates the flow and $\mathbf{c}$ is a ternary state vector taking values in $\{u, g, b\}^F$ with $c(i) = g$ by convention (in the previous subsection and Figures 4, 5 we have chosen to omit this element for presentation simplicity). See Figure 5 for an example with $F = 3$. The routing of packets in this network follows exactly the same rules as in the 2–flows case. Specifically,

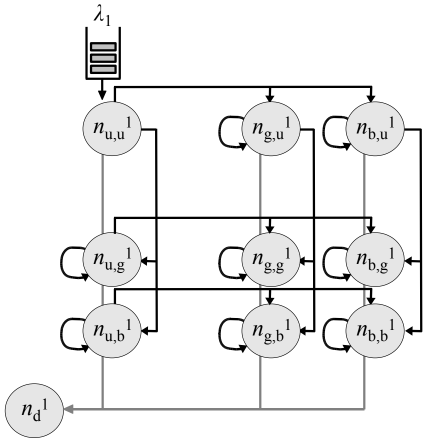

Fig. 5. The virtual subnetwork of flow 1 for the case of 3 flows. the gray arrows connect all nodes to the destination.

when combining two packets, say belonging to flow i and $j \neq i$, the routing is determined by the elements $c_j(i)$ and $c_i(j)$ correspondingly and by the overhearing probabilities q_{ji} and q_{ij}. Also, in case of failure, only the states $c_i(j)$ and $c_j(i)$ become affected. An exception to this is when a receiver node is the destination of more than one flow. Since, the overhearing event concerns primarily the nodes and not the flows, the feedback information should update all the states of vector $\mathbf{c}$ that correspond to flows that have as destination this given node. Note that this complicates the construction of the virtual network but does not affect the analysis for the stability of it. In Figure 5, all possible links are identified for the case of disjoint receivers. Note for example, that n^1_{uu} is not directly connected with n^1_{gg} since in order to move from n^1_{uu} to n^1_{gg} at least two transmissions are needed.

The special case when we consider symmetric flows i, j is captured by the model by simply setting $q_{ij} = q_{ji} = 1$. In this case, the problem becomes simpler as regards these two flows. Note that for $F = 2$ and $1, 2$ symmetric flows, the problem is the same with [4]. However, in the general case there can be symmetric and non-symmetric flows arriving at the relay.

The problem of scheduling the packets in the original downlink system is mapped to the problem of scheduling the packets in the corresponding virtual network. By representing one queue in the original problem with a subnetwork, we have added another dimension which captures the knowledge obtained progressively about the overhearing states of each packet using the feedback received at the relay. For two packets backlogged in the same virtual node, the scheduler has the same belief and thus these two packets are stochastically equivalent in terms of transmission efficiency (i.e. the have the same expected reward), a desired property that did not exist in the previous set-up.

B. Optimal control of the virtual network

The virtual network developed in section IV-A differs from the general network treated in [9] in the following important aspect. When a control $I \in \mathcal{I}$ is selected and a packet is chosen for transmission by node j, the destination of the packet is random: one of the outgoing neighbors of node j. Hence the results in [9] cannot be applied directly. However, the methods used in [9] can be extended to analyze the network of interest and to develop an algorithm with maximal stability region.

The virtual network $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ consists of the union of F subnetworks, $\mathcal{G}_i = (\mathcal{N}_i, \mathcal{E}_i)$, $i = 1, 2$. Let $\mathcal{E}^j, \mathcal{N}^j$ be respectively the set of outgoing links and neighbors of node $j \in \mathcal{N}$, and x_j its backlog. Also, for a given control I, let $w_{(j,k)}(I)$ be the probability that a packet transmitted by node j ends up at node $k \in \mathcal{N}^j$.

We assume a slotted system and transmission rates as follows: when control $I \in \mathcal{I}$ is chosen, the maximum number of packets that may be "transmitted" by node j over the link set $\mathcal{E}^j$ is $\mu_j(I)$.

- If a control I involves transmission from a single node j located at subnetwork i, then

$$\mu_k(I) = \begin{cases} r_i^{down} & \text{if } k = j \\ 0 & \text{otherwise} \end{cases},$$

  where r_i^{down} is the maximum number of packets that may be transmitted from the relay to δ_i in a slot.
- If the control involves an XOR packet from nodes j_1, j_2 located at subnetworks $\mathcal{G}_1, \mathcal{G}_2$ respectively, then

$$\mu_k(I) = \begin{cases} \min\{r_1^{down}, r_2^{down}\} & \text{if } k = j_1 \text{or } k = j_2 \\ 0 & \text{otherwise} \end{cases}.$$

We can now present the throughput region of the system. Define the following set of flow variables for each of the virtual network links $\mathbf{f} = \{f_e, e \in \mathcal{E}\}$. For control $I \in \mathcal{I}$ define the set of vectors $\mathbf{f}$,

$$\begin{aligned} \Gamma(I) \quad = \quad & \{\mathbf{f} = \{f_e\} : e = (j,k), f_{(j,k)} = w_{(j,k)}(I)\hat{f}_j : \\ & 0 \le \hat{f}_j, \le \mu_j(I), j \in \mathcal{N}, k \in \mathcal{N}^j\} \end{aligned}$$

and define the convex hull of the sets $\Gamma(I), : I \in \mathcal{I}$, $\mathcal{C} = \text{conv}(\Gamma(I), I \in \mathcal{I})$. The throughput region of the system can now be defined:

Throughput Region: The throughput region of the system is the set of arrival rates $\boldsymbol{\lambda} = \{\lambda_j\}_{j \in \mathcal{N}}$, $\lambda_j \ge 0$, for which there exists a vector $\mathbf{f} \in \mathcal{C}$ such that for any node $j \in \mathcal{N}$ except the destination nodes it holds

$$\sum_{e=(k,j) \in \mathcal{E}^k} f_e + \lambda_j \le \sum_{e \in \mathcal{E}^j} f_e.$$

Evidently, we have $\lambda_j = 0, \forall j \in \mathcal{N} \setminus \{n_u^1, n_u^2\}$, thus the throughput region can be thought as a two-dimensional object. The derivation of the throughput region is based on an extension of the methodology developed in [11]. The same reasoning can be used to extend the characterization for the case of F flows and pairwise XORing, see [12]. Based on the throughput region described above and using Lyapunov function techniques as in [9], it can be shown that the algorithm described below has maximal stability region.

Algorithm 2: At each decision slot:

1) For each control $I = \{j\}, j \in \mathcal{G}$ form the cost $Z(I) = x_j \mu_j(I)$

2) For each control $I = \{i, j\}, i, j \in \mathcal{G}, j \neq i$
   - form the weights

$$z_i(I) = \max\left\{x_i - \sum_{k \in \mathcal{N}^i} w_{(i,k)}(I) x_k, 0\right\},$$

$$z_j(I) = \max\left\{x_j - \sum_{k \in \mathcal{N}^j} w_{(j,k)}(I) x_k, 0\right\},$$

   - and then the cost

$$Z(I) = z_i(I)\mu_i(I) + z_j(I)\mu_j(I).$$

3) Then select $I^* = \arg\max_{I \in \mathcal{I}}\{Z(I)\}$.

V. Conclusions

Scheduling of pairwise XORs with statistical overhearing information and feedback is optimized using the dynamic backpressure policy on a virtual network created for this problem. The virtual network mechanism and structure is explained and an optimal algorithm is proposed that stabilizes the system. Future work of interest pertains to generalizing the model, considering uplink and downlink jointly as well as multihop scheduling, allowing for storing of encoded packets, including channel erasures and developing approximation algorithms for the case of $|\mathcal{P}| > 2$.

Acknowledgments

The work is supported by the European Commission IST projects, STREP-FP7-INFO-ICT-224218: OPNEX and STREP-FP7-INFSO-ICT-215252: N-CRAVE.

References

[1] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft, “XORs in The Air: Practical Wireless Network Coding,” in *ACM SIGCOMM*, 2006.

[2] S. Rayanchu, S. Sen, J. Wu, S. Banerjee, and S. Sengupta, “Loss-Aware Network Coding for Unicast Wireless Sessions: Design, Implementation, and Performance Evaluation,” in *ACM SIGMETRICS*, 2008.

[3] E. Rozner, A. Iyer, Y. Mehta, L. Qiu, and M. Jafry, “ER: Efficient Retransmission Scheme for Wireless LANs,” in *ACM CONEXT*, 2007.

[4] P. Chaporkar and A. Proutiere, “Adaptive Network Coding and Scheduling for Maximizing Throughput in Wireless Networks,” in *ACM MOBICOM*, 2007.

[5] A. Eryilmaz and D. S. Lun, “Control for inter-session network coding,” in *in Proc. of ITA Workshop*, Feb. 2007.

[6] T. Ho, Y. Chang, and K. J. Han, “On constructive network coding for multiple unicasts,” in *in Proc. of 44th Allerton conference on Communication, Control and Computing*, Sep. 2006.

[7] P. Chaporkar, A. Proutiere, H. Asnani, and A. Karandikar, “Scheduling with limited information in wireless systems,” in *ACM MobiHoc*, 2009, pp. 75–84.

[8] A. Fu, E. Modiano, and J. Tsitsiklis, “Optimal transmission scheduling over a fading channel with energy and deadline constraints,” *IEEE Trans. on Wireless Communications*, vol. 5, no. 3, pp. 630–641, 2006.

[9] L. Georgiadis, M. Neely, and L. Tassiulas, “Resource allocation and cross-layer control in wireless networks,” *Foundations and Trends in Networking*, vol. 1, pp. 1–147, 2006.

[10] 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. 37, pp. 1936–1948, 1992.

[11] E. M. M.J. Neely and C. Rohrs, “Dynamic power allocation and routing for time-varying wireless networks,” *IEEE Journal on Selected Areas in Communications*, vol. 23, pp. 89–103, Jan. 2005.

[12] https://sites.google.com/site/gpasxos/reports/report_NC.pdf.