Throughput Regions and Optimal Policies in Wireless Networks with Opportunistic Routing

Lavy Libman* Georgios S. Paschos†,§ Leonidas Georgiadis‡,§ Xin Zhao*
*School of Computer Science and Engineering, University of New South Wales, Australia
†Laboratory for Information and Decision Systems, Massachussetts Institute of Technology, USA
‡Dept. of Electrical and Computer Engineering, Aristotle University of Thessaloniki, Greece
§Centre for Research & Technology Hellas – Informatics and Telematics Institute (CERTH-ITI), Greece
lavy.libman@unsw.edu.au, gpasxos@mit.edu, leonid@auth.gr, xinzhao@cse.unsw.edu.au

***Abstract*—*Opportunistic routing* in wireless networks has been proposed as a method to combat the volatility of wireless links, by leveraging their broadcast nature and choosing the next hop for each transmitted packet post-facto, using the actual reception outcomes at the respective neighbors, rather than based on *a priori* information. Much of the research on the topic has focused on protocol design issues, e.g. coordination mechanisms among the next-hop candidates; however, the fundamental performance bounds of the scheme are not yet known. In this paper we study the theoretical throughput region of opportunistic routing, for a generic network model with an arbitrary matrix of packet erasure probabilities between any two nodes, which cannot be mapped onto any classical model due to the existence of undirected broadcast from each node. We introduce a generic technique involving a transformation into a *virtual network* consisting of nodes corresponding to packet states in the original network, and define two different throughput-optimal scheduling policies in the virtual network, one based on a backpressure-like approach, and another that uses a dynamic programming algorithm which finds the minimum time to clear the system from any initial queued backlog. These policies can support both a unidirectional (half-duplex) flow between a given source and destination, and a bidirectional (full-duplex) connection with inter-session network coding in intermediate nodes.**

I. Introduction

The prevailing approach to network routing, widely used in various network-layer protocols in wired networks, is based on modeling the network as a graph of links between pairs of nodes and calculation of the shortest path in that graph. While this approach is used in traditional wireless routing protocols such as AODV and OLSR [1] as well, its success in wireless networks is limited since it does not take into account the broadcast nature of the wireless medium, nor the high volatility of wireless channels, e.g. caused by fading or local interference from other traffic. Indeed, much of the traditional research on routing in wireless networks focused predominantly on methods to overcome the interference and maximize network capacity for multiple co-existing flows, e.g. using channel assignment methods based on coloring of the link conflict graph (cf. [2] and references therein). In recent years, there has been growing attention towards methods that embrace the broadcast nature of wireless communications and aim to exploit the capability of nodes to overhear their peers' transmissions, from *cooperative relaying* in the physical layer to *opportunistic routing* and *wireless network coding* techniques at higher layers, which are the focus of this paper.

G.S. Paschos is supported by the Greek National Strategic Reference Framework (NSRF) Action: Support of Post-Doctoral Researchers (project WiNC). L. Georgiadis is supported by the EU FP7 ICT Programme grant 257616 (CONECT – Cooperative Networking for High Capacity Transport Architectures). X. Zhao was supported by the Australian Research Council (ARC) Discovery grant DP1096353 (Multimedia Content Distribution over Multihop Broadband Wireless Networks).

The term *opportunistic routing* (OR) is used collectively to describe routing schemes where the next hop of each packet is decided in real time, depending on the instantaneous performance of wireless links, rather than predetermined in advance. Some of the early examples of the concept include SDF [3] and GeRaF [4], which used explicit coordination schemes involving exchanges of acknowledgments or RTS/CTS frames between the transmitter and the potential next-hop forwarders. In the ExOR protocol [5], batches of packets are simply broadcast without acknowledgments, and after every batch, each neighbor (in a predetermined priority order) retransmits the packets it has successfully received, skipping those that it could overhear being already forwarded by one of the other neighbors. The demonstration of the significant boost in throughput achievable by ExOR has led to further more recent extensions, such as *modrate*, which optimizes the rate selection jointly with exploiting overhearing opportunities [6], and opportunistic routing in the presence of correlated links [7].

The idea of network coding (NC), where data packets either from the same flow (*intra-flow* coding) or multiple flows (*inter-flow* coding) are mixed so that a single transmission holds different information content for different receivers, has been applied in a number of proposed wireless network protocols, in combination with opportunistic routing. Examples of protocols that use intra-flow coding include MORE, which boosts the reliability of ExOR by transmitting linear combinations rather than individual packets in each batch [8], and its further extensions such as CodeOR and PipelineOR, which allow a "sliding window" of multiple batches of coded packets [9], [10]. Inter-flow coding in wireless networks was first introduced by COPE, which uses broadcasting of bitwise-XOR combinations of packets from different flows (or from opposite directions of a full-duplex flow between the same endpoints) [11]. COPE itself does not involve opportunistic routing (the flow routes are determined by an underlying routing protocol); furthermore, the decision about transmitting such combinations requires the knowledge of the set of previously overheard packets in each neighbor node, which is achieved via explicit periodical "overhearing reports". These shortcomings are addressed in COPR, where the scheduling and coding decisions are based on an oblivious backpressure algorithm [12], and NCRAWL, where such decisions are optimized based on statistical properties of the links [13].

Unlike the above studies (and most of the other related work in this space) that focus on improving the performance of specific protocols, in this paper we focus on a more fundamental question which has remained outside the scope of existing research, namely: what is the theoretical throughput limit of a wireless system with opportunistic routing? Such questions about the theoretical network capacity or *throughput region* (i.e. the maximal region of arrival rates that can be serviced by the system in a stable manner) date back as far as the seminal work in [14], which introduced the backpressure algorithm and established its throughput optimality in general queueing networks. More recently, a detailed treatment of capacity and stability in wireless networks, with a focus on cross-layer design approaches, was provided in [15]. An adaption of backpressure routing for wireless networks with opportunistic multi-receiver capabilities was developed in [16], while the additional possibilities (and corresponding constraints) arising with network coding have been studied both for general network topologies [17] and more specifically in wireless contexts [18], [19]. Without exception, all of the above studies are limited to considering only the case of *single-copy* routing. Indeed, the classical model of a network of interconnected queues cannot be directly used with any scheme that involves replication of packets in multiple nodes, such as opportunistic routing, due to the inherent dependencies among queue states (e.g., a packet arriving at the destination automatically disappears from all queues where its copies are present) that cannot be accounted for by legacy queueing models.

Motivated by the above, in this work we consider the maximal stable throughput region in a wireless network where packets can be overheard by multiple nodes and routed opportunistically to their destinations. Specifically, we consider a model consisting of two endpoints A and B and a number of intermediate relay nodes, with a known matrix of packet overhearing (alternatively, erasure) probabilities between every pair of nodes. We assume two unicast flows in opposite directions between A and B, fed by stochastic arrivals of packets, and seek throughput-optimal policies that can be applied by an omniscient scheduler, which is allowed to utilize information about which node has overheard which packets at any time. The control action that the scheduler is allowed to decide at any time is a transmission of either a native packet or a XOR of two packets from opposite directions (i.e. inter-flow coding). In this study, we do not consider coded transmissions of packets from the same flow, since intra-flow coding is known not to increase the throughput of a unicast flow, and its practical benefits (such as lower sensitivity to topology parameters) are irrelevant in the context of our current study [20]. The extension of the model of this paper to include scenarios such as multicast, where intra-flow coding can be beneficial [21], is left for future work.

Our contribution in this paper is twofold. First, we provide a generic technique of transforming the physical wireless network to an equivalent virtual one, where each virtual node queue holds packets overheard by a specific subset of the physical network nodes, and characterize the throughput region for the network model of the paper. We then demonstrate how known throughput-optimal policies from traditional network stability theory, such as the backpressure or epoch-based evacuation policies [22], can be adapted to the virtual network and discuss the impact of the virtual network transformation on some of the known properties of these policies.

We emphasize that this paper does not aim to propose a practical protocol to attain the maximal throughput region. Rather, our discussion of throughput-optimal coding and routing focuses on centralized policies, with full information about the state of all nodes and links at all times. While such policies may not be directly suitable for implementation in a real network, they (in the same way as the classical backpressure algorithm) serve as a baseline for future work on practical distributed heuristics and approximations for networks with opportunistic routing, and provide a useful upper bound and benchmark for the performance of any such practical schemes.

The rest of the paper is structured as follows. Section II formally defines the system model and assumptions. Section III presents the details of the virtual network transformation and the equivalence of scheduling policies between the physical and the virtual network representation. Sections IV and V discusses the backpressure-based and the epoch-based evacuation policies and their adaptations to the virtual network. Section VI discusses the differences between the two policies with respect to other performance metrics, such as delay and average number of transmissions per packet, and suggests a hybrid application of both policies depending on the network load. Finally, the paper is concluded in Section VII.

II. Model Description

We consider a static wireless network consisting of a distinguished pair of endpoint nodes, A and B, plus K potential relay nodes R_k ($k = 1, \dots, K$). A packet transmission from any node i can be successfully overheard by any other node j with a probability P_{ij}, where $i, j \in \{A, B, R_k\}$; equivalently, the channel between nodes i, j is a packet erasure channel with an erasure probability of $1 - P_{ij}$. The channels are assumed to be memoryless (thus, the sequence of packet erasures in each channel over time is i.i.d), and for simplicity of presentation, we also assume the erasure probabilities to be independent across channels; however, the subsequent analysis and the throughput-optimal policies described therein can be extended to allow correlated channels in a straightforward manner.

We emphasize that the matrix P_{ij} need not necessarily be symmetrical, and may contain zero elements (corresponding to pairs of nodes unable to overhear each other at all). Thus, the model is not limited to a single-hop setting; for example, it is possible for the shortest path between A and B to consist of at least H hops, if the relays are partitioned into $H - 1$ groups, such that channels with positive probability exist only between nodes within the same group or adjacent groups. Nevertheless, we assume that all the nodes are within a common range of interference; i.e., only one transmission is allowed to take place anywhere in the network at any one time.

We consider a bidirectional flow consisting of packets that are generated at each of the endpoints A, B according to some independent random process, to be delivered to the respective opposite endpoint. As long as there exist packets not yet arrived to their destination, a *scheduling policy* chooses one node at a time to broadcast either a single packet that it has previously overheard, or a coded combination of two packets from opposite directions, e.g. a bitwise XOR as in COPE [11]. Clearly, this model includes a *unidirectional* flow between a single source-destination pair as a special case, simply by setting an empty (zero-rate) packet arrival process in the opposite direction. For simplicity, throughout the paper we assume that all packets are of identical size and all nodes transmit at the same raw bitrate; hence, all transmissions take the same duration which we call a *slot*, and we shall use normalized units of packets per slot for all throughput-related quantities in the network. However, we emphasize that the methods and policies described hereafter can be readily extended to include variable packet sizes and/or heterogeneous transmission bitrates.

III. The Virtual Network Transformation

In an opportunistic network where copies of the same packet can exist in several nodes, scheduling policies based on queue occupancy of individual nodes (such as the classical backpressure algorithm and its variations) cannot be directly applied, since simple counting of packets queued in a node ignores inherent dependencies that cannot be handled within traditional models. For example, if a particular packet is overheard by two relay nodes R_1 and R_2, and is then broadcast by R_1 and successfully received at the destination, it immediately disappears from the queue of R_2 as well. We resolve this limitation by describing a transformation into a *virtual network*, whose nodes consist of all possible subsets of real nodes where a packet may have been overheard.

Specifically, the virtual network consists of the following nodes:

- for every subset of relays $\mathcal{S} \subseteq \{R_1, \ldots, R_K\}$ (including the empty set $\mathcal{S} = \varnothing$), a node labeled $\langle A, \mathcal{S}\rangle$ and a node labeled $\langle B, \mathcal{S}\rangle$;
- a *destination node* for packets from A to B, labeled $\langle \mathcal{B}\rangle$;
- a *destination node* for packets from B to A, labeled $\langle \mathcal{A}\rangle$.

Furthermore, the following edges are defined between nodes in the virtual network:

- for every node $\langle A, S\rangle$, an edge from that node to every other node $\langle A, S'\rangle$ such that $S \subseteq S'$, and to the destination node $\langle \mathcal{B}\rangle$;
- for every node $\langle B, S\rangle$, an edge from that node to every other node $\langle B, S'\rangle$ such that $S \subseteq S'$, and to the destination node $\langle \mathcal{A}\rangle$.

For brevity, we henceforth refer to the flow of packets from A to B as the *forward* flow, and that of packets from B to A as the *reverse* flow.

With every virtual node $\langle A, \mathcal{S}\rangle$ we associate a virtual queue, which at any point in time consists of all the forward packets that have been overheard by every relay $R_k \in \mathcal{S}$, not overheard by any $R_k \notin \mathcal{S}$, and not yet received by B. Thus, each packet can only be associated with one virtual queue at a time. Intuitively, when a packet is in the queue of $\langle A, \mathcal{S}\rangle$ and is then broadcast (either by A or a relay $R_k \in \mathcal{S}$), it can either remain in the same queue (if not received/overheard by any new physical nodes), or move to a queue of another virtual node $\langle A, \mathcal{S} \cup \mathcal{S}^{new}\rangle$ (corresponding to being successfully received by new relays in the set $\mathcal{S}^{new}$, but not by B), or move to the virtual destination node $\langle \mathcal{B}\rangle$ and leave the system, if successfully received by the destination. The probabilities of these transitions depend on the channel probability matrix (P_{ij}), as well as the choice of physical node doing the broadcast. The same definition of virtual queues applies to nodes $\langle B, \mathcal{S}\rangle$, for packets in the reverse direction (from B to A). We now proceed to formally define the relevant notation.

An *action* $\mathcal{I}$, chosen by the scheduling policy in the virtual network in a time slot, is defined as a tuple (N, V_a, V_b) where:

- N is a physical node, $N \in \{A, B, R_1 \ldots R_K\}$;
- V_a is a node in the virtual half-network corresponding to the forward flow (i.e. either $\langle A, \mathcal{S}\rangle$ or $\langle \mathcal{B}\rangle$);
- V_b is a node in the virtual half-network corresponding to the reverse flow (i.e. either $\langle B, \mathcal{S}\rangle$ or $\langle \mathcal{A}\rangle$);
- if V_a is of the form $\langle A, \mathcal{S}\rangle$ (i.e. it is not the virtual destination node $\langle \mathcal{B}\rangle$), then $N \in \{A\} \cup \mathcal{S}$;
- if V_b is of the form $\langle B, \mathcal{S}\rangle$, then $N \in \{B\} \cup \mathcal{S}$.

Generally, an action (N, V_a, V_b) corresponds to a transmission by node N of a coded combination of the packets taken from the head of the queues of V_a and V_b; however, if V_a or V_b is a virtual destination node (which does not have a queue associated with it by definition), then the action represents a transmission of a single native packet from the opposite flow.

For completeness, we also define the *idle action* $\varnothing$, representing a slot in which no transmission is attempted at all. This is introduced to address the possibility of an empty network, and avoids some technical complications by allowing us to assume, without loss of generality, that any non-idle action always involves transmissions from non-empty queues only. We denote $\mathcal{C}$ to be the set of all possible actions, i.e. tuples satisfying the above conditions, plus the idle action $\varnothing$.

We now consider the transition probabilities in the virtual network that are induced by a choice of a particular action $\mathcal{I} = (N, V_a, V_b)$. We first present the transition probabilities in the virtual half-network corresponding to the forward flow, i.e. from virtual node V_a to its neighbors. Without loss of generality, we assume that V_a has a non-empty queue.* We denote $T^{\mathcal{I}}(V'_a)$ to be the transition probability to the virtual node V'_a, i.e. the probability that a packet taken from the queue of V_a is transferred to the queue of V'_a (or leaves the network, if $V'_a = \langle \mathcal{B}\rangle$).

When node N broadcasts a packet, it is overheard by the destination with a probability P_{NB}. Accordingly,

$$T^{\mathcal{I}}(\mathcal{B}) = P_{NB}, \tag{1}$$

*This implies that V_a is of the form $\langle A, \mathcal{S}\rangle$; no transition probabilities are defined when $V_a = \langle \mathcal{B}\rangle$, i.e. when the action does not involve a packet from the forward flow at all.

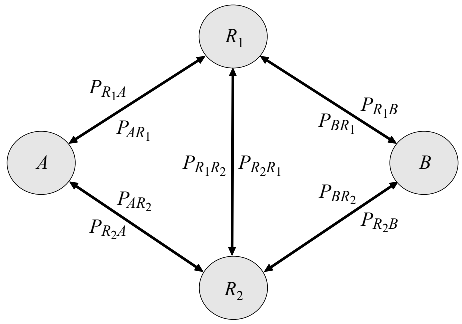


Fig. 1: A physical network with $K = 2$ relay nodes and no direct link between endpoints A, B.

regardless of whether the packet is overheard by any other relays. For all other transitions, one must distinguish between a broadcast of a single native packet (i.e. $V_b = \langle \mathcal{A} \rangle$ in the opposite virtual half-network), and that of a combined coded packet. In the former case, the transition probability from $V_a = \langle A, \mathcal{S} \rangle$ to $V'_a = \langle A, \mathcal{S}' \rangle$ is calculated as follows:

$$T^{\mathcal{I}}(V'_a) = \begin{cases} (1 - P_{NB}) \cdot \prod\limits_{R \in \mathcal{S}' \setminus \mathcal{S}} P_{NR} \cdot \prod\limits_{R \notin \mathcal{S}'} (1 - P_{NR}) & \mathcal{S} \subseteq \mathcal{S}' \\ 0 & \mathcal{S} \not\subseteq \mathcal{S}' \end{cases} \tag{2}$$

Note that this formula includes the case where $S' = S$ (i.e. the transmission fails to be received by any new relays), using the convention that a product over an empty set is defined as 1.

Otherwise, i.e. when a combined coded packet is transmitted, a transition from $\langle A, \mathcal{S} \rangle$ to $\langle A, \mathcal{S}' \rangle$ requires not only that all nodes in $\mathcal{S}' \setminus \mathcal{S}$ overhear the transmitted packet, but, furthermore, are able to decode it; this is due to our stipulation that coded packets are not allowed to be stored in the relays and must be discarded if not decoded immediately. Accordingly, if $V_b = \langle B, \mathcal{S}_b \rangle$, then every relay in $\mathcal{S}' \setminus \mathcal{S}$ must be a member of $\mathcal{S}_b$, i.e. have already previously overheard the corresponding packet in the opposite direction. Hence, for this case, we have

$$T^{\mathcal{I}}(V'_a) = \begin{cases} 0 & \text{if } \mathcal{S} \not\subseteq \mathcal{S}' \text{ or } (\mathcal{S}' \setminus \mathcal{S}) \not\subseteq \mathcal{S}_b; \\ (1 - P_{NB}) \cdot \prod\limits_{R \in \mathcal{S}' \setminus \mathcal{S}} P_{NR} \cdot \prod\limits_{R \in \mathcal{S}_b \setminus \mathcal{S}'} (1 - P_{NR}) & \text{otherwise.} \end{cases} \tag{3}$$

The transition probabilities for packets in the reverse flow $T^{\mathcal{I}}(V'_b)$, i.e. from the virtual node V_b, are defined in the same manner as above and are omitted for brevity. It should be noted that, even for coded packet transmissions, the transition probabilities in the two virtual half-networks are independent, as they always depend on erasure probabilities in separate (non-overlapping) sets of links.

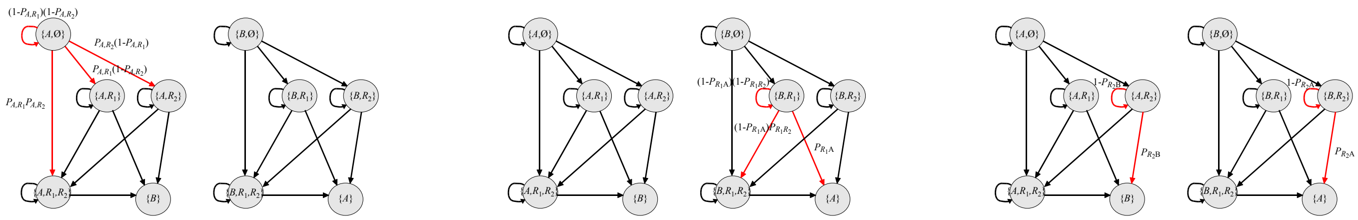
To illustrate the virtual network transformation, consider the example network depicted in Figure 1, with two relay nodes ($K = 2$) and no direct link between nodes A, B (i.e. $P_{AB} = P_{BA} = 0$). Figure 2 shows the construction of the corresponding virtual network and the possible transitions for three different control actions. The virtual network consists of two halves (the forward and the reverse half); the nodes in the figure are labeled after the states (i.e. subsets of physical nodes) they represent respectively, and all possible transitions are denoted by solid arrows. First, Figure 2a describes the control action $(A, \langle A, \emptyset \rangle, \langle \mathcal{A} \rangle)$, i.e., the physical node A is transmitting a packet destined to B that has not yet been heard by any other node, and with no coding with packets from the reverse flow. The probabilities of the transitions possible with this control action are shown next to the respective arrows, departing from node $\langle A, \emptyset \rangle$. Then, Figure 2b shows a different control action, $(R_1, \langle \mathcal{B} \rangle, \langle B, \{R_1\} \rangle)$, i.e., the physical node R_1 is transmitting a reverse packet destined to A, that was previously overheard only by node R_1 (apart from the source B). Again there is no mixing with the opposite flow, and the relevant transition arrows from node $\langle B, \{R_1\} \rangle$ are annotated with the corresponding probabilities. Finally, Figure 2c showcases the control action $(R_2, \langle A, \{R_2\} \rangle, \langle B, \{R_2\} \rangle)$, i.e. the physical node R_2 transmits a coded combination of two packets from the forward and reverse flows, where the forward packet (destined to B) is previously overheard by nodes A, R_2 while the reverse one (destined to A) is previously overheard by B, R_2. In this case, a transition can occur independently in each virtual half-network, from the respective node to the destination. Note that the transitions to virtual nodes $\langle A, \{R_1, R_2\} \rangle$ and $\langle B, \{R_1, R_2\} \rangle$ are not possible in this case, since node R_1 cannot decode the transmitted packet.

We conclude this section by noting that the size of the resulting virtual network corresponds to the number of distinct node subsets that a packet can be overheard by. While, in principle, this can be exponential in the size of the physical network, in reality the number of peers that can overhear and cooperatively forward a packet transmitted by a given node is restrained by practical implementation constraints (e.g. a fixed size of header fields used for the underlying coordination protocol [3], [5]), implying that the size of the relevant portion of the virtual network remains bounded even for a large network.

IV. The Backpressure Policy and Throughput Region in the Virtual Network

The virtual network construction described above ensures that any scheduling policy based on the state of the physical system can be emulated by a corresponding policy in the virtual network. More formally, if the system state (of the physical network) is generically defined as a vector representing the number of packets overheard at each node, and the scheduling policy is defined as a mapping from the system state to a transmission action to be taken in the time slot, then the space of all possible policies is in one-to-one correspondence with the possible mappings in the virtual network from node queue length vectors to actions.[†] In addition, since there can be at most a finite number of physical copies corresponding to any packet in a queue in the virtual network, any of the common definitions of stability in queueing networks [23] is equivalent for the physical and the virtual network. Consequently, the throughput region of arrival rates is identical in both networks, which implies that we can henceforth focus exclusively on

[†]It is implicitly assumed that scheduling policies in the physical network are allowed the same set of actions as in the virtual network, i.e. either transmissions of single native packets or of a XOR combination of two packets from opposing flows.



(a) (b) (c)

Fig. 2: Illustration of the virtual network transformation.

considering scheduling policies and their properties (such as throughput-optimality) in the virtual network. We point out that the above equivalence still holds even if non-Markovian policies are allowed (i.e. policies that consider the entire history of the system state rather than just the current value), or if packets within a flow are not all treated equally but rather belong to a number of priority classes (by defining corresponding priority queues in the virtual network nodes).

We note that the characterization of the throughput region for a standard network with constrained actions [14] cannot be directly applied for the virtual network since it does not satisfy some of the standard model assumptions; e.g., a control action involving a transmission of a packet from a virtual node does not uniquely determine the next hop (queue) to which the packet arrives, as the transitions take place over random subsets of links. To that end, we quote a result from [24], where a similar model of a virtual network with random transitions was employed for a different problem context. Let $\mathcal{N}, \mathcal{E}$ denote the set of virtual nodes and virtual edges, respectively, and for each virtual node $v \in \mathcal{N}$ let $\mathcal{E}_{in}(v), \mathcal{E}_{out}(v)$ be the set of the node's incoming and outgoing edges, respectively. Also, let $\mathbf{f} = \{f_e : e \in \mathcal{E}\}$ denote a vector of average service rates in each edge. For any given action $\mathcal{I} \in \mathcal{C}$, define $\Gamma(\mathcal{I})$ to be the set of rate vectors $\mathbf{f}$ satisfying the following property: there exists some value of $0 \leq \mu_{\mathcal{I}} \leq 1$ such that, for all $e = (v, v') \in \mathcal{E}$,

$$f_e = \begin{cases} T^{\mathcal{I}}(v')\mu_{\mathcal{I}} & \text{if } v \in \{V_a, V_b\} \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

Theorem 1 (Throughput region of the virtual network [24]). *The throughput region is the closure of the set of arrival rates $\lambda = \{\lambda_v\}, v \in \mathcal{N}$ for which there exists a vector $\mathbf{f}$ in the convex hull of $\{\Gamma(\mathcal{I}) : \mathcal{I} \in \mathcal{C}\}$ such that, for any node $v \in \mathcal{N}$,*

$$\sum_{e \in \mathcal{E}_{in}(v)} f_e + \lambda_v \leq \sum_{e \in \mathcal{E}_{out}(v)} f_e.$$

Note that, in our case, by construction, $\lambda_v > 0$ only makes sense for the origin nodes in each half-network, i.e. $v = \langle A, \emptyset \rangle$ and $v = \langle B, \emptyset \rangle$; for all other nodes $\lambda_v = 0$.

We point out that Theorem 1 defines the throughput region of the system in a descriptive form. Indeed, any desired maximal point on the boundary of the throughput region (e.g. maximize λ_A subject to a fixed value of λ_B or vice versa, or maximize some desired linear combination of λ_A, λ_B) can be computed by assigning a variable of $\mu_{\mathcal{I}}$ to each action $\mathcal{I}$ (this is the weight of the action in the convex hull) and solving the corresponding linear program that arises from the inequalities of Theorem 1, together with the additional constraint that $\sum_{\mathcal{I}} \mu_{\mathcal{I}} \leq 1$.

We now describe a policy in the virtual network based on the idea of the well-known backpressure algorithm [14], that uses queue length differences between respective senders and receivers. To that end, we first define the backpressure associated with an action $\mathcal{I} = (N, V_a, V_b)$ in the virtual network, adapted to the fact that packets may transit to a random set of nodes, and the corresponding queue length differences therefore need to be taken in expectation.

We denote the vector of virtual queue lengths by $\mathbf{L} = \langle L(V_{a,1}), \ldots, L(V_{a,2^K}), L(V_{b,1}), \ldots, L(V_{b,2^K}) \rangle$, where $V_{a,1}, \ldots, V_{a,2^K}$ is a list (in some order, which is immaterial) of all the virtual nodes in the forward virtual half-network, excluding the virtual destination node $\langle \mathcal{B} \rangle$; $V_{b,1}, \ldots, V_{b,2^K}$ is a list of all the virtual nodes in the reverse virtual half-network, excluding the virtual destination node $\langle \mathcal{A} \rangle$; and $L(\cdot)$ denotes the queue length of the respective virtual node. The backpressure is then computed separately in the two virtual half-networks, as follows. In the forward half-network, the backpressure is

$$C_a^{\mathcal{I}} = T^{\mathcal{I}}(\langle \mathcal{B} \rangle) \cdot L(V_a) + \sum_{i=1}^{2^K} T^{\mathcal{I}}(V_{a,i}) \cdot [L(V_a) - L(V_{a,i})], \tag{5}$$

i.e., the expected value of the queue length difference between the virtual node from which the packet departs (V_a) and where it arrives (either $\langle \mathcal{B} \rangle$ or $V_{a,i}$). Similarly, the backpressure in the reverse half-network is

$$C_b^{\mathcal{I}} = T^{\mathcal{I}}(\langle \mathcal{A} \rangle) \cdot L(V_b) + \sum_{j=1}^{2^K} T^{\mathcal{I}}(V_{b,j}) \cdot [L(V_b) - L(V_{b,j})]. \tag{6}$$

In particular, notice that if action $\mathcal{I}$ does not involve a packet from the forward (or reverse) flow, then the associated backpressure in the respective virtual half-network is 0.

Finally, the backpressure-based scheduling policy is defined as follows. At each time slot,

- compute the backpressures (5)–(6) for every action $\mathcal{I}$ that can be validly applied with the current queue lengths $\mathbf{L}$ (i.e. satisfying the constraints of the definition of an action and not transmitting a packet from an empty queue);
- choose the action $\mathcal{I}$ that attains the highest *total backpressure* $\max\{C_a^{\mathcal{I}}, 0\} + \max\{C_b^{\mathcal{I}}, 0\}$.

Theorem 2. *The backpressure-based scheduling policy is throughput-optimal in the virtual network.*

Proof: The proof follows the same line as Theorem 2 of [24], as the policy satisfies the same necessary assumptions.

We omit the details due to space constraints. ■

V. The Epoch-Based Evacuation Policy

In this section, we discuss an alternative throughput-optimal policy, which is based on the idea of epoch-based operation and the optimal evacuation of packets within each epoch. Specifically, we consider the evolution of the system to be divided into a sequence of *epochs* as follows.

- A new epoch commences with the arrival of a new packet (or packets) to an empty system;
- An epoch ends as soon as all the packets that were present in the system at the start of the epoch leave the system;
- If there are packets in the system at the end of an epoch (which, by definition, must have arrived after the start of that epoch), a new epoch commences at that point.

A policy is called *epoch-based* if, within each epoch, it only processes the packets present at the start of the epoch, while ignoring all new packets arriving during the current epoch (these packets thus remain in the queues of the source nodes at the start of the next epoch).

Our interest in epoch-based policies is motivated by [22], where it was shown that, under very broad assumptions, an epoch-based policy achieves the maximal throughput region if it minimizes the expected evacuation time in each epoch individually. Consequently, for the rest of this section, we focus on a virtual network with a given vector $\mathbf{L}$ of instantaneous queue lengths, and describe how to find the fastest evacuation schedule that clears the system from that point assuming that the arrival process is "frozen" from that point, i.e. no further arrivals occur. To that end, for any particular policy, we define the *clearing time* of $\mathbf{L}$, denoted by $D(\mathbf{L})$, to be the expected number of slots until the system is cleared (i.e. all packets are successfully delivered to their respective destinations) under that policy. Thus, the clearing time of the zero vector, i.e. where all queues are empty, is 0. We now proceed to derive the clearing time expressions for an arbitrary $\mathbf{L}$.

First, suppose that for a particular $\mathbf{L}$, the policy chooses an action $\mathcal{I}$ that transmits a native packet from the forward flow only, taken from the queue of node V_a. The clearing time of $\mathbf{L}$ thus consists of 1 (counting the transmission in this slot) plus a weighted sum of the clearing times of all the possible queue length vectors to which the system can transition after this action. Therefore,

$$D(\mathbf{L}) = 1 + T^{\mathcal{I}}(\langle\mathcal{B}\rangle)\cdot D(\mathbf{L}-\mathbf{1}_{V_a}) + \sum_{i=1}^{2^K} T^{\mathcal{I}}(V_{a,i})\cdot D(\mathbf{L}-\mathbf{1}_{V_a}+\mathbf{1}_{V_{a,i}}), \quad (7)$$

where $\mathbf{1}_{V_a}$ and $\mathbf{1}_{V_{a,i}}$ are indicator vectors, i.e. vectors containing a 1 at the position corresponding to the respective virtual node and 0 elsewhere. Observe that one of the terms in the sum in the right-hand side of (7) corresponds to $\mathbf{L}$ itself (namely, the term where $V_{a,i} = V_a$); this corresponds to the outcome when there is no change in the network state, i.e. the transmitted packet fails to be overheard by any new physical node. Extracting it out of the sum, we get

$$D(\mathbf{L}) = \frac{1}{1-T^{\mathcal{I}}(V_a)} + \frac{T^{\mathcal{I}}(\langle\mathcal{B}\rangle)}{1-T^{\mathcal{I}}(V_a)}\cdot D(\mathbf{L}-\mathbf{1}_{V_a}) + \sum_{\substack{i=1\\ V_{a,i}\neq V_a}}^{2^K} \frac{T^{\mathcal{I}}(V_{a,i})}{1-T^{\mathcal{I}}(V_a)}\cdot D(\mathbf{L}-\mathbf{1}_{V_a}+\mathbf{1}_{V_{a,i}}). \quad (8)$$

Expression (8) can be intuitively interpreted as follows: the clearing time is the expected number of transmissions until a successful transition to any other vector occurs, followed by a weighted sum of the clearing times of new vectors, where the weights this time are the respective *conditional* transition probabilities given that a transition has occured. For an action involving a native packet from the reverse flow only, the clearing time expression is similar and omitted for brevity.

Now, consider the case where the chosen action for a vector $\mathbf{L}$ transmits a coded combination of two packets, taken from the virtual nodes V_a and V_b. The clearing time expression must now consider the possible transitions in both of the virtual half-networks; since the respective transition probabilities are independent, we obtain

$$\begin{aligned} D(\mathbf{L}) = {} & \frac{1}{1-T^{\mathcal{I}}(V_a)T^{\mathcal{I}}(V_b)} + \\ & \frac{T^{\mathcal{I}}(\langle\mathcal{B}\rangle)T^{\mathcal{I}}(\langle\mathcal{A}\rangle)}{1-T^{\mathcal{I}}(V_a)T^{\mathcal{I}}(V_b)}\cdot D(\mathbf{L}-\mathbf{1}_{V_a}-\mathbf{1}_{V_b}) + \\ & \sum_{i=1}^{2^K} \frac{T^{\mathcal{I}}(V_{a,i})T^{\mathcal{I}}(\langle\mathcal{A}\rangle)}{1-T^{\mathcal{I}}(V_a)T^{\mathcal{I}}(V_b)}\cdot D(\mathbf{L}-\mathbf{1}_{V_a}-\mathbf{1}_{V_b}+\mathbf{1}_{V_{a,i}}) + \\ & \sum_{j=1}^{2^K} \frac{T^{\mathcal{I}}(\langle\mathcal{B}\rangle)T^{\mathcal{I}}(V_{b,j})}{1-T^{\mathcal{I}}(V_a)T^{\mathcal{I}}(V_b)}\cdot D(\mathbf{L}-\mathbf{1}_{V_a}-\mathbf{1}_{V_b}+\mathbf{1}_{V_{b,j}}) + \\ & \underset{(V_{a,i},V_{b,j})\neq(V_a,V_b)}{\sum_{i=1}^{2^K}\sum_{j=1}^{2^K}} \frac{T^{\mathcal{I}}(V_{a,i})T^{\mathcal{I}}(V_{b,j})}{1-T^{\mathcal{I}}(V_a)T^{\mathcal{I}}(V_b)}\cdot \\ & D(\mathbf{L}-\mathbf{1}_{V_a}-\mathbf{1}_{V_b}+\mathbf{1}_{V_{a,i}}+\mathbf{1}_{V_{b,j}}). \end{aligned} \quad (9)$$

For two queue length vectors $\mathbf{L},\mathbf{L}'$, we say that $\mathbf{L}$ is *dependent* on $\mathbf{L}'$ if there exists any action $\mathcal{I}$ such that $D(\mathbf{L}')$ appears in the right-hand side of the corresponding expression (8)–(9) with a non-zero transition probability coefficient. Thus, every dependency relationship can be associated with a transition of native packets between virtual nodes (in either one or both of the half-networks). Therefore, the dependency relationship is acyclical; a directed cycle of dependency relationships is not possible since a transition from a virtual node $\langle A,S\rangle$ can happen only to either $\langle\mathcal{B}\rangle$ or $\langle A,S'\rangle$ where S' is a strict superset of S (and similarly for transitions from a virtual node $\langle B,S\rangle$).

We therefore conclude that the optimal clearing time for a given vector $\mathbf{L}$ can be found by a dynamic programming algorithm, as follows. Assuming that the optimal clearing times have already been found for all the vectors $\mathbf{L}$ depends on, the algorithm tries every action $\mathcal{I}$ that can be validly applied with $\mathbf{L}$ (i.e. satisfying the constraints of the definition of an action and not transmitting a packet from an empty queue), calculates the resulting $D(\mathbf{L})$ using the corresponding expression (8)–(9), and then records the minimum $D(\mathbf{L})$ as well as the corresponding action. When used in the context of an epoch-based policy — to find the optimal evacuation policy for an initial vector of

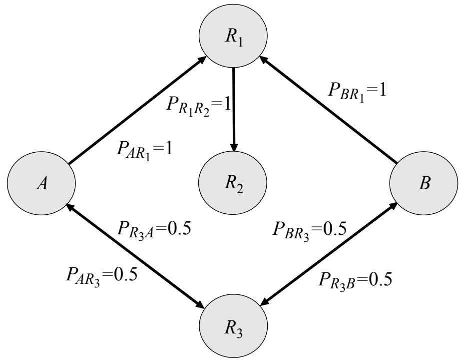


Fig. 3: Example of backpressure inefficiency at low arrival rates.

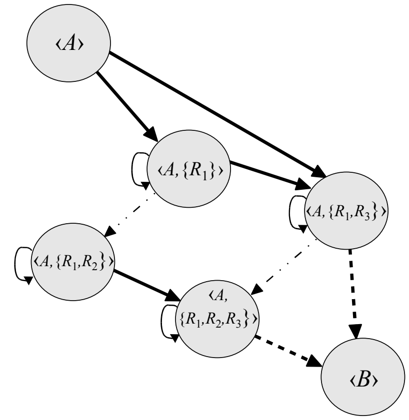


Fig. 4: The virtual network corresponding to Figure 3.

M_A and M_B forward and reverse packets, respectively, queued at the source nodes at the start of the epoch — this dynamic programming algorithm finds the optimal actions for all queue vectors with a total size of up to M_A and M_B.‡ Due to space constraints, we omit a detailed step-by-step description of the algorithm (which is similar to, e.g., the well-known Bellman-Ford algorithm for shortest paths in graphs).

VI. Discussion

Having described two different throughput-optimal policies for wireless networks with opportunistic routing, we now discuss their relative merits and provide some insights for deciding when it is better to apply one or the other.

It is generally known that, in the traditional model of networks with constrained actions, the backpressure policy — despite being optimal in a throughput sense — can be suboptimal with respect to other metrics, e.g. delay and/or number of transmissions per packet, when the arrival rate is low and not close to the capacity limit [15]. We now present an example which shows that the same phenomenon exists in the context of virtual networks as well. Consider the network in Figure 3, featuring a path of high-quality links from each source to R_1 and then to R_2 which is a "dead end" (no outgoing links from R_2, i.e. zero probability of overhearing its transmissions). The only path from A to B with a nonzero probability is $A \to R_3 \to B$ (and, similarly, in the opposite direction); hence, clearly, the capacity limit for the corresponding unidirectional

‡This algorithm is an extension of the one presented in [25], which was designed to optimize the clearing time for exactly one packet in each direction.

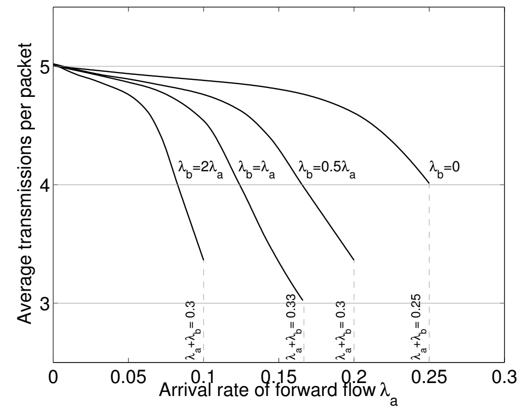


Fig. 5: Average number of transmissions for the backpressure policy.

flow is the reciprocal of the path's Expected Transmission Count (ETX) metric (in this case, $\frac{1}{\frac{1}{0.5}+\frac{1}{0.5}} = \frac{1}{4}$ packets per slot). For a bidirectional flow, note that R_3 is able to clear $0.25\cdot 2+0.5\cdot 1=1$ packet per slot on average with the help of network coding, as long as it has packets from both directions in the queue; thus, assuming a symmetric arrival rate to both sources, the throughput limit is $\frac{1}{\frac{2}{0.5}+2\cdot 1} = \frac{1}{6}$ packet pairs per slot, or equivalently $\frac{1}{3}$ packets per slot in total.

The corresponding virtual half-network from A to B for this example is depicted in Figure 4 (the reverse half-network is identical). Assume that a single packet arrives at A, and after the first transmission, finds itself in either the virtual node $\langle A,\{R_1\}\rangle$ or $\langle A,\{R_1,R_3\}\rangle$ (each with probability 0.5). In the former case, it is easily seen that the action achieving the highest backpressure value (which, for a single packet, is simply equal to the total outgoing transition probability to any other node) is a transmission by R_1, which has a transition probability of 1 to the node $\langle A,\{R_1,R_2\}\rangle$, better than a retransmission by A which only has a transition probability of 0.5 (to the node $\langle A,\{R_1,R_3\}\rangle$). Alternatively, if the packet arrived to the virtual node $\langle A,\{R_1,R_3\}\rangle$ after the initial transmission, then the highest backpressure value in the next slot is again achieved by a transmission by R_1 (transition probability 1 to the node $\langle A,\{R_1,R_2,R_3\}\rangle$), rather than a transmission by R_3 (probability of only 0.5 to the destination $\langle\mathcal{B}\rangle$).

Thus, we conclude that, as long as only one packet is present in the network, in either direction (i.e. arrivals occur at a sufficiently low rate), a wasted transmission by R_1 always takes place, which serves no other purpose than to transmit the packet to the "dead end" R_2, resulting in suboptimal delay and energy consumption. Figure 5 shows the average number of transmissions per packet for the backpressure policy as a function of the arrival rate, assuming that the arrival processes at both sources are independent and Poisson. We observe a gradual improvement as the arrival rate approaches the capacity limit, since the growing queue lengths at the "dead end" nodes $\langle A,\{R_1,R_2\}\rangle$ and $\langle A,\{R_1,R_2,R_3\}\rangle$ negate the wasted transmissions by R_1. The figure also shows clearly the benefit of network coding, with the lowest average transmissions and highest total throughput limit (i.e., $\lambda_a+\lambda_b$) attained when the packet arrival rates are equal on both sides.

Obviously, this inefficiency does not occur with the epoch-based policy from section V, which does not myopically maximize the probability of a packet to advance a step in the virtual network, but rather "looks ahead" to find the fastest evacuation path in each epoch and avoids dead ends in the process, leading to superior delay and energy performance. However, this advantage comes at a cost of higher computational complexity; while the backpressure policy only requires a calculation of backpressure values, which is linear in the number of possible actions (i.e., roughly, the size of the virtual network), computing the fastest evacuation policy takes polynomial time in the number of packets at the start of the epoch, with the degree of the polynomial determined by the size of the network. This suggests the possibility of a "best-of-both" combination that can be achieved by a *hybrid* policy, which follows the fastest-evacuation policy as long as the network is lightly loaded (i.e. the number of packets is small), and falls back to the backpressure one when the load rises above some threshold. The exploration of such hybrid policies and the best way to set the relevant threshold is left as a direction for future work.

VII. Conclusion

In this paper, we considered the throughput region in wireless networks that employ *opportunistic routing*, where any packet can be overheard by multiple nodes and forwarded towards the destination by any of them based on instantaneous channel conditions, rather than via a single predetermined route. Traditional models from the theory of network capacity cannot be directly applied in this context due to the fact that, with opportunistic routing, multiple copies of the same packet can be enqueued in multiple nodes, and disappear from all the nodes as soon as any of the copies is delivered to the destination. Accordingly, we have described a generic transformation that works around the above limitation, using a *virtual network* in which nodes correspond to subsets of physical nodes that may overhear a packet, and the transition of a packet between virtual nodes is probabilistic. We have established the equivalence of the throughput region in the physical and virtual network and described two policies that attain the maximal throughput, one based on a variation of the classical backpressure algorithm, and another based on minimum-time evacuation of packets in a sequence of epochs. The virtual network transformation technique and the associated throughput-optimal policies were demonstrated for either a unidirectional or bidirectional flow between a pair of endpoints, in the latter case involving network coding between packets traveling in opposite directions.

The throughput regions considered in this study are potentially limited by the requirement that every transmission of a coded packet (e.g. a XOR combination of packets from opposite directions) must be able to be decoded immediately by the next hop. It remains an open question whether policies that allow coded packets to be retained by nodes that are unable to decode them into individual native packets, and later forwarded or even combined further with additional packets, can attain a higher throughput region. The resolution of this question, as well as a generalization of the model to multiple simultaneous flows between an arbitrary number of source-destination pairs, are left for future work.

References

[1] T. Clausen and P. Jacquet. Optimized link state routing protocol (OLSR). RFC 3626, October 2003.

[2] K.N. Ramachandaran, E.M. Belding, K.C. Almeroth, and M.M. Buddhikot. Interference-aware channel assignment in multi-radio wireless mesh networks. In *Proc. IEEE INFOCOM*, Barcelona, Spain, April 2006.

[3] P. Larsson. Selection diversity forwarding in a multihop packet radio network with fading channel and capture. *ACM Mobile Computing and Communications Review*, 5(4):47–54, October 2001.

[4] M. Zorzi and R.R. Rao. Geographic random forwarding (GeRaF) for ad hoc and sensor networks: Energy and latency performance. *IEEE Trans. on Mobile Computing*, 2(4):349–365, October 2003.

[5] S. Biswas and R. Morris. ExOR: Opportunistic multi-hop routing for wireless networks. In *Proc. ACM SIGCOMM*, pages 133–144, Philadelphia, PA, August 2005.

[6] M. Afanasyev and A.C. Snoeren. The importance of being overheard: throughput gains in wireless mesh networks. In *Proc. ACM Internet Measurement Conference (IMC)*, Chicago, IL, November 2009.

[7] A. Basalamah, S.M. Kim, S. Guo, T. He, and Y. Tobe. Link correlation aware opportunistic routing. In *Proc. IEEE INFOCOM mini-conference*, Orlando, FL, March 2012.

[8] S. Chachulski, M. Jennings, S. Katti, and D. Katabi. Trading structure for randomness in wireless opportunistic routing. In *Proc. ACM SIGCOMM*, Kyoto, Japan, August 2007.

[9] Y. Lin, B. Li, and B. Liang. CodeOR: Opportunistic routing in wireless mesh networks with segmented network coding. In *Proc. IEEE ICNP*, Orlando, FL, October 2008.

[10] Y.Y. Lin, C.C. Huang, and J.L. Huang. PipelineOR: A pipelined opportunistic routing protocol with network coding in wireless mesh networks. In *Proc. IEEE Vehicular Technology Conference (VTC)*, Taipei, Taiwan, May 2010.

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

[12] T. Cui, L. Chen, and T. Ho. Energy efficient opportunistic network coding for wireless networks. In *Proc. IEEE INFOCOM*, Phoenix, AZ, April 2008.

[13] I. Broustis, G.S. Paschos, D. Syrivelis, L. Georgiadis, and L. Tassiulas. NCRAWL: Network coding for rate adaptive wireless links. arXiv:1104.0645, April 2011.

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

[15] L. Georgiadis, M.J. Neely, and L. Tassiulas. Resource allocation and cross-layer control in wireless networks. *Foundations and Trends in Networking*, 1(1):1–147, 2006.

[16] M.J. Neely and R. Urgaonkar. Optimal backpressure routing for wireless networks with multi-receiver diversity. *Ad Hoc Networks*, 7(5):862–881, July 2009.

[17] A. Eryilmaz, D.S. Lun, and B.T. Swapna. Control of multi-hop communication networks for inter-session network coding. *IEEE Trans. on Information Theory*, 57(2):1092–1110, February 2011.

[18] Y.E. Sagduyu and A. Ephremides. On broadcast stability of queue-based dynamic network coding over erasure channels. *IEEE Trans. on Information Theory*, 55(12):5463–5478, December 2009.

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

[20] A.F. Dana, R. Gowaikar, R. Palanki, B. Hassibi, and M. Effros. Capacity of wireless erasure networks. *IEEE Trans. on Information Theory*, 52(3):789–804, March 2006.

[21] H.-C. Lu and W. Liao. Cooperative multicasting in network-coding enabled multi-rate wireless relay networks. In *Proc. IEEE INFOCOM*, Orlando, FL, March 2012.

[22] L. Georgiadis, G.S. Paschos, L. Tassiulas, and L. Libman. Minimal evacuation times and stability. arXiv:1211.4660, November 2012.

[23] M. Bramson. *Stability of Queueing Networks*, volume 1950 of *Lecture Notes in Mathematics*. Springer, 2008.

[24] G.S. Paschos, L. Georgiadis, and L. Tassiulas. Scheduling with pairwise xoring of packets under statistical overhearing information and feedback. *Queueing Systems*, April 2012.

[25] T. Mehmood and L. Libman. Towards optimal forwarding in wireless networks: Opportunistic routing meets network coding. In *Proc. IEEE LCN*, Zurich, Switzerland, October 2009.