Toward Practical Opportunistic Routing with Intra-session Network Coding for Mesh Networks

Božidar Radunović[1], Christos Gkantsidis[1], Peter Key[1], Pablo Rodriguez[2]

[1] Microsoft Research
Cambridge, UK
{bozidar,chrisgk,peter.key}@microsoft.com

[2] Telefonica Research
Barcelona, Spain
pablorr@tid.es

***Abstract*—We consider opportunistic routing in wireless mesh networks. We exploit the inherent diversity of the broadcast nature of wireless by making use of multi-path routing. We present a novel optimization framework for opportunistic routing based on network utility maximization (NUM) that enables us to derive optimal flow control, routing, scheduling, and rate adaptation schemes, where we use network coding to ease the routing problem. All previous work on NUM assumed unicast transmissions; however, the wireless medium is by its nature broadcast and a transmission will be received by multiple nodes. The structure of our design is fundamentally different; this is due to the fact that our link rate constraints are defined per broadcast region instead of links in isolation. We prove optimality and derive a primal-dual algorithm that lays the basis for a practical protocol. Optimal MAC scheduling is difficult to implement, and we use 802.11-like random scheduling rather than optimal in our comparisons. Under random scheduling, our protocol becomes fully decentralized (we assume ideal signaling). The use of network coding introduces additional constraints on scheduling, and we propose a novel scheme to avoid starvation. We simulate realistic topologies and show that we can achieve 20-200% throughput improvement compared to single path routing, and several times compared to a recent related opportunistic protocol (MORE).**

***Index Terms*—wireless mesh networks, network coding, opportunistic routing, broadcast, multi-path routing, flow control, fairness, rate adaptation**

I. Introduction

One of the main challenges in building wireless mesh networks ([1], [2], [3]) is to guarantee high performance despite the unpredictable and highly-variable nature of the wireless channel. In fact the use of wireless channels presents some unique opportunities that can be exploited to improve the performance. For example, the broadcast nature of the medium can be used to provide opportunistic transmissions as suggested in [4]. In addition, in wireless mesh networks, there are typically multiple paths connecting each source destination pair, hence using some of these paths in parallel can improve performance [5], [6].

However most of the existing work on optimal wireless protocol design (c.f. [7]) ignores the broadcast nature of the channel. Instead, a transmitter selects *a priori* the next-hop for a packet, and if the selected next-hop has not received the packet, the packet is retransmitted (even though another next-hop neighbor may have received it correctly). The routing is not opportunistic (as in [4]) and the diversity of the broadcast medium is ignored.

The main focus of this paper is the optimal use of both multiple paths and opportunistic transmission. We use intra-session network coding [8] to simplify the problem of scheduling packet transmissions across multiple paths, as others have done to [5], [6], [9]. We propose a network optimization framework that optimizes the rate of packet transmissions between source and destination pairs.

In order to use the resources of a wireless mesh network efficiently, the system needs to take into account: (a) the existence of multiple paths, (b) the unreliable nature of wireless links, (c) the existence of multiple transmission powers and rates (which in turn affects the probability of correct packet reception), (d) the broadcast nature of the channel, (e) competition among many flows, (f) fairness and efficiency. Observe that optimizing across all these parameters implies optimizing across multiple layers of the networking stack; for example, the choice of transmission power and rate is typically done at the physical layer, whereas coordination among different flows is typically done at the network layer. As we shall see, it is important to perform such cross-layer optimizations to achieve optimal performance.

We use an optimization framework to design a distributed maximization algorithm. We account for transport layer controls and address questions of fairness by maximizing the aggregate utility of the end-to-end flows, where we associate a utility function $U(\cdot)$ with a flow. Because we use network coding, our optimization leverages existing theory [10], [9]. Our algorithm is a primal-dual algorithm [11]. The primal formulation expresses the optimization problem as a function of the rates of the various flows in the network; the dual formulation uses as variables the queue lengths (per flow and per node). The main advantage of using the dual formulation of the optimization problem is that the dual variables (also referred as shadow prices) relate to queue lengths and can be directly used by back-pressure algorithms for flow control [12], [7]. As a simple example, a large number of queued packets for a particular flow at an internal node can be interpreted that the path going through that node is congested and should be avoided. The main advantage of using the primal-dual formulation is that it adapts the primal variables (i.e. flow rates) more slowly, hence, allows TCP-like window-based rate control modeling (as originally mentioned by Erylimaz et al. [12]). We propose a novel algorithm for cross-layer optimization and prove, using Lyapunov functions, that it converges to the optimal rate allocation.

Despite using similar optimization techniques to prior work (e.g. [12], [7], [13], [14], [11]), the solution to our problem is very different. We define rate constraint for each set of broadcast receivers. Consequently, dual variables are related to these broadcast sets, and allow us to adjust the level of opportunism as a function of a congestion in the rest of the network.

The proposed optimization framework is difficult to implement; indeed, the joint scheduling, rate and power control problem is NP-hard [15]. Additionally, current wireless MAC protocols use uncontrolled randomized channel scheduling. We propose a distributed heuristic based on the optimal algorithm. We show that, even in the absence of optimal channel scheduling, the other aspects of the optimization problem (i.e. flow selection and transmission rate selection) still give performance advantages. Hence, our heuristic hints toward an implementation in practical systems. The fundamental idea behind our algorithm (and, of its distributed implementation) is to assign *credits* to nodes, transfer credits between nodes, and schedule on the basis of credits (see Sec. III for more details).

The main contributions of our paper are as follows:

• We propose a network wide optimization algorithm that maximizes rate-based global network performance, and extends previous work by incorporating broadcast/opportunistic routing, multi-path routing, and fairness/rate control (Sections II and III). We introduce a notion of virtual packets, called credits, that enable us to decouple routing and flow control from actual packet transmissions and delivery. We prove the optimality of the algorithm.

• Based on the optimization algorithm, we give a distributed implementation (assuming ideal signaling) of routing, rate adaptation, and flow control for networks with random scheduling (Section III-C) that outperforms existing algorithms. We prove that our algorithms extends and outperforms a recent proposal, MORE [5]. The distributed algorithm can be used with the current 802.11 MAC, and indeed is MAC independent.

• Practical network coding schemes use finite generation sizes. We show that a naive approach for scheduling generations may lead to starvation. We propose a novel heuristic and we demonstrate that it circumvents network starvation (Section IV-A).

• We demonstrate that rate selection is important for optimizing performance in 802.11a networks (Section IV-B). We confirm the findings from [5] that such optimizations are not necessary for 802.11b networks.

• Using simulation on realistic topologies, we show we can achieve 20-100% throughput improvement with our distributed implementation compared to single path routing, and 20-300% compared to MORE [5] (Section V).[1]

The rest of the paper is organized as follows. Section II describes the model we are using. Section III gives the optimization problem, describes an approximation of the problem that can be computed in a distributed system, and compares with a recent proposal for multi-path routing. Section IV discusses some practical issues, namely the effect of limited generation sizes, and the effect of randomized 802.11-compatible channel scheduling. Section V evaluates the performance of our system using simulation. Section VI provides related work.

[1]Observe that MORE optimizes the number of delivered packets for flows in isolation, and,when multiple flows are active, may perform worse than single path routing w.r.t. rates.

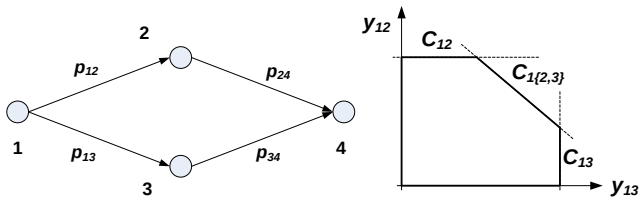


Fig. 1. A network with 4 nodes is shown on the left. For example, an activation profile $\{(1,2),(3,4)\}$ depicts a profile where nodes 1 and 3 are transmitting, node 2 is receiving a packet from node 1, while node 4 is receiving from node 3. Profile $\{(1,\{2,3\})\}$ depicts node 1 transmitting and nodes 2 and 3 receiving (the same) packet from node 1. The feasible rate region for (y_{12}, y_{13}) is given on the right, described by inequalities: $\sum_{j\in J} y_{1j} \leq C_{1J}$ for all $J \subseteq \{2,3\}$.

II. Model

In this section we introduce our notation, denoting vectors in bold typeface. We extend the model of a wireless erasure network developed in [16] to include multiple flows.

A. PHY And MAC Characteristics

We consider a network comprising of a set of nodes $\mathcal{N}$, $N = |\mathcal{N}|$. Whenever a node transmits a packet, several nodes may receive it. We model packet transmission from node i to a set of nodes $K_i \subseteq \mathcal{N}$ with a hyperarc (i, K_i). We define an activation profile $S = \{S_l\}$ to be a set of hyperarcs active at the same time. There may be several constraints on feasible activation profiles. For example, a node may be limited to receive from but one node, or transmit to only one node at a time. The only condition we shall impose is that a node can be the *source* of only one hyperarc in one activation profile. All the other constraints can be expressed through reception probabilities and our model is general enough to incorporate them (in particular, it is possible that a node transmits while some information is being sent to it, in which case we shall set the probability of successful reception to 0; see below for details). We denote by $\mathcal{S}$ the set of feasible activation profiles and let $\mathrm{SRC}(S) = \{i \in \mathcal{N} \,|\, \exists K_i \subseteq \mathcal{N}, (i, K_i) \in S\}$ be the set of transmitters in activation profile S.

Each transmission has two associated parameters, power $P \in \mathcal{P}$ and rate $R \in \mathcal{R}$, where $\mathcal{P}$ is the set of allowed transmission powers (e.g. $\mathcal{P} \in [0, P^M]$, where P^M is given by regulations) and $\mathcal{R}$ is sets of available PHY transmission rates, defined by supported spreading, coding, and modulations.

Consider an activation profile S in which node i transmits to set of nodes K_i, and suppose node i is transmitting with power P_i and rate R_i. We can associate power vector $\mathbf{P} = (P_i)_{i\in\mathcal{N}}$ rate vector $R = (R_i)_{i\in\mathcal{N}}$ to these transmissions. Let $T_{ij}(\mathbf{P}, R_i, S)$ be an indicator of a random event such that $T_{ij}(\mathbf{P}, R_i, S) = 1$ if a packet is successfully delivered from

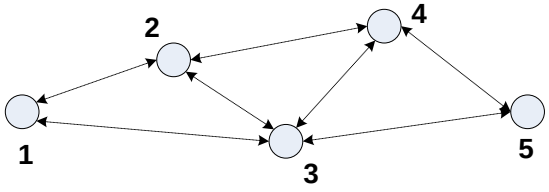


Fig. 2. Forwarding example. Lines connect nodes that can exchange packets. Opportunistic routing implies multiple-paths. Possible paths from 1 to 5 are for example $(1,2,3,4,5),(1,2,3,5),(1,2,4,5),(1,3,4,5),(1,3,5)$. Path $(1,2,1,3,5)$ is also possible but will be typically eliminated by a routing protocol due to suboptimality.

i to $j \in K_i$. It depends on the packet transmission power P_i and rate R_i, as well as on the interference from concurrent transmissions described through $\mathbf{P}$ and S. We also denote by

$$T_{iJ}(\mathbf{P}, R_i, S) = 1 - \prod_{j \in J}(1 - T_{ij}(\mathbf{P}, R_i, S))$$

the indicator of the event that *at least one of* the nodes from J, ($J \cap K_i \neq \emptyset$) receives a packet. Consequently

$$p_{iJ}(\mathbf{P}, R_i, S) = \mathrm{Prob}(T_{iJ} = 1).$$

By convention, we assume $p_{iJ}(\mathbf{P}, R_i, S) = 0$ if $J \cap K_i = \emptyset$, for $(i, K_i) \in S$. Our model does not require any assumptions on channel conditions; in particular, events T_{ij} and T_{ik} are not assumed independent.

We can now calculate $C_{iJ}(\mathbf{P}, R_i, S)$, the average number of packets per unit time conveyed from node i to any of the nodes in $J \subseteq \mathcal{N}$. We have

$$C_{iJ}(\mathbf{P}, R_i, S) = R_i\, p_{iJ}(\mathbf{P}, R_i, S) \tag{1}$$

B. Traffic, Forwarding And Flow Scheduling

There is a set of unicast end-to-end flows $\mathcal{C}$ in the network, and each flow $c \in \mathcal{C}$ has a source and a destination node $\mathrm{Src}(c), \mathrm{Dst}(c) \in \mathcal{N}$ respectively. We denote by f_c the rate of flow c.

Opportunistic routing does not *a priori* rely on a notion of a path or a route. Consider the example of Figure 2: a packet going from node 1 to node 5 may be relayed by any of the nodes 2, 3, 4, depending on which node happens to hear it. Thus implicitly, opportunistic routing implies multi-path routing *without* pre-specifying the path. Formally, a packet from node i can reach node j if there exists an activation profile that allows such a transmission (there exists $S \in \mathcal{S}$ such that $(i, K_i) \in S$ and $j \in K_i$). The goal of our optimization framework is to derive how many packets should be forwarded by each of the nodes.

In practice, an external routing protocol can be combined with the opportunistic approach to further improve efficiency. Consider again the example from Figure 2: if a packet is broadcast by node 2 destined for 5, it may be received and potentially forwarded by node 1. Our opportunistic routing protocol will, as described latter, eliminate such events, but will take some time to discover them. Instead, one may use a readily available external routing protocol to promptly eliminate obviously suboptimal paths (e.g. path $(1,2,1,3,5)$ from Figure 2). Our model can be easily extended to include constraints imposed by an external routing protocol, but we omit such extensions to simplify the exposition. We do constrain the set of available paths in the numerical results, as explained in Section V.

A node that transmits a packet cannot control who will receive the transmitted packets due to channel randomness. A node that receives a packet has to decide whether it will forward it or not. This decision is made through credit assignment, described in Section II-D.

Whenever a node is about to transmit, it needs to decide which flow it will transmit. This is defined through a flow-scheduling profile matrix $\mathbf{A}$. If node i transmits a packet from flow c we set $A_{ic} = 1$, otherwise $A_{ic} = 0$. We say that a flow scheduling profile is valid if for each $i \in \mathcal{N}$ there exists only one $c \in \mathcal{C}$ such that $A_{ic} = 1$. $\mathcal{A}$ denotes the set of all valid flow scheduling profiles.

To illustrate the use of flow scheduling profile, consider the example in Figure 1 having two flows $\mathcal{C} = \{1, 2\}$, both from 1 to 4. The number of packets for flow c sent by node 1 and received by node 2 depends not only on how often $(1, 2)$ is scheduled, but also on how often $(1, \{2, 3\})$ is scheduled to transmit a packet from c. This is why the flow scheduling decision is assigned to a node instead of a link, which is in sharp contrast to [17], [18], [12].

C. Network Coding

We assume network coding per flow is used [16], [5]. The main benefit of network coding is that it facilitates scheduling. If the same packet is received by several nodes, a mechanism is needed to prevent two or more nodes forwarding the same packets [4]. To eliminate this problem, each relay forwards a random linear combination of all previously received packets from the same flow. It has been show in [9] that the random linear combinations received at the destination will be independent with high probability, hence the packets can be restored.

Ideally, network coding should be performed across the entire flow. However, this is not practical. Instead, packets are divided in **generations** and only packets from the same generation are combined. For more details see [16], [5]. In Section III we analyze the optimal network design assuming very large generation sizes (as in [16]); we address finite generation sizes in Section IV.

D. Credits

As remarked on earlier, whenever a packet is transmitted, it may be received by several nodes, and it is important to decide which should forward packets, to avoid redundant transmissions (as explained in [19], [5]).

We introduce the concept of credits, which is similar to the control decision variable of Neely [19]. One credit is created for each packet at the source node. Credits are identified with a generation, not a specific packet. They are conserved until they arrive at the flow's destination. In this way we guarantee that the destination will receive as many linear combinations of

the packets as the number of packets generated at the source, and hence will be able to decode the packets.

Credits are interpreted as the number of packets of a specific flow to be forwarded by a node. By controlling the rate of credits we control the rate of packets forwarded by next-hop relays. Consider again the example from Figure 1. Node 1 should adapt the rate of packets forwarded through 2 and 3 not only as a function of link qualities p_{12} and p_{13}, but also as a function of p_{24} and p_{34}, the quality of paths from 2 and 3 to the destination 4. For example, if $p_{24} \ll p_{34}$ then node 2 should not forward any packet, regardless of how many it has received. Node 1 cannot control what nodes 2 and 3 receive due to randomness of the channel. Instead, node 1 sends a credit to node 2 (or node 3) whenever it wants node 2 (or node 3) to forward a packet.

The main advantage of the credit scheme is that it simplifies scheduling. Credits are declarations of intent. The actual packet transmissions may occur at arbitrary time instants. Due to the use of network coding, we only need to ensure that the total number of packets per generation transmitted between each two nodes corresponds to the number of credits. Thus, scheduling is done at a generation level and not at the packet level, incuring significantly smaller overhead (especially when the generation size is large).

In practice, credits can be piggybacked with packet transmissions. The receiving node only updates its credits when a successful packet transmission actually occurs. In this work we assume there is an ideal (no loss and no delay) signaling plane that transmits credits and feedbacks.

As each credit delegates one packet to a node, we may express all the rates in the system in terms of credits. For example, y_{ij}^c is the rate of credits of flow c passed from node i to node j, and it equals the rate of innovative linear combinations of packets of flow c delivered from i to j. Theorem 1 shows that the rate of independent packets received at a destination of each flow will correspond to the number of credits delivered, when the generation size is large.

E. Dynamics And Stability

We further assume the system is slotted in time. In each slot $t = 0, 1, \ldots$ a medium access protocol assigns an activation profile $S(t)$ and a flow-scheduling profile $A(t)$, and to each transmitter $i \in \mathrm{SRC}(S(t))$ we assign transmit power $P_i(t)$ and rate $R_i(t)$. Let $y_{ij}^c(t)$ be the number of credits for flow c transmitted from node i to node j during slot t, and let $x_{iJ}^c(t)$ denote the number of packets of flow c actually delivered from i to any of the nodes in J during slot t (as if all nodes in J are grouped as a single receiver). Let $f_c(t)$ be the number of fresh packets/credits generated at the source of flow c.

Note that, because each successful packet delivery is always associated with a credit transmission, we look at credit queues. Let $q_i^c(t)$ be the amount of credits of flow c queued at node i. The system is stable if every queue size is bounded. We define stability more formally in Section III-D.

F. Constraints Of The Model And Possible Generalizations

We make several simplifying assumptions to make the analysis tractable. Firstly, we assume that the system is slotted. All operations within a slot occur concurrently and instantly. Secondly, we assume the perfect signaling. There is no loss or delay in signaling messages (credit transfers and acknowledgments).

Our results can be extended to consider imperfect signaling and arbitrary but limited delays in the system (see for example [20, Part 2] on a discussion how does a delayed feedback affect the speed of convergence). Also, see [21] for a practical implementation of a back-pressure based system and its interaction with TCP.

III. Optimal Flow Control For Fairness

In this Section we introduce the optimization problem (Sec. III-B), propose an algorithm for solving it (Sec. III-C), and prove that the algorithm converges (Sec. III-D). Sec.III-A introduces some further notation that is needed for the description of the optimization problem. Finally, in Sec. III-E we compare our algorithm with the MORE algorithm proposed in [5].

A. Feasible Average Rate Set

In this section we define a set of constraints on average rates in the system. Assume an assignment of average end-to-end rates f_c, for each flow c, and denote the average rate vector by $\mathbf{f} = (f_c)_{c \in \mathcal{C}}$. The vector of rates is valid under the following three conditions. First, traffic at node i is stable if the total ingress traffic is smaller than the total egress traffic, which we write as

$$\sum_{j \neq i} y_{ji}^c + f_c 1_{\{i = \mathrm{Src}(c)\}} \leq \sum_{j \neq i} y_{ij}^c, \tag{2}$$

for all $i \neq \mathrm{Dst}(c)$, where $1_x = 1$ if x is true, 0 otherwise, and where $y_{ij} \geq 0$.

Second, traffic at each broadcast region is stable if we do not receive more credits than we can actually forward (see also Fig. 1):

$$\sum_{j \in J} y_{ij}^c \leq x_{iJ}^c, \quad \text{for all } J \in \mathcal{N}. \tag{3}$$

Recall that y_{ij}^c is the average number of credits of flow c node i *assigns* to node j and x_{iJ}^c is the average number of packets of flow c actually *delivered* from i to any of the nodes in J.

Finally, we define scheduling constraints. A schedule is a sequence $\{(S(t), \mathbf{R}(t), \mathbf{P}(t), \mathbf{A}(t))\}_{t \geq 0}$ which defines scheduling profile $S(t)$, routing profile $\mathbf{A}(t)$ and power and rate allocations $\mathbf{R}(t), \mathbf{P}(t)$ in each slot $t \geq 0$. Since we are interested in long-term average rates, we define

$$\alpha_{S,\mathbf{R},\mathbf{P},\mathbf{A}} = \lim_{T \to \infty} \frac{1}{T} \sum_{t \leq T} 1_{\{S(t)=S, \mathbf{R}(t)=\mathbf{R}, \mathbf{P}(t)=\mathbf{P}, \mathbf{A}(t)=\mathbf{A}\}}$$

to be the fraction of time the network uses scheduling profile S, routing profile $\mathbf{A}$ and power and rate allocations $\mathbf{R}, \mathbf{P}$. By definition, $\alpha_{S,\mathbf{R},\mathbf{P},\mathbf{A}} \geq 0$ and $\sum_{S,\mathbf{R},\mathbf{P},\mathbf{A}} \alpha_{S,\mathbf{R},\mathbf{P},\mathbf{A}} \leq 1$. The schedule defined by $\{\alpha_{S,\mathbf{R},\mathbf{P},\mathbf{A}}\}_{S,\mathbf{R},\mathbf{P},\mathbf{A}}$ is stable if it can support broadcast traffic $\{x_{iJ}^c\}_{i,J,c}$, and we write the scheduling constraints as

$$x_{iJ}^c \leq \sum_{S,\mathbf{A},\mathbf{R},\mathbf{P}} \alpha_{S,\mathbf{R},\mathbf{P},\mathbf{A}} A_{ic} C_{iJ}(\mathbf{P}, R_i, S) \tag{4}$$

We use the following characterization of feasible rates from [9]:

Definition 1: Vector $\mathbf{f}$ is said to be *feasible* if each flow c can transport information from $\mathrm{Src}(c)$ to $\mathrm{Dst}(c)$ at average rate f_c.

Theorem 1: Let $\mathcal{F}$ be the set of average end-to-end rate vector $\mathbf{f} = (f_c)_{c\in\mathcal{C}}$ such that there exists vectors $\mathbf{y} = (y_{ij}^c)_{i,j\in\mathcal{N},c\in\mathcal{C}}$, $\mathbf{x} = (x_{iJ}^c)_{i\in\mathcal{N},J\subseteq\mathcal{N},c\in\mathcal{C}}$, and $\boldsymbol{\alpha} = (\alpha_{S,\mathbf{R},\mathbf{P},\mathbf{A}})_{S\in\mathcal{S},\mathbf{R}\in\mathcal{R}^N,\mathbf{P}\in\mathcal{P}^N,\mathbf{A}\in\mathcal{A}}$ that satisfy (2), (3), and (4) subject to $\alpha_{S,\mathbf{R},\mathbf{P},\mathbf{A}} \geq 0$ and $\sum_{S,\mathbf{R},\mathbf{P},\mathbf{A}} \alpha_{S,\mathbf{R},\mathbf{P},\mathbf{A}} \leq 1$. The vector $\mathbf{f}$ is feasible when coding generation size goes to infinity if and only if it belongs to $\mathcal{F}$. Moreover, the set of feasible end-to-end rates $\mathcal{F}$ is convex.

Proof: Proof of feasibility follows directly from [9]. The set is convex since all constraints are convex. ∎

B. Utility Maximization

For each flow $c \in \mathcal{C}$ we define a utility function $U_c(\cdot)$ to be a strictly concave, increasing function of end-to-end flow rate f_c. The utility of flow c is then $U_c(f_c)$. For example, $U_c(f_c) = \log(f_c)$ represents proportional fairness [22] and $U_c(f_c) \propto -1/f_c$ approximates TCP's utility [11]. The goal of utility maximization is to achieve a trade-off between efficiency and fairness. Proportional fairness is an example of such an approach [22].

We can write the network-wide optimization problem as

$$\begin{array}{ll} \max & \sum_{c\in\mathcal{C}} U_c(f_c) \\ s.t. & \mathbf{f} \in \mathcal{F}. \end{array} \tag{5}$$

Since set $\mathcal{F}$ is convex and the objective is strictly concave, there exists a unique solution $\mathbf{f}^*$ to the maximization problem. Corresponding $\mathbf{y}^*, \mathbf{x}^*$ also exist but are not necessarily unique.

Let us denote with μ_i^c and ξ_{iJ}^c the Lagrangian multipliers associated with inequalities (2) and (3), respectively. To simplify the notation we will also define $\mu_{\mathrm{Dst}(c)}^c = 0$. We can write the KKT conditions at the optimal point for the constraints (2) and (3)

$$\mu_i^{c*}\left(\sum_{j\neq i} y_{ij}^{c*} - \sum_{j\neq i} y_{ji}^{c*} - f_c^* 1_{i=\mathrm{Src}(c)}\right) = 0, \tag{6}$$

$$\xi_{iJ}^{c*}\left(x_{iJ}^{c*} - \sum_{j\in J} y_{ij}^{c*}\right) = 0, \tag{7}$$

Also, combining the condition that the rates are non-negative ($f_c \geq 0$) and the gradient KKT condition, we can write

$$f_c^*\left(U_c'(f_c^*) - \mu_{\mathrm{Src}(c)}^{c*}\right) = 0. \tag{8}$$

Note that here we do not write the KKT for the constraint (4) but we keep it in the explicit form (see for example (10)). We see that $\mu_i^{c*} = 0$ if egress traffic is larger than ingress traffic and there is no accumulation of credits at node i. Hence intuitively we can relate μ_i^{c*} to $q_i^c(t)$, the number of credits for flow c queued at i. Similarly we can relate ξ_{iJ}^{c*} to the number of packets queued for broadcasting at i. In Section III-C we express this relationship more formally. We will also use (8) to develop a flow control algorithm.

As a consequence of KKT, and by optimizing the dual problem [23] one can derive the conditions

$$0 \geq \mu_i^{c*} - \mu_j^{c*} - \sum_{J\subseteq\mathcal{N}\,|\,j\in J} \xi_{iJ}^{c*}, \tag{9}$$

$$C_{iJ}^* = C_{iJ}(\mathbf{P}^*, R_i^*, S^*), \tag{10}$$

$$(\mathbf{P}^*, R_i^*, S^*) = \underset{\{(\mathbf{P},R_i,S)\}}{\mathrm{argmax}} \sum_i \max_c \sum_J \xi_{iJ}^{c*} C_{iJ}(\mathbf{P}, R_i, S).$$

We will use (10) in Section III-C to derive the optimal scheduling.

C. Maximization Algorithm

We now present an algorithm that converges to the optimal value of (5).

Node and Transport Credits : Recall that $q_i^c(t)$ is the amount of commodity c credits queued at node i. We call such credits *node credits*. In addition, let $w_{iJ}^c(t)$ be the number of credits of commodity c queued at i and corresponding to packets that have to be delivered to *any* of the nodes in J (as previously decided by the credit transmission scheme). We call these *transport credits*.

When a credit for flow c is transferred from node i to node j, we decrease q_i^c, we increase q_j^c, and we increase w_{iJ}^c for all $J \ni j$ (all by one unit). Note that this transfer happens instantly, and before a corresponding packet has actually been transmitted. We decrease w_{iJ}^c when a packet from flow c is actually delivered from i to any of the nodes in J. The amount of node credits is conserved: when a responsibility for a credit is transfered from i to j we decrease q_i^c and increase q_j^c. Transport credits are of a different nature. They are created when a forwarding decision is made and are destroyed when the actual delivery takes place. Transport credits are local and they are never transferred between nodes.

Routing protocol: Node credits represent intentions of packet transmissions and a routing protocol describes when and how such node credits transferred. Let $y_{ij}^c(t)$ be the number of node credits for flow c transferred from node i to node j at time t and let us define $w_{ij}^c(t) = \sum_{X\subseteq\mathcal{N}\,|\,j\in X} w_{iX}^c(t)$. A *back-pressure* between nodes i and j is defined as

$$z_{ij}^c(t) = q_i^c(t) - w_{ij}^c(t) - q_j^c(t), \tag{11}$$

the difference between the excess credits queued of flow c at node i not destined for node j ($q_i^c - w_{ij}^c$) and the node credits at node j (q_j^c). Back-pressure in (11) includes the credits queued at the next hop as well as the credits assigned to different broadcast regions, which is the main difference compared to previous work [17], [18], [12].

A credit is transferred from i to j with the following dynamics

$$y_{ij}^c(t) = \frac{q_i^c(t)}{|\{k\,|\,z_{ik}^c(t) > 0\}|} 1_{\{z_{ij}^c(t)>0\}}, \tag{12}$$

where $1_{\{x>0\}}$ is 1 if $x > 0$ or 0 otherwise, and $|\{k\,|\,z_{ik}^c(t) > 0\}|$ is the number of neighbors of i that have a positive back-pressure for flow c. The queue $q_i^c(t)$ then has the following

dynamics in time

$$q_i^c(t+1) = q_i^c(t) + f_c(t)1_{i=\mathrm{Src}(c)} + \sum_j y_{ji}^c(t) - \sum_j y_{ij}^c(t).$$

Note that by definition $\sum_j y_{ij}^c(t) \leq q_i^c(t)$ hence $q_i^c(t)$ is always positive.

The intuition is as follows. We transfer a credit from i to j only if the back-pressure is positive. Moreover, we share all of the available credits $q_i^c(t)$ equally among all neighbors with a positive back-pressure.

Scheduling, rate and power control: The optimal centralized scheduling, rate and power control algorithm is the tuple $(S(t), P(t), R(t), A(t))$ that solves the following optimization problem

$$\overline{WC}_i(t, \mathbf{P}, R_i, S) = \max_c \sum_J w_{iJ}^c(t) C_{iJ}(\mathbf{P}, R_i, S), \quad (13)$$

$$(S(t), \mathbf{P}(t), \mathbf{R}(t)) = \operatorname*{argmax}_{S,\mathbf{P},\mathbf{R}} \sum_{i\in\mathcal{N}} \overline{WC}_i(t, \mathbf{P}, R_i, S), \quad (14)$$

$$C_{iJ}(t) = C_{iJ}(\mathbf{P}(t), R_i(t), S(t)), \quad (15)$$

$$c_i^*(t) = \operatorname*{argmax}_c \sum_K w_{iK}^c(t) C_{iK}(t), \quad (16)$$

$$A_{ic}(t) = 1_{\{c=c_i^*(t)\}}, \quad (17)$$

$$x_{iJ}^c(t) = A_{ic} C_{iJ}(t). \quad (18)$$

Intuitively, (13) follows from (10) and the fact we can equate ξ_{iJ}^c and w_{iJ}^c, since the transport credit update equation (21) corresponds to the gradient update equation of ξ_{iJ}^c. The other update rules follow directly from KKT conditions.

Equations (13)-(18) represent a joint scheduling, rate, and power control problem. We find the optimal scheduling, power and rate control $(S(t), P(t), R(t))$ by solving (14). Then, equation (16) is used to select which flow will be transmitted by each node in slot t.

The main novelty in our approach is that we explicitly incorporate all broadcast regions in the scheduling algorithm in Equation (13) through broadcast transport credits $w_{iJ}^c(t)$. This is in contrast to previous works on back-pressure [17], [18], [12], [14] that are not able to exploit the broadcast diversity. It is only [19] that consider network optimization with broadcast diversity, but using different optimization techniques. Also, unlike [5], [4], we are able to make a trade-off between broadcast diversity and network congestion.

Another difference with respect to [17], [18], [12], [14] is that, as explained in Section II-B, we cannot decouple the flow selection process $A(t)$ and routing/scheduling/rate/power control. Also, unlike in [17], [18], [12], [19], we do not explicitly use back-pressure information for scheduling in (13) - (18); instead we use transport credits $w_{iJ}^c(t)$.

The algorithm (13)-(18) is centralized. Observe that all equations except (14) and (15) use *local information only*. Hence, with the exception of (14) and (15), the problem could have been solved with a *distributed* algorithm. We use this observation to propose a heuristic based on modified rules (14) and (15), and to derive a practical, distributed protocol that is presented in Section IV.

Flow control: The optimal flow rate at the source, $f_c(t)$ can be calculated using a primal-dual approach, as in [12]

$$f_c(t+1) = \left[f_c(t) + \gamma\left(U_c'(f_c(t)) - q_{\mathrm{Src}(c)}^c(t)\right)\right]^+, \quad (19)$$

where $[x]^+ = \max\{x, 0\}$. Each flow adapts its rate based on the previous rate and current number of credits queuing for transmission at the source node for that flow ($q_{\mathrm{Src}(c)}$). The primal-dual approach well describes additive-increase multiplicative-decrease transport protocols, like TCP [11].

D. Convergence Of The Algorithm

We now consider a fluid model of the system, and show that it converges to the optimal point. Analysis of a discrete-time model can be derived from our fluid-model analysis, using a similar approach to [12].

We assume that time is continuous and that queue evolutions are governed by the following differential equations

$$\dot{q}_i^c(t) = \left(f_c(t)1_{i=\mathrm{Src}(c)} + \sum_j y_{ji}^c(t) - \sum_j y_{ij}^c(t)\right)_{q_i^c(t)\geq 0} \quad (20)$$

$$\dot{w}_{iJ}^c(t) = \left(\sum_{j\in J} y_{ij}^c(t) - x_{iJ}^c(t)\right)_{w_{iJ}^c(t)\geq 0} \quad (21)$$

where $(x)_{y\geq z} = x$ if $y \geq z$ and $(x)_{y\geq z} = 0$ otherwise. Similarly, flow rate evolution in the fluid-model is given by

$$\dot{f}_c(t) = \gamma\left(U_c'(f_c(t)) - q_{\mathrm{Src}(c)}^c\right)_{f_c(t)\geq 0}. \quad (22)$$

We next prove that the algorithm presented in Section III-C stabilizes the system with flow rates that maximize the optimization problem (5).

Definition 2: We say that link (i, j) is active for flow c if there exist a finite number T such that for each t that satisfies $y_{ij}^c(t) > 0$, there exists $t', t < t' < t+T$ such that $y_{ij}^c(t') > 0$.

Theorem 2: Starting from any vectors $\mathbf{f}(0), \mathbf{q}(0), \mathbf{w}(0)$ and applying rules (11)-(22), the rate vector $\mathbf{f}(t)$ converges to $\mathbf{f}^*$ as t goes to infinity. Furthermore, queue sizes $q_i^c(t), q_j^c(t)$ and $w_{ij}^c(t)$ on all active links (i, j) for flow c are bounded, and converge to the shadow prices μ_i^{c*}, μ_j^{c*} and ξ_{iJ}^{c*} respectively.

Also, if a node is completely disconnected from the rest of the network, or in any way not used by a flow, credits will neither arrive to nor will leave from the node. Thus technically, we cannot guarantee that an arbitrary initial number of credits at this node will converge to any particular value. Instead, we consider only links that have at least "some" traffic throughout the network run-time (called active links, formalized in Definition 2).

The proof uses a Lyapunov function with stability defined on the set of active link, and we show that on all active links that carry a positive amount of traffic, the delays are bounded and hence the system is stable. The details are given in the Appendix.

E. Comparison with MORE

In this section we compare the performance of our algorithm with the MORE forwarding algorithm described in [5], [24]. We summarize it here for sake of completeness. Consider a single flow and the delivery of a single packet from the source of the flow Src to the destination of the flow Dst. Let β_i be the number of transmissions made by node i to successfully deliver the packet. The goal of MORE is to minimize the number of transmissions [24, Eq.(1)-(4)]:

$$\text{argmin} \sum_{i\in\mathcal{N}} \beta_i \tag{23}$$

$$\text{s.t.} \sum_{j\in\mathcal{N}} \hat{y}_{ij} - \sum_{j\in\mathcal{N}} \hat{y}_{ji} = 1_{\{i=\text{Src}\}} \tag{24}$$

$$\hat{y}_{ij} \geq 0, \tag{25}$$

$$\beta_i C_{iJ} \geq \sum_{j\in J} \hat{y}_{ij}. \tag{26}$$

Note that the MORE forwarding algorithm is designed for a single flow. As the authors note in [5], [24], its performance drops as the number of flows increases.

Our algorithm converges to the optimal solution of the optimization problem (5) (as shown in Theorem 2), hence MORE is at best as good as our algorithm. We first show under what conditions MORE is guaranteed to give the optimal solution. We then illustrate by two examples that MORE can yield strictly suboptimal rate allocations.

Theorem 3: If there is only one flow in the system, if transmission rates and powers of all nodes are fixed and if only one node can transmit at a time (that is $|\text{SRC}(S)| = 1$ for all $S \in \mathcal{S}$), MORE and our algorithm give the same performance.

Proof: Since only one node can transmit at a time, we have $\mathcal{S} = \mathcal{N}$. Furthermore, transmission powers and rates are fixed, hence (3) and (4) reads as $\sum_{j\in J} y_{ij} \leq \alpha_i C_{iJ}$. We also omit c as there is only one flow in the system.

We start with the MORE optimization problem (23) - (26) and we introduce $f = 1/\left(\sum_{i\in\mathcal{N}} \beta_i\right)$, $y_{ij} = f\,\hat{y}_{ij}$ and $\alpha_i = f\,\beta_i$. The optimization (23) - (26) is then equivalent to

$$\min 1/f \tag{27}$$

$$s.t. \sum_{j\in\mathcal{N}} y_{ij} - \sum_{j\in\mathcal{N}} y_{ji} = f 1_{\{i=\text{Src}\}} \tag{28}$$

$$\alpha_i C_{iJ} \geq \sum_{j\in J} y_{ij}, \tag{29}$$

$$\sum_{i\in\mathcal{N}} \alpha_i = 1, \tag{30}$$

which is exactly the optimization problem (5). ∎

We next give two examples where the performance of MORE is strictly suboptimal. Consider a hexagonal network depicted in Figure 3. Let us first consider a case with a single flow f_1, ($f_2 = 0$), and where $p_{12} = p_{24} = p_{46} = 0.8$ and $p_{13} = p_{35} = p_{56} = 0.2$. Since not all links interfere, the conditions of Theorem 3 are clearly not satisfied. The optimal rate allocation that maximizes (5) is $f = 0.313$ with $y_{1-2-4-6} = 0.251$ and $y_{1-3-5-6} = 0.063$. However, MORE will transmit all packets over the path $1-2-4-6$, hence the total rate will be $f_{\text{MORE}} = 0.267$, some 15% less than the optimal. Intuitively, the reason why MORE is suboptimal is that it does not consider possibility that links $3-5$ and $5-6$ transmit in parallel with $4-6$ and $2-4$. (Recall that MORE's goal is to minimize the number of transmissions and not to maximize the flow rate.) It will then conclude that forwarding any packet to 3 is largely suboptimal, since links $3-5$ and $5-6$ are of a bad quality. Thus, if p_{35} and p_{56} are sufficiently smaller than p_{12}, p_{24} and p_{46}, as in this example, MORE will not use route $1-3-5-6$ at all.

In the second example we again consider the same hexagonal network but with two flows (f_1, f_2) active. Since MORE does not take into consideration contention among flows, it will again assign all traffic to the path $1-2-4-6$. This traffic will contend with the traffic from flow 2, and feasible end-to-end rate allocations have to satisfy $3f_1 + f_2 = p$. Note that the routing scheme is fixed by MORE and does not depend on flow control applied by transport layer. If for example the transport layer on top of MORE is designed to maximize log utility, the optimal rates will be $f_1^{\text{MORE}} = p_{12}/6 = 0.133$, $f_2^{\text{MORE}} = p_{12}/2 = 0.4$. On the contrary, our distributed algorithm will adapt routing to contentions among flows, and it will assign $y_1 = 0.12$, $y_2 = 0.03$, $f_1 = 0.15$ and $f_2 = y_3 = 0.4$. As we can see, our algorithm balanced flow 1 by decreasing y_1 and increasing y_2. As a result, the rate of flow f_2 stayed the same while the rate of f_1 increased.

IV. Practical Issues

In this section we consider two practical issues that concern implementation of the protocol proposed in Section III in a mesh network: finite coding generation size and rate adaptation for randomized scheduling. We leave other practical issues, such as the effect of delayed feedback, for future work.

A. Finite Generation Size

Previous results assume that generation size used for network coding tends to infinity (see Theorem 1). Practical reasons, such as the complexity and performance of decoding, and header overhead for storing the coefficient vector, require us to limit the size of the header; some practical systems limit the size to 32 bytes [5], [6]. We now modify our optimization framework for a finite generation size.

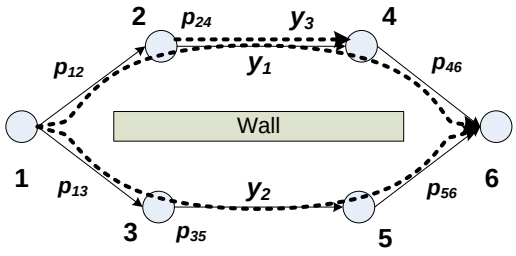
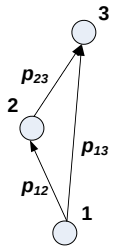


Fig. 3. A network with 6 nodes. Due to a dividing wall, nodes 2 and 4 do not interfere with nodes 3 and 5. The set of activation profiles is $\mathcal{S} = \{\{(1,\{2,3\})\},\{(2,4)\},\{(3,5)\},\{(4,6)\},\{(5,6)\},\{(2,4),(3,5)\},\{(2,4),(5,6)\},\{(3,5),(4,6)\}\}$. There are two flows in the system, flow $f_1 = y_1 + y_2$ which is assigned 2 routes ($y_1 = y_{1-2-4-6}$ and $y_2 = y_{1-3-5-6}$) and flow $f_2 = y_3$ which is assigned a single route ($y_3 = y_{2-4}$).



$$
\begin{aligned}
w_{12}(0,1) &= 0,\\
w_{12}(0,2) &= 5,\\
w_{1\{2,3\}}(0,1) &= 0,\\
w_{1\{2,3\}}(0,2) &= 4,\\
w_{13}(0,1) &= 1,\\
w_{13}(0,2) &= 0
\end{aligned}
$$

Fig. 4. A simple example of a network with nodes $\{1,2,3\}$ and a single flow, going from 1 to 3 (directly or via node 2), where set $p_{12} = p_{23} = 1$ and p_{13} is close to 0. The values of corresponding transport credits $w_{iJ}(t,g)$ are given on the right at time $t = 0$ for generations $g = 1, 2$ (in the example we assume only two generations are in the networks).

Let $\mathcal{G}$ be the set of generations. Let us define $q_i^c(t,g)$ and $w_{iJ}^c(t,g)$ be the number of node credits and transport credits for generation $g \in \mathcal{G}$ of flow c queued at i. Similarly, we define $f_c(t,g), y_{ij}^c(t,g), x_{iJ}^c(t,g)$ as before but with respect to generation g (thus we have for example $y_{ij}^c(t) = \sum_{g\in\mathcal{G}} y_{ij}^c(t,g)$, and by analogy for the other variables). The encoding processes $f_c(t,g)$ are defined at the source. For example, if the size of each generation is G, we have $\int_t f_c(t,g)\mathrm{d}t = G$ for all $g \in \mathcal{G}$.

Extending the distributed maximization algorithm from Section III-C to this setting is not straightforward. For example, a naive way to modify scheduling rule (18) would be to schedule the oldest generation available

$$g_i^*(c,t) = \min\{g \,|\, (\exists J) w_{iJ}^c(t,g) > 0\}, \tag{31}$$

that is $x_{iJ}^c(t,g) = C_{iJ}(t)$, if $c = c_i^*(t)$ and $g = g_i^*(c_i^*(t),t)$. However, this rule may yield poor rates.

To see why, consider the example of Figure 4 where one link $(1-3)$ has very poor quality compared to the others. For simplicity we assume there are only two generations in the system and we assume that node credits $q_i^c(t)$ are constant hence only the transport credits are the focus of the scheduling policy. The oldest generation that has a transport credit at node 1 is generation 1. Thus, when node 1 is selected to transmit, it will transmit a packet from generation $g_1^*(t) = 1$, according to (31). On one hand, node 2 will certainly receive the transmitted packet but, since $w_{12}(0,1) = w_{1\{2,3\}}(0,1) = 0$, node 2 does not need more packets from generation 1 and the received packet will be useless. On the other hand, node 3 is not likely to receive the transmitted packet as $p_{13} \approx 0$. Therefore, in the next slot $w_{13}(1,1)$ will most probably remain at 1 and generation one is again selected for transmission. The same situation will repeat until the transmitted packet is finally received by node 3, and only then we will be able to transmit a packet from generation 2.

It is easy to see that in such a scenario we may completely starve the network. Instead of benefiting from diversity, the opportunistic routing acts as a hindrance. The main reason for starvation is the fact that the naive approach implicitly restricts scheduling diversity over a single generation. Observe that the problem persists even when the number of generations increases and the queues build up; the naive approach tries to maintain the optimal traffic splitting strictly per each generation instead on a cross-generation, long-term average. Nevertheless, there is no reason why we should not be able to exploit the diversity of link 1-3, because even a few occasional packets transmitted over that link will improve overall performance.

Finding a jointly optimal coding and scheduling strategy that maximizes system utility for finite generation sizes is a difficult problem. Apart from the scheduling issue, when the generation size becomes finite, the coding results from [16] no longer hold. This implies that packets received at the destination may not be linearly independent and Theorem 1 does not hold. Instead, we propose a heuristic inspired by the proof of Theorem 2, which minimizes the drift $\dot{W}(\mathbf{f}(t), \mathbf{q}(t), \mathbf{w}(t))$. It consists of modifying rules (12) and (18) to

$$z_{ij}^c(t,g) = q_i^c(t,g) - w_{ij}^c(t,g) - q_j^c(t,g), \tag{32}$$

$$y_{ij}^c(t,g) = \frac{q_i^c(t,g)}{|\{k \,|\, z_{ik}^c(t,g) > 0\}|} 1_{\{z_{ij}^c(t,g)>0\}}, \tag{33}$$

$$g_i^*(c,t) = \operatorname*{argmax}_g \sum_J w_{iJ}^c(t,g) x_{iJ}^c(t)\, 1_{\{w_{iJ}^c(t,g)>0\}}, \tag{34}$$

$$x_{iJ}^c(t,g) = \begin{cases} C_{iJ}(t) & \text{if } c = c_i^*(t), g = g_i^*(c,t) \\ 0, & \text{otherwise} \end{cases}. \tag{35}$$

where $1_{\{w_{iJ}^c(t,g)>0\}}$ is true when there are queued transport credits w_{iJ}^c for generation g.

Intuitively, the idea behind the heuristic is to transmit the generation that has the highest chance of being useful. We select flow c to transmit according to (16). The benefit of such a transmission is $\sum_K w_{iK}^c(t) C_{iK}(t)$, the expected decrease in transport credits. However, this is only true if the generation size G is infinite. The transmitted packet from generation g will not be useful if the generation size is finite, and some of the nodes no longer need any more packets from generation g (that is $w_{iJ}^c(t,g) = 0$ for some J, as illustrated in the example). To maximizes the actual expected decrease of the amount of transport credits we select a generation g that maximizes $\sum_J w_{iJ}^c(t,g) x_{iJ}^c(t,g)\, 1_{\{w_{iJ}^c(t,g)>0\}}$, which is indeed (34).

To see how the new policy works, consider again the example of Figure 4. Unlike the naive policy (31) that selects generation 1 as the oldest generation among all queued generations, the new policy (34) will select the most useful generation which is generation 2, which circumvents network starvation. A detailed explanation of how this policy is derived is given in the Appendix. Performance simulations for finite generation sizes are given in Section V.

The policy (33) - (35) needs a slight caveat, in that some credits from old generations may get stuck in the network (as the credit from generation 1 did in the previous example). A straightforward extension is to reassign the credits from the selected generation $g_i^*(c,t)$ and from the oldest queued generation such that the total number of credits is not altered, and to guarantee in-order delivery. However in practice, unless a network is very asymmetric, this is not needed, as the simulations in Section V verify.

B. Rate Adaptation For 802.11-compatible Scheduling

Finding the optimal scheduling rule (14) is an NP-hard centralized optimization problem, as Sharma et al. show [15].

Some recent research [25], [15], [26] explores decentralized implementations of similar problems. Applying these ideas to our setting is outside the scope of this paper, and left for future work. Instead, we consider a more realistic, suboptimal scheduling process and we show how our algorithm can be applied as a distributed heuristic.

We assume that nodes always transmit packet at the full power $P_i(t) = \{0, P^{MAX}\}$, which reflects current practice in most existing wireless mesh networks deployments. We call a set of feasible activation profiles $\mathcal{S}$ *802.11-compatible* if for all $S \in \mathcal{S}$ and for all $(i_1, J_1) \in \mathcal{S}$ there is no $(i_2, J_2) \in \mathcal{S}$ such that reception probabilities $p_{i_1,i_2} > 0, (\exists j \in J_2) p_{i_1,j} > 0$ or $(\exists j \in J_1) p_{i_2,j} > 0$. Intuitively, this corresponds to 802.11-like protocol with RTS/CTS mechanism. When node i_1 establishes communication with nodes J_1, all nodes involved in communication send an RTS/CTS. All nodes that hear the RTS/CTS ($p > 0$) will be prevented from transmission or reception during the same slot.

Furthermore, we will assume that the underlying scheduling process $\{S(t)\}_t$ is outside our control, and that it is independent of the actions of our protocol. At every time t, the scheduling process will select a set of non-interfering nodes $I(t) \in \mathcal{N}$ to transmit (i.e. for each $i, j \in I(t), p_{ij} = 0$). Each node $i \in I(t)$ has a set of possible destinations $J_i(t) = \{j \in \mathcal{N} \,|\, p_{ij} > 0, (\forall k \in I(t), k \neq i), p_{kj} = 0\}$, which in turns define a schedule $S(t) = \{i, J_i\}_{i \in I(t)}$. A set of activation profiles $\mathcal{S} = \{\{i, J_i\}_{i \in I} \,|\, I \in \mathcal{P}(\mathcal{N})\}$ is clearly 802.11-compatible.

With such a schedule $S(t)$, the optimization (13) - (18) simplifies to

$$(c_i^*(t), R_i^*(t)) = \underset{c, R_i}{\operatorname{argmax}} \sum_{K \subseteq J_i} w_{iK}^c(t) \bar{C}_{iK}(R_i). \qquad (36)$$

where $\bar{C}_{iK}(R_i) = \mathbb{E}_S[C_{iK}(R_i, S)]$ are the average rates observed over a long period of time. This optimization can be easily solved in a distributed manner, locally and separately at each node. Node i can estimate $\bar{C}_{iK}(R_i)$ either by probing, or using statistics from previous transmissions. In practice, as reported in [27], successful transmissions T_{ij} and T_{ik} are often independent for $j \neq k$ which further simplifies the estimation. The rest of (36) clearly relies only on local information. Alternatively, if one deploys a form of interference-aware scheduling (e.g. [28]) which disposes of $S(t)$ in any slot, algorithm (36) can be improved by estimating $C_{iK}(R_i, S)$ for each scheduling policy $S = S(t)$ separately, instead of using the average estimate $\bar{C}_{iK}(R_i)$. In the simulation part of our paper we implement the simpler, interference-oblivious form, as given in (36).

Note that for an arbitrary scheduling process $\{S(t)\}_t$, the distributed routing (12) and rate adaptation (36) algorithm do not necessarily minimize the optimization problem (5). The optimal algorithm will depend on the characteristics of $\{S(t)\}_t$, $\{S(t)\}_t$ will depend on our routing algorithm, and it is difficult to characterize these dependences. We present (12) and (36) as a heuristic that can be used as a practical implementations of opportunistic multi-path routing in networks with 802.11-compatible scheduling. We illustrate by simulations in Section V that in the case of random, 802.11-compatible scheduling, the heuristic (36) outperforms a conventional, single-path routing approach.

V. Simulation Results

We now present simulation results which quantify the performance advantages of the opportunistic routing, scheduling and flow control algorithms defined in the previous sections. We are primarily interested in algorithms that can be applied in 802.11-like mesh networks, where the scheduling algorithm is not under our control. Hence in our simulations we used an 802.11-compatible schedule $\{S(t)\}_t$, as defined in Section IV-B, assuming $I(t)$ is randomly selected among backlogged nodes.

We use the roofnet network topology based on 802.11b cards, given in [4], for our simulations. We further assume the (802.11-compatible) node-exclusive model with random channel parameters from [1]). We developed a slotted discrete-event simulator that implements the routing, flow and rate control algorithms. A scheduler randomly selects a set of non-exclusive nodes for transmission in each slot. The amount of data transmitted in each slot is proportional to the transmission rate and the packet loss probability is obtained from [1] (assuming that a concurrent transmission is allowed; otherwise it is set to 0). Unless stated otherwise, we assume finite generation size of 32 and use rules (33) - (35) to select which generation to transmit. In addition, we allow credit and packet transmissions by a node only if a node has received an innovative packet for a given generation since the previous transmission. We used $U_c(\cdot) = \log(\cdot)$, hence the rate allocation that maximizes (5) is the proportionally fair rate allocation [22].

We looked at three performance metrics. The first one is the improvement in total utility $\sum_c U(f_c) - \sum_c U(f'_c)$. Allocation $\mathbf{f}$ is better than $\mathbf{f'}$ if the sum is positive. The proportional fair rate maximizes the optimization problem (5) hence has the highest utility.[2] The second metric is the total rate improvement $\sum_c f_c / \sum_c f'_c$. Allocation $\mathbf{f}$ is better than $\mathbf{f'}$ if the quotient is larger than 1. The proportionally fair allocation does not always have highest total rate. The third metric is the Jain's fairness index improvement. Jain's fairness index is defined as $FI(\mathbf{f}) = (\sum_c f_c)^2 / (|\mathcal{C}| \sum_c f_c^2)$ and the Jain's fairness index improvement is $FI(\mathbf{f})/FI(\mathbf{f'})$. Note that the fairness index can be deceiving as a metric in some cases: if $\mathbf{f}$ has all rates larger than $\mathbf{f'}$ it may still have smaller fairness index although the system has clearly improved.

We compared our algorithm with a conventional, single path routing algorithm, and with the MORE algorithm [5]. To make the comparison fair, we assumed that the single-path routing algorithm used the same kind of jointly-optimal routing and flow-control approach as our scheme, which boils down to [12], constrained to the best path. In contrast, MORE does not integrate flow control or flow scheduling with the routing algorithm. When simulating the MORE algorithm, defined in [5], [24], we assumed that each source had a large backlog

[2]Since in the simulations we use random and not the optimal scheduling, the resulting rate allocation does not necessarily have the highest utility.

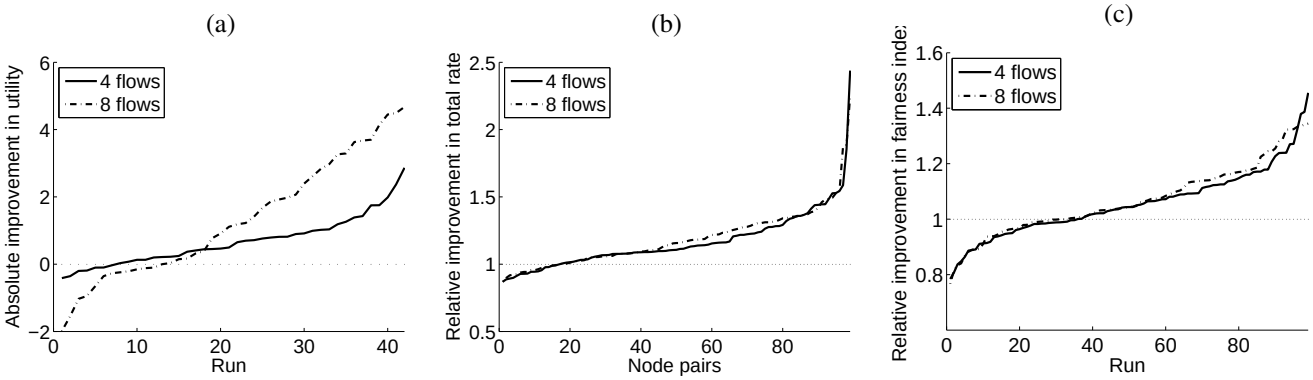


Fig. 5. Cumulative performance improvement of opportunistic over single-path (f_c^s - rates using single path, f_c^m - rates using multiple paths): (a) Absolute improvement in utility ($\sum_c \log(f_c^m) - \sum_c \log(f_c^s)$); (b) Relative improvement in total rate ($\sum_c f_c^m / \sum_c f_c^s$); (c) Relative improvement in fairness index ($FI(\mathbf{f}^m)/FI(\mathbf{f}^s)$). We perform the experiments with 4 and 8 concurrent flows. In all cases we ran 100 experiments and sorted them by performance improvement. In many cases, for single-path routing, some flow had zero rates for the duration of the simulation, caused by slow convergence; we omitted such plots (as they would give an infinite utility difference).

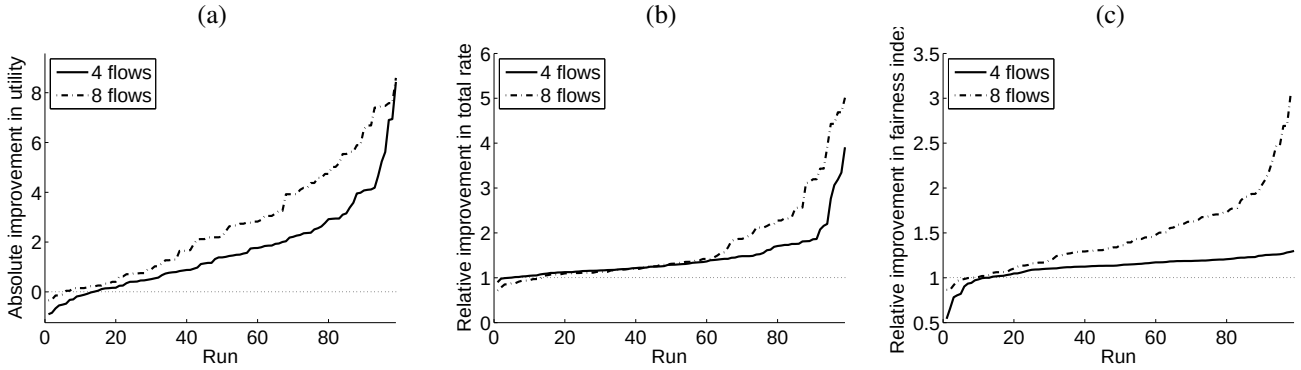


Fig. 6. Cumulative performance improvement of our algorithm over MORE (f_c^s - rates using single path, f_c^m - rates using multiple paths): (a) Relative improvement in utility ($\sum_c \log(f_c^m) - \sum_c \log(f_c^s)$); (b) Improvement in total rate ($\sum_c f_c^m / \sum_c f_c^s$); (c) Improvement in fairness index ($FI(\mathbf{f}^m)/FI(\mathbf{f}^s)$). We perform the experiments with 4 and 8 concurrent flows. In all cases we ran 100 experiments then sorted them by performance improvement.

of packets to transmit, and that each relay performed FIFO scheduling among packets from different flows.

We ran simulations to obtain end-to-end rate allocations. Figure 7, (a) illustrates the optimal rate allocations, obtained by our algorithm, for 8 randomly selected source-destination pairs on one example. In this case, we can see both utility and total rate increase if we use the opportunistic routing instead of the single-path routing, where we benefit from broadcast and multi-path diversity.

We then ran the previous experiment with 100 random realizations of 4 or 8 coexisting unicast sessions and compared the performances of the different algorithms with respect to the two performance metrics.

Single Path vs. Opportunistic: We start by illustrating the benefits of the opportunistic routing over the single-path routing in Figure 5. We first look at the network utility. The rate allocation obtained by the optimal algorithm (Section III-C) always maximizes the utility. However, this is not the case for the distributed heuristic (Section IV-B). In our simulations we saw that in about 90% of the runs, the distributed heuristic for opportunistic routing achieves higher utility than does single-path routing. In only about 10% – 15% of cases is the utility for single-path routing higher.

Also, in more than 80% of runs, our decentralized heuristic achieved higher total rate than the conventional, single-path algorithm. In more than half of the runs, the total rate has increased by 20%, and in some cases by over 100%. From these results we see that there is a significant advantage in using our opportunistic routing algorithm over the single-path one. Fairness index also improves in more than half of the cases.

Decentralized Heuristic vs. MORE: We next compare our decentralized heuristics with MORE. The results are depicted in Figure 6. Network utility is increased in about 90% of the runs. Total rate is increased in almost all of the runs, sometimes up to a factor of 4 – 5. The fairness index has also increased in most of the cases. The performance of MORE drops with the number of flows.

From these results we can see that in many cases MORE behaves worse than the single-path routing. This resonates with the findings of [5] where the benefits of opportunistic routing decrease as the number of flows increase (for an explanation, see Section III-E). MORE is essentially a routing protocol, whereas our single-path routing algorithm also includes more intelligent flow control and flow scheduling.

Effects of Finite Generation Size: Figure 7, (b) illustrates the impact of a finite generation size. The performance drop is due to imperfect generation scheduling (34) and occasional

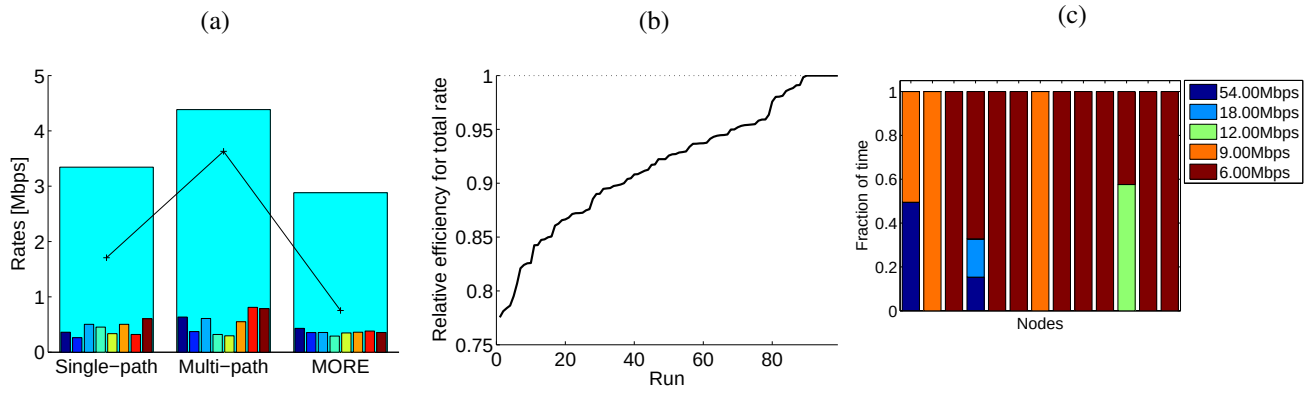


Fig. 7. (a) End-to-end rate allocation example: eight flows among randomly selected source-destination pairs. The vertical axis shows flows rates. Small bars denote rates per flow. Large bars show total rate. The marks connected by a line gives utilities with an arbitrary scaling (to fit the figure). (b) Relative efficiency for total rate ($\sum_c f_c^{finite} / \sum_c f_c^{infinite}$) for opportunistic routing with a finite generation size of $G = 32$. (c) Optimal distribution of PHY rates for roofnet network with 802.11a cards, for one random selection of 8 source-destination pairs.

linear dependency of received packets. We can see that a finite generation size of 32 packets can cause a performance drop of up to 22% with respect to the infinite generation size. [3] However, a larger generation size would impose more overhead in transmitting larger coefficients in the packet header.

The Optimal PHY Rate Selection: Finally, we consider the optimal PHY rate selection at different nodes. We analyze how frequently each node uses each PHY rate. In the case of roofnet topology with 802.11b cards we find that in almost all cases it is optimal to use the highest rate of 11 Mbps, which confirms the findings from [5]. We then used the SNR data from roofnet topology and measurements from [29] to analyze the approximatively optimal performance in the same network topology with 802.11a cards. The results are depicted in Figure 7, (c). As expected, the optimal PHY rate selection is no longer uniform, a consequence of the large number of available rates. This demonstrates the need for an intelligent PHY rate selection algorithm.

VI. Related Work

One of the first uses of opportunistic routing for unicast sessions in wireless mesh networks is presented in [4]. It has been extended in [5] to include network coding to facilitate scheduling. However, in [5] the authors do not explicitly consider multiple flows, fairness, or scheduling, and in fact show that the performance benefit drops as number of flows increases.

One of the works most similar to our own is the optimization framework for opportunistic routing that minimize power consumption, presented in [19], which shows significant benefits over [4]. Nevertheless, [19] considers neither network coding, nor the TCP-like primal-dual rate adaptation. It is not clear how to generalize [19] to use network coding with finite generation size and to derive a generation scheduling policy analog to (34).

Another related paper is the work on energy-efficient opportunistic network coding [30], which use a similar formulation of an optimization problem, which can further be extended to intersession network coding. [30] does not consider a problem of a limited generation size, TCP dynamics, nor a suboptimal rate allocation algorithms and has a less general interference model. Cross-layer design for network coding with unicast or multi-cast sessions is considered in [9]; however, [9] considers only stability and not any form of rate maximization. End-to-end rate maximization of a single flow is also considered in [31].

Several theoretical analysis of linear network coding algorithms for unicast sessions have been performed [16], [10]. Network coding for unicast sessions is used also in COPE [32]. Compared to COPE, we perform encoding operations only between packets of the same flow; in that respect our approach is orthogonal to COPE. Optimal control of intersession network coding is presented for example in [33]; however, it does not consider opportunistic routing and its inherent diversity.

Our work is an example of cross-layer optimization, and we have built on top of and extended exiting research. Cross-layer design in wireless is a widely research topics (see [7], [13], [14] and references therein). Optimizing network performance in terms of network utility is originally proposed in [22]; see [11], [14] for an overview. Our primal-dual approach is similar to [11], where it is shown that it can capture different versions of TCP. None of the algorithms in [7], [13], [14], [11] consider opportunistic routing, broadcast diversity and intra-session network coding.

VII. Conclusions

This paper proposes an optimization framework for addressing questions of multi-path routing in wireless mesh networks. We have extended previous work by incorporating the broadcast nature of wireless and simultaneously addressing fairness issues. Implicit in our approach is the use of network coding, which enables us to define notions of credits that are associated with number of packets in a generation, rather than specific packets. Using our framework we show that our algorithm significantly outperforms single-path routing and MORE [5].

[3] Note that an additional performance drop for finite generation size may occur due to imperfect signaling, but we do not consider it in our model.

When scheduling is pre-determined by a MAC protocol, such as by random scheduling or 802.11-like scheduling, we have shown how our approach leads to a distributed heuristic, which still outperforms existing approaches. Using a simulation results on a realistic topology, we found in our examples that for 802.11b, using the maximal rate is optimal, but for 802.11a this was not the case. We have addressed some of the practical issues associated with having a finite generation size for network codes.

Our primal-dual rate adaptation can be used to model window-based flow control schemes, such as TCP. The performance of applications that run on top of our system and use TCP is an interesting open problem. Another interesting direction is to analyze the performance of our protocol with more realistic signaling schemes.

REFERENCES

[1] "MIT roofnet - publications and trace data," http://pdos.csail.mit.edu/roofnet/doku.php?id=publications, 2005.

[2] J. Eriksson, S. Agarwal, P. Bahl, and J. Padhye, "Feasibility study of mesh networks for all-wireless offices," in *ACM/Usenix MobiSys*, 2006.

[3] M. Caesar, M. Castro, E. Nightingale, G. O'Shea, and A. Rowstron, "Virtual ring routing: Network routing inspired by DHTs," in *ACM SigComm*, 2006.

[4] S. Biswas and R. Morris, "ExOR: opportunistic multi-hop routing for wireless networks," in *ACM SIGCOMM*, 2005.

[5] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, "Trading structure for randomness in wireless opportunistic routing," in *ACM SigComm*, 2007.

[6] B. Radunovic, C. Gkantsidis, P. Key, S. Gheorgiu, W. Hu, and P. Rodriguez, "Multipath code casting for wireless mesh networks," in *MSR-TR-2007-68*, March 2007.

[7] L. Georgiadis, M. J. Neely, and L. Tassiulas, "Resource allocation and cross-layer control in wireless networks," *Foundations and Trends in Networking*, vol. 1, no. 1, pp. 1–144, 2006.

[8] R. Ahlswede, N. Cai, S. R. Li, and R. W. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, 2000.

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

[10] D. Lun, M. Medard, and R. Koetter, "Network coding for efficient wireless unicast," in *IEEE International Zurich Seminar on Communications*, February 2006.

[11] R. Srikant, *The Mathematics of Internet Congestion Control*. Birkhauser, 2003.

[12] A. Eryilmaz and R. Srikant, "Joint congestion control, routing and mac for stability and fairness in wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 8, pp. 1514–1524, August 2006.

[13] X. Lin, N. B. Shroff, and R. Srikant, "A tutorial on cross-layer optimization in wireless networks," *IEEE J. on Selected Areas in Comm.*, vol. 24, no. 8, Aug 2006.

[14] M. Chen, S. Low, M. Chiang, and J. Doyle, "Cross-layer congestion control, routing and scheduling design in ad hoc wireless networks," in *INFOCOM*, 2006.

[15] G. Sharma, R. Mazumdar, and N. Shroff, "On the complexity of scheduling in wireless networks," in *Proc. MOBICOM*, 2006.

[16] D. Lun, M. Medard, and M. Effros, "On coding for reliable communication over packet networks," in *Proc. 42nd Allerton Conference*, 2004.

[17] 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*, vol. 37, no. 12, 1992.

[18] M. Neely, E. Modiano, and C. Rohrs, "Dynamic power allocation and routing for time-varying wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 1, pp. 89–103, January 2005.

[19] M. Neely, "Optimal backpressure routing for wireless networks with multi-receiver diversity," in *Conference on Information Science and Systems*, March 2006.

[20] D. Bertsekas and J. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Prentice Hall, 1989.

[21] B. Radunovic, C. Gkantsidis, G. Gunawardena, and P. Key, "Horizon: Balancing tcp over multiple paths in wireless mesh network," in *Proc. MOBICOM*, 2008.

[22] F. P. Kelly, A. Maulloo, and D. Tan, "Rate control in communication networks: shadow prices, proportional fairness and stability," *Journal of the Operational Research Society*, vol. 49, pp. 237–252, 1998.

[23] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[24] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, "MORE: A network coding approach to opportunistic routing," in *MIT-CSAIL-TR-2006-049*, 2006.

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

[26] E. Modiano, D. Shah, and G. Zussman, "Maximizing throughput in wireless networks via gossiping," in *Proc. ACM SIGMETRICS / IFIP Performance'06*, June 2006.

[27] A. K. Miu, H. Balakrishnan, and C. E. Koksal, "Improving Loss Resilience with Multi-Radio Diversity in Wireless Networks," in *11th ACM MOBICOM Conference*, Cologne, Germany, September 2005.

[28] R. Gummadi, R. Patra, H. Balakrishnan, and E. Brewer, "Interference Avoidance and Control," in *7th ACM Workshop on Hot Topics in Networks (Hotnets-VII)*, Calgary, Canada, October 2008.

[29] L. Huang, B. Johnson, D. Tadas, and M. Stoler, "802.11a performance over various channels," in *IEEE 802.11-03/0682-00-000k*, 2003.

[30] T. Cui, L. Chen, and T. Ho, "Effcient opportunistic network coding for wireless networks," in *Proc. INFOCOM*, 2008.

[31] K. Zeng, W. Lou, and H. Zhai, "On end-to-end throughput of opportunistic routing in multirate and multihop wireless networks," in *INFOCOM*, 2008.

[32] 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.

[33] A. Eryilmaz and D. Lun, "Control for inter-session network coding," in *NetCod*, 2007.

APPENDIX

Before proving Theorem 2 we first introduce the following lemma

Lemma 1: The following equalities and inequalities hold for any t:

$$\sum_{i,j\in\mathcal{N}}\sum_{c\in\mathcal{C}} y_{ij}^{c*}(\mu_i^{c*}-\mu_j^{c*}) = \sum_{c\in\mathcal{C}} f_c^*\mu_i^{c*}, \tag{37}$$

$$\xi_{iJ}^{c*}x_{iJ}^{c*} = \xi_{iJ}^{c*}\sum_{j\in J} y_{ij}^{c*}, \tag{38}$$

$$\sum_{i\in\mathcal{N},J\in\mathcal{P}(\mathcal{N})}\sum_{c\in\mathcal{C}}\xi_{iJ}^{c*}x_{iJ}^{c}(t) \le \sum_{i\in\mathcal{N},J\in\mathcal{P}(\mathcal{N})}\sum_{c\in\mathcal{C}}\xi_{iJ}^{c*}x_{iJ}^{c*}, \tag{39}$$

$$\sum_{i\in\mathcal{N},J\in\mathcal{P}(\mathcal{N})}\sum_{c\in\mathcal{C}}\xi_{iJ}^{c*}x_{iJ}^{c}(t) \le \sum_{i,j\in\mathcal{N},c\in\mathcal{C}}\xi_{ij}^{c*}y_{ij}^{c*}, \tag{40}$$

$$\sum_{i,j\in\mathcal{N},c\in\mathcal{C}} y_{ij}^{c*}(q_i^c(t)-q_j^c(t)) = \sum_{c\in\mathcal{C}} f_c^* q_{\mathrm{Src}(c)}^c(t), \tag{41}$$

$$\sum_{i\in\mathcal{N},J\in\mathcal{P}(\mathcal{N})} w_{iJ}^c(t)\sum_{j\in J} y_{ij}^c(t) = \sum_{i,j\in\mathcal{N}} w_{ij}^c(t)y_{ij}^c(t), \tag{42}$$

$$\sum_{i\in\mathcal{N},J\in\mathcal{P}(\mathcal{N})}\sum_{c\in\mathcal{C}} w_{iJ}^c(t)x_{iJ}^c(t) \ge \sum_{i,j\in\mathcal{N},c\in\mathcal{C}} w_{ij}^c(t)y_{ij}^{c*} \tag{43}$$

Proof: Equalities (37) and (38) follow directly from (6) and (7). From (4) we further have $\sum_{c\in\mathcal{C}} x_{iJ}^c(t) = C_{iJ}(t)$;

Together with (10) this implies

$$\sum_{i,J,c}\xi_{iJ}^{c*}x_{iJ}^{c}(t)\leq\sum_{i,J,c}\left(\sum_{J}\xi_{iJ}^{c*}C_{iJ}(t)\right)\leq\sum_{i,J,c}\left(\sum_{J}\xi_{iJ}^{c*}C_{iJ}^{*}\right)=\sum_{i,J,c}\xi_{iJ}^{c*}x_{iJ}^{c*}$$

which proves (39). From (38) and (39) we derive (40). Since by definition $q_{\mathrm{Dst}(c)}^{c}(t)=0$ we have

$$\sum_{i,j\in\mathcal{N},c\in\mathcal{C}}y_{ij}^{c*}(t)(q_{i}^{c}(t)-q_{j}^{c}(t))=\sum_{c\in\mathcal{C},j\in\mathcal{N}}y_{\mathrm{Src}(c)j}^{c*}q_{\mathrm{Src}(c)}^{c}(t)$$

from which we derive (41). Equality (42) follows from the definition of w_{ij}^{c}. Also, by definition of scheduling (13)-(18), we have

$$\sum_{i,J,c}w_{iJ}^{c}(t)x_{iJ}^{c}(t)\geq\sum_{i}\max_{c}\sum_{J}w_{iJ}^{c}(t)C_{iJ}(\forall C_{iJ})\quad(44)$$

$$\geq\sum_{i,J,c}w_{iJ}^{c}(t)x_{iJ}^{c*}\quad(45)$$

which yields (43). ∎

A. Proof of Theorem 2

Proof: We will follow the idea of the proof of Theorem 2 from [12]. First, let us define Lyapunov function

$$W(\mathbf{f},\mathbf{q},\mathbf{w})=\frac{1}{2\gamma}\sum_{c\in\mathcal{C}}(f_{c}-f_{c}^{*})^{2}+\frac{1}{2}\sum_{i\in\mathcal{N},c\in\mathcal{C}}(q_{i}^{c}-\mu_{i}^{c*})^{2}+\frac{1}{2}\sum_{i\in\mathcal{N},J\in\mathcal{P}(\mathcal{N})}\sum_{c\in\mathcal{C}}(w_{iJ}^{c}-\xi_{iJ}^{c*})^{2}.\quad(46)$$

We want to show that the derivative $\dot{W}\leq 0$. For brevity, we define $f_{i}^{c}(t)=f_{c}(t)1_{\{i=\mathrm{Src}(c)\}}$. Using (20), (21) and (22) gives derivative of W as

$$\begin{aligned}&\dot{W}(\mathbf{f}(t),\mathbf{q}(t),\mathbf{w}(t))\\&\quad=\sum_{c}(f_{c}(t)-f_{c}^{*})(U_{c}'(f_{c}(t))-q_{\mathrm{Src}(c)}^{c}(t))_{f_{c}(t)\geq 0}\\&+\sum_{i,c}(q_{i}^{c}(t)-\mu_{i}^{c*})\left(f_{i}^{c}(t)+\sum_{j}y_{ji}^{c}(t)-\sum_{k}y_{ik}^{c}(t)\right)_{q_{i}^{c}(t)\geq 0}\\&+\sum_{i,J,c}(w_{iJ}^{c}(t)-\xi_{iJ}^{c*})\left(\sum_{j\in J}y_{ij}^{c}(t)-x_{iJ}^{c}(t)\right)_{w_{iJ}^{c}(t)\geq 0}.\quad(47)\end{aligned}$$

As in (10)-(13) from [12] we have that, when $q_{i}^{c}(t)<0$ the derivative $\dot{q}_{i}^{c}(t)$ is by definition positive; also $\mu_{i}^{c*}\geq 0$. The same holds for $w_{iJ}^{c}(t)$ and $f_{c}(t)$ and we can upperbound the derivative

$$\begin{aligned}\dot{W}(\mathbf{f}(t),\mathbf{q}(t),\mathbf{w}(t))&\leq\sum_{c}(f_{c}(t)-f_{c}^{*})(U_{c}'(f_{c}(t))-q_{\mathrm{Src}(c)}^{c}(t))\\&+\sum_{i,c}(q_{i}^{c}(t)-\mu_{i}^{c*})\left(f_{i}^{c}(t)+\sum_{j}y_{ji}^{c}(t)-\sum_{k}y_{ik}^{c}(t)\right)\\&+\sum_{i,J,c}(w_{iJ}^{c}(t)-\xi_{iJ}^{c*})\left(\sum_{j\in J}y_{ij}^{c}(t)-x_{iJ}^{c}(t)\right)\quad(48)\end{aligned}$$

Let us further add and subtract $U_{c}'(f_{c}^{*})=\mu_{\mathrm{Src}(c)}^{c*}$, as in [12], to obtain

$$\begin{aligned}\dot{W}(\mathbf{f}(t),\mathbf{q}(t),\mathbf{w}(t))&\leq\sum_{c}(f_{c}(t)-f_{c}^{*})(U_{c}'(f_{c}(t))-U_{c}'(f_{c}^{*}))\\&+\sum_{i,c}\mu_{i}^{c*}\left(\sum_{k}y_{ik}^{c}(t)-\sum_{j}y_{ji}^{c}(t)-f_{i}^{c*}\right)\\&+\sum_{i,c}q_{i}^{c}(t)\left(\sum_{j}y_{ji}^{c}(t)-\sum_{k}y_{ik}^{c}(t)+f_{i}^{c*}\right)\\&+\sum_{i,J,c}(w_{iJ}^{c}(t)-\xi_{iJ}^{c*})\left(\sum_{j\in J}y_{ij}^{c}(t)-x_{iJ}^{c}(t)\right).\end{aligned}$$

Due to concavity of U_{c} we have $(f_{c}(t)-f_{c}^{*})(U_{c}'(f_{c}(t))-U_{c}'(f_{c}^{*}))\leq 0$. Next, let us pick any set of link rates $\{y_{ij}^{c*}\}_{i,j,c}$ that correspond to the optimal flow allocation $\{f_{c}^{*}\}_{c}$. We expand f_{c}^{*} using (37) and (41) to obtain

$$\begin{aligned}\dot{W}(\mathbf{f}(t),\mathbf{q}(t),\mathbf{w}(t))&\leq\sum_{i,j,c}(\mu_{i}^{c*}-\mu_{j}^{c*})(y_{ij}^{c}(t)-y_{ij}^{c*})\\&+\sum_{i,j,c}(q_{i}^{c}-q_{j}^{c*})(y_{ij}^{c*}-y_{ij}^{c}(t))\\&+\sum_{i,J,c}(w_{iJ}^{c}(t)-\xi_{iJ}^{c*})\left(\sum_{j\in J}y_{ij}^{c}(t)-x_{iJ}^{c}(t)\right).\end{aligned}$$

Let us denote $z_{ij}^{c*}=\mu_{i}^{c*}-\mu_{j}^{c*}-\xi_{ij}^{c*}$. Then, from (40), (42) and (43) we have

$$\begin{aligned}&\dot{W}(\mathbf{f}(t),\mathbf{q}(t),\mathbf{w}(t))\\&\leq\sum_{i,j,c}(y_{ij}^{c}(t)-y_{ij}^{c*})\times\\&\quad\left[(\mu_{i}^{c*}-\mu_{j}^{c*}-\xi_{ij}^{c*})-(q_{i}^{c}(t)-q_{j}^{c}(t)-w_{ij}^{c}(t))\right]\quad(49)\\&=\sum_{i,j,c}(y_{ij}^{c}(t)-y_{ij}^{c*})(z_{ij}^{c*}-z_{ij}^{c}(t))\quad(50)\\&\overset{(a)}{=}\sum_{i,j,c}y_{ij}^{c}(t)z_{ij}^{c*}-(y_{ij}^{c}(t)-y_{ij}^{c*})z_{ij}^{c}(t)\quad(51)\\&\overset{(b)}{\leq}-\sum_{i,j,c}(y_{ij}^{c}(t)-y_{ij}^{c*})z_{ij}^{c}(t)\overset{(c)}{\leq}0.\quad(52)\end{aligned}$$

where (a) follows from KKT and the fact that $y_{ij}^{c*}z_{ij}^{c*}=0$, (b) from (9) and (c) from the fact that whenever $q_{i}^{c}(t)>0$ and $z_{ij}^{c}(t)>0$ then $y_{ij}^{c}(t)$ can be made arbitrarily large in the fluid model limit as the slot length goes to zero.

Hence we have that $\dot{W}(\mathbf{f}(t),\mathbf{q}(t),\mathbf{w}(t)) \le 0$ for all $\mathbf{f}(t) > 0, \mathbf{q}(t) > 0, \mathbf{w}(t) \ge 0$. Let us define

$$\mathcal{Q}(t) = \left\{ (\mathbf{q},\mathbf{w}) \,|\, \sum_{i,j,c} (y_{ij}^c(t) - y_{ij}^{c*})(z_{ij}^{c*} - z_{ij}^c(t)) \right\} \quad (53)$$

Let us define $\mathcal{E} = \{(\mathbf{f},\mathbf{q},\mathbf{w}) \,|\, \dot{W}(\mathbf{f},\mathbf{q},\mathbf{w}) = 0\}$. It is easy to see from (50) that $\mathcal{E} \subseteq \mathcal{Q}$. We can further apply LaSalle's invariance principle as in [12] to show that $\mathbf{f}(t)$ converges to $\mathbf{f}^*$ and $(\mathbf{q}(t),\mathbf{w}(t))$ converges to $\mathcal{Q}(t)$.

However, set $\mathcal{Q}(t)$ is not bounded in general. If link (i,j) is active for flow c then for every $t, y_{ij}^c(t) > 0$ we have $q_j^c(t) + w_{ij}^c(t) = q_i^c(t) - z_{ij}^{c*}$. If the maximum node degree in a network is D, we have that $q_j^c(t) + w_{ij}^c(t) \le q_i^c(t) - z_{ij}^{c*} + 2DT$. Since $q_{\mathrm{Src}(c)}^c$ converges to $U_c'(f_c^*)$ we see that queues $q_i^c(t), q_j^c(t)$ and $w_{ij}^c(t)$ are bounded for all active links (i,j) of each flow c. ∎

B. Derivation of (33)-(35)

Let us write modified queue evolution equations:

$$\dot{q}_i^c(t,g) = \left(f_i^c(t,g) + \sum_j y_{ji}^c(t,g) - \sum_j y_{ij}^c(t,g) \right)_{q_i^c(t,g) \ge 0}$$

$$\dot{w}_{iJ}^c(t,g) = \left(\sum_{j \in J} y_{ij}^c(t,g) - x_{iJ}^c(t,g) \right)_{w_{iJ}^c(t,g) \ge 0}.$$

Note that (20) and (21) do not hold anymore. Consequently, we cannot claim that (48) follows from (47) and the proof of Theorem 2 cannot be applied.

We first consider $\dot{q}_i^c(t,g)$. We see from (33) that $y_{ik}^c(t,g) > 0$ only if $q_i^c(t,g) > 0$. Thus, the exact queue evolution is given by

$$\dot{q}_i^c(t,g) = f_i^c(t,g) + \sum_j y_{ji}^c(t) - \sum_k y_{ik}^c(t).$$

Next, let us look at the evolution of $w_{iJ}^c(t,g)$. We have that $\dot{w}_{iJ}^c(t,g) = 0$ only if $w_{iJ}^c(t,g) = 0$ and $x_{iJ}^c(t,g) > 0$, thus if $g = g^*(c,t)$. Therefore, we can write

$$\dot{w}_{iJ}^c(t,g) \ge \sum_{j \in J} y_{ij}^c(t) - x_{iJ}^c(t,g)\, 1_{\{w_{iJ}^c(t,g) \ge 0\}},$$

$$\dot{w}_{iJ}^c(t) \ge \sum_{j \in J} y_{ij}^c(t) - x_{iJ}^c(t)\, 1_{\{w_{iJ}^c(t,g^*(c,t)) \ge 0\}}.$$

and from (47) we can write

$$\dot{W}(\mathbf{f}(t),\mathbf{q}(t),\mathbf{w}(t))$$

$$\le \sum_c (f_c(t) - f_c^*)(U_c'(f_c(t)) - q_{\mathrm{Src}(c)}^c(t)) \quad (54)$$

$$+ \sum_{i,c} (q_i^c(t) - \mu_i^{c*}) \left(f_i^c(t) + \sum_j y_{ji}^c(t) - \sum_k y_{ik}^c(t) \right) \quad (55)$$

$$+ \sum_{i,J,c} (w_{iJ}^c(t) - \xi_{iJ}^{c*}) \left(\sum_{j \in J} y_{ij}^c(t) - x_{iJ}^c(t) \right) \quad (56)$$

$$+ \sum_{i,J,c} w_{iJ}^c(t) x_{iJ}^c(t)\, 1_{\{w_{iJ}^c(t,g^*(c,t)) < 0\}}. \quad (57)$$

Intuitively (57) means that if we decide to transmit generation $g^*(c,t)$, we will not remove any credit from queues for which there is no such generation queued (that is $w_{iJ}^c(t,g^*(c,t)) < 0$). Note that this cannot happen with infinite generation sizes as $w_{iJ}^c(t,g^*(c,t)) < 0$ implies $w_{iJ}^c(t) < 0$. Since we have already proven in Theorem 2 that (54) + (55) + (56) ≤ 0, we want to minimize (57), which is equivallent to maximization in (34) and (35) (with a slight caveat that the condition $w_{iJ}^c(t,g^*(c,t)) < 0$ in the fluid model reads as $w_{iJ}^c(t,g^*(c,t)) = 0$ in the packet model, due to continuity). It is also easy to see from (57) that the naive policy (31) may yield an unbounded drift.

Božidar Radunović is a Researcher in the Systems and Networking group at Microsoft Research, Cambridge. Božidar received his PhD in technical sciences from EPFL, Switzerland, in 2005, and his BSc at the School of Electrical Engineering, University of Belgrade, Serbia, in 1999. He was a PhD student at LCA, EPFL from 2000-2005. Then spent one year at TREC, at ENS Paris, in 2006. In 2008 he has been awarded IEEE William R. Bennett Prize Paper Award in the Field of Communications Systems.

Christos Gkantsidis is a researcher in the the Systems and Networking Group of Microsoft Research at Cambridge, UK. Christos did his Ph.D. at the College of Computing at Georgia Tech, Atlanta, GA, USA. under the supervision of Prof. Milena Mihail. He did his undergrad at the Computer Engineering and Informatics Department of the University of Patras, Rio, Greece. He has also worked at Sprint Labs, CA, USA and in the Computer Technology Institute, Patras, Greece.

Peter Key is a principal researcher in the the Systems and Networking Group of Microsoft Research at Cambridge, UK. Peter graduated in 1978 from Oxford University with a BA in Mathematics. Following an Msc in Statistics from University College, London in 1979, he was a Research Assistant in the Statistics and Computer Science department of Royal Holloway College, London University until 1982. He was awarded a PhD in 1985 with a thesis on Bayesian forecasting. He is a fellow of the Royal Statistical Society.

Pablo Rodriguez is the Scientific Director of the Internet systems and networking group at the Telefonica Research Lab in Barcelona. Prior to Telefonica, he was a researcher at Microsoft Research, Cambridge. Pablo have also worked as a Member of Technical Staff at Bell Labs (NJ, USA) and as a software architect for various startups in the Silicon Valley including Inktomi (acquired by Yahoo!) and Tahoe Networks (now part of Nokia). He received his Ph.D. from the Swiss Federal Institute of Technology (EPFL, Lausanne) while working at Institut Eurécom, (Sophia Antipolis, France).