An Optimization Framework for Practical Multipath Routing in Wireless Mesh Networks

Božidar Radunović[1], Christos Gkantsidis[1], Peter Key[1], Pablo Rodriguez[2], and Wenjun Hu[3]

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

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

[3] Cambridge University
Cambridge, UK
wenjun.hu@cl.cam.ac.uk

June 2007

Technical Report
MSR-TR-2007-81

Microsoft Research
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052
http://www.research.microsoft.com

Abstract

We consider wireless mesh networks, and exploit the inherent broadcast nature of wireless by making use of multipath routing. We present an optimization framework that enables us to derive optimal flow control, routing, scheduling, and rate adaptation schemes, where we use network coding to ease the routing problem. 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 random scheduling rather than optimal for comparisons. Under random scheduling, our protocol becomes fully decentralised. We use simulation to show on realistic topologies that we can achieve 20-200% throughput improvement compared to single path routing, and several times compared to a recent related opportunistic protocol (MORE).

1 Introduction

One of the main challenges in building wireless mesh networks ([1, 2, 3]) is to guarantee high performance. The difficulty is mainly caused by the unpredictable and highly-variable nature of the wireless channel. However, the use of wireless channels presents some unique opportunities that can be used to improve the performance. For example, the broadcast nature of the medium can be used to provide opportunistic transmissions as suggested in [4]. Also, in wireless mesh networks, there are typically multiple paths connecting each source destination pair; using some of these paths in parallel can improve performance [5, 6]. The optimal use of multiple paths and of opportunistic transmissions is the main focus of this work. We use network coding [7] to simplify the problem of scheduling packet transmissions across multiple paths, similarly to [5, 6, 8]. 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 the wireless mesh network efficiently, the system needs to take into account: (a) the existence of multiple paths, (b) the unreliable nature of the 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) the competition among many flows, (f) fairness and efficiency. Observe that optimizing across all these parameters results in 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 optimize all these parameters simultaneously 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 framework borrows heavily from [9, 8]. Our algorithm is a primal-dual algorithm [10]. 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 [11, 12]. 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. [11]). We propose a novel algorithm for cross-layer optimization and prove, using Lyapunov functions, that it converges to the optimal rate allocation.

The proposed optimization framework is difficult to implement; indeed, the joint scheduling, rate and power control problem is NP-hard [13]. 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 can be implemented in practical systems. The fundamental idea of our algorithm (and, of the distributed implementation) is to assign credits to nodes, transfer credits between nodes, and schedule on the basis of credits (see Sec. 3 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 2 and 3). We introduce a notion of virtual packets, called credits, that enable us to decouple routing and flow control from actual packet transmissions. We prove the optimality of the algorithm.

• Based on the optimization algorithm, we give a distributed implementation of routing, rate adaptation, and flow control for networks with random scheduling (Section 3.3) that outperforms existing algorithms. We prove that our algorithms extends and outperforms MORE [5]. The distributed algorithm can be used with the current 802.11 MAC, and indeed is MAC independent. We also show how it can be used with practical network coding schemes with finite generation sizes (Section 4.1).

• We demonstrate that rate selection is important for optimizing performance in 802.11a networks (Section 4.2). 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 5).[1]

2 Model

In this section we introduce the notation used in the paper. We extend the model of wireless erasure network developed in [14] to include multiple flows. Vectors are denoted in bold.

2.1 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 $J \subseteq \mathcal{N}$ with a hyperarc (i, J). 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. We denote by $\mathcal{S}$ the set of feasible activation profiles and let $\mathrm{SRC}(S) = \{i \in \mathcal{N} \mid \exists J \subseteq \mathcal{N}, (i, J) \in S\}$ be the set of transmitters in activation profile S.

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

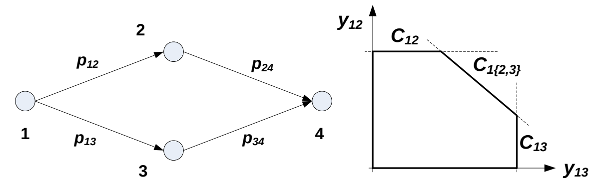

Figure 1: A network with 4 nodes is given on the left. For example, an activation profile $\{(1, 2), (3, 4)\}$ depicts a profile where nodes 1 and 3 are transmitting, nodes 2 is receiving a packet from node 1, while node 4 is receiving from node 3. Profile $\{(1, \{2, 3\})\}$ depicts node 1 transmitting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

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 J, 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 $\mathbf{P} = (P_i)_{i \in \mathcal{N}}$ to these transmissions. Let $T_{ij} = 1$ if a packet is successfully transmitted from i to $j \in J$. We define $p_{ij}(\mathbf{P}, R_i, S) = \mathrm{Prob}(T_{ij}(\mathbf{P}, R_i, S) = 1)$ to be the probability that node $j \in J$ will successfully receive the packet from i, given the above conditions. We also assume that T_{ij} and T_{lk} are independent for $i \neq l$ or $j \neq k$, which is justified by measurements (c.f. [15]). By convention, we assume $p_{ij}(\mathbf{P}, R_i, S) = 0$ if $j \notin J$, for $(i, J) \in S$.

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

$$C_{iK}(\mathbf{P}, R_i, S) = R_i \left(1 - \prod_{j \in K} (1 - p_{ij}(\mathbf{P}, R_i, S)) \right). \tag{1}$$

Note that $C_{iK}(\mathbf{P}, R_i, S) = 0$ if $K \cup J = \emptyset$, for $(i, J) \in S$.

2.2 Traffic, Routing 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. Each flows may take an arbitrary route from a source to a destination, subject to constraints imposed by the set of activation profiles. We denote by f_c the rate of flow $c \in \mathcal{C}$. One flow may take multiple routes to the destination. We assume routing is done at each hop. Each relay will decide how much traffic from a flow it will forward to other nodes. This decision is made through credit assignment, as described next in Section 2.4.

Whenever a node is active, 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 $A_{ic} = 1$. Let $\mathcal{A}$ be 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 equals $A_{1c}(C_{12}(\mathbf{P}, R_i, S) + C_{1\{2,3\}}(\mathbf{P}, R_i, S))$ and it depends not only on how often $(1, 2)$ is scheduled, but also on how often $(1, \{2, 3\})$ is scheduled. This is why the flow scheduling decision is assigned to a node, instead to a link as in [16, 17, 11].

2.3 Network Coding

We assume network coding per flow is used [14, 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 forward the same packets [4]. To eliminate this problem, each relay forwards a random linear combination of all previously received packets from the same flow.

Ideally, network coding should be performed at the scale of the whole 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 [14, 5]. In Section 3 we analyze networks with very large generation sizes (as in [14]) and we address the finite generation sizes in Section 4.

2.4 Credits

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 [18, 5]). We will use a concept of credits, which is similar to the control decision variable of Neely [18].

Credits are created for each packet at the source node, and identified with a generation, not a specific packet. They are interpreted as the number of packets of a specific flow to be transferred by the node. Credits are conserved until they arrive at the flow's destination.

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. Our results can be extended to consider imperfect signaling (cf. [12]).

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. 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.

2.5 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 $R_i(t)$. We further denote by $y_{ij}^c(t)$ the number of credits for flow c transmitted from node i to node j during slot t, and with $x_{iJ}^c(t)$ the number of packets of flow c actually transmitted from i to any of the nodes in J during slot t. Let $f_c(t)$ be the number of fresh packets/credits generated at the source of flow c.

Note that, because each packet transmission is always associated to a credit transmissions, 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 will define stability more formally in Section 3.4.

3 Optimal Flow Control For Fairness

In this Section we introduce the optimization problem (Sec. 3.2), propose an algorithm for solving it (Sec. 3.3), and prove that the algorithm converges (Sec. 3.4). Sec.3.1 introduces some further notation that is needed for the description of the optimization problem. Finally, in Sec. 3.5 we compare it theoretically with the MORE algorithm proposed in [5].

3.1 Feasible Rate Set

Assume an assignment of end-to-end rates f_c, for each flow c, and denote the rate vector by $\mathbf{f} = (f_c)_{c\in\mathcal{C}}$. The vector of rates is valid under the following three conditions. First, by conservation of the flow of credits at each node $i \neq \mathrm{Dst}(c)$:

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

Also, $y_{ij} \geq 0$. Second, due to the constraints of the broadcast regions (see also Fig. 1):

$$\sum_{j\in J} y_{ij}^c \leq x_{iJ}^c \qquad (3)$$

Assume now that variables $\alpha_{S,\mathbf{R},\mathbf{P},\mathbf{A}}$ define a schedule and denote a fraction of time 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 third condition comes from scheduling constraints:

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

Note that (as explained in Section 2.2), although (4) implies $\{\sum_c x_{iJ}^c\}_{i,J}$ belongs to $\mathrm{Hull}(\{C_{iJ}(\mathbf{P}, R_i, S)\}_{i,J})$, the converse is not true.

We will use the following characterization of feasible rates from [8]:

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 rate* f_c.

Theorem 1. *Let* $\mathcal{F}$ *be the set of 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. Follows directly from [8]. □

3.2 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 [19] and $U_c(f_c) \propto -1/f_c$ approximates TCP's utility [10]. The goal of utility maximization is to achieve trade-off between efficiency and fairness. A typical example of such approach is proportional fairness [19].

We can write the network-wide optimization problem as

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

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 $\mu^c_{\mathrm{Dst}(c)} = 0$. We can write the KKT conditions at the optimal point

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

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

$$f_c^*\left(U_c'(f_c^*) - \mu^{c*}_{\mathrm{Src}(c)}\right) = 0, \quad (8)$$

We see that μ_i^{c*} can be positive only if more traffic of flow c comes into node i than leaves it. 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 3.3 we will express this relationship more formally. We will also use (8) to develop a flow control algorithm.

As a consequence of KKT, using some elementary algebra one can derive

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

$$C_{iJ}^* = \underset{C_{iJ}\in\{C_{iJ}(\mathbf{P},R_i,S)\}}{\mathrm{argmax}} \sum_i \max_c \sum_J \xi_{iJ}^{c*} C_{iJ} \quad (10)$$

Notably we will use (10) in Section 3.3 to derive the optimal scheduling.

3.3 Maximization Algorithm

We next present an algorithm that converges to the optimal value of (5). In the following we assume that the feedback is ideal, hence that the acknowledgments and credits are transmitted instantaneously and without errors. We leave the analysis of signaling with losses and delays for future work.

Node and Transport Credits : Recall that $q_i^c(t)$ is the amount of credits of commodity c queued at node i. We call these credits *node credits*. In addition, let $w_{iJ}^c(t)$ be the number of credits of commodity c queued at i and corresponding to the packets that have to be delivered to *any* of the nodes in J (as previously decided by the credit transmission scheme). We call these credit *transport credits*. When a credit for flow c is passed 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 of them by one unit). We decrease w_{iJ}^c when a packet from flow c is actually transmitted from i to any of the nodes in J.

Routing protocol: Node credits represent intentions of packet transmissions and a routing protocol describes when and how are 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

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

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). A credit is routed from i to j only if the back-pressure is positive:

$$y_{ij}^c(t) = M\, 1_{\{z_{ij}^c(t)>0\}}, \quad (11)$$

where $1_{\{x>0\}}$ is 1 if $x > 0$ or 0 otherwise. In Section 3.4 we will derive conditions on M to guarantee convergence of the algorithm.

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

$$\bar{w}_i(t,\mathbf{P},R_i,S) = \max_c \sum_J w^c_{iJ}(t) C_{iJ}(\mathbf{P},R_i,S) \quad (12)$$

$$(S(t),\mathbf{P}(t),\mathbf{R}(t)) = \underset{S,\mathbf{P},\mathbf{R}}{\text{argmax}} \sum_{i\in\mathcal{N}} \bar{w}_i(t,\mathbf{P},R_i,S), \quad (13)$$

$$C_{iJ}(t) = C_{iJ}(\mathbf{P}(t),R_i(t),S(t)), \quad (14)$$

$$c^*_i(t) = \underset{c}{\text{argmax}} \sum_K w^c_{iK}(t) C_{iK}(t), \quad (15)$$

$$A_{ic}(t) = 1_{\{c=c^*_i(t)\}}, \quad (16)$$

$$x^c_{iJ}(t) = A_{ic} C_{iJ}(t) \quad (17)$$

Equations (12)-(17) 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 (13). Then, equation (15) is used to select which flow will be transmitted by each node in slot t. Note that, as explained in Section 2.2, we cannot decouple the flow selection process $A(t)$ and routing/scheduling/rate/power control as it was done in similar approaches that do not use opportunistic routing (e.g. [16, 17, 11]). Also, unlike in [16, 17, 11, 18], we do not explicitly use back-pressure information for scheduling in (12)-(17); instead we use transmission credits $w^c_{iJ}(t)$.

Observe that all equations except (13) use local information only. Hence, with the exception of (13) (assuming that we could somehow compute the values $C_{iJ}(t)$), the problem could have been solved with a distributed algorithm. Recall from Sec. 2.1 that $C_{iJ}(t)$ relate to channel scheduling. If channel scheduling was determined by the MAC protocol, as is typical in most current wireless technologies (where nodes randomly compete for the wireless channel), then the quantities $C_{iJ}(t)$ are explicitly known and the rest of the optimization problem can be computing using local computations. Of course, with random scheduling it is not possible to achieve optimal performance, but, on the other hand, we can implement the optimization framework with a distributed algorithm and still observe performance benefits (as we shall see in Sec. 5).

Flow control: The optimal flow rate at the source, $f^c(t)$ can be calculated using a primal-dual approach, as in [11]

$$f_c(t+1) = \left[f_c(t) + \gamma\left(U'_c(f_c(t)) - q^c_{\text{Src}(c)}(t)\right)\right]^+, \quad (18)$$

where $[x]^+ = \max\{x,0\}$. Each flow adapts its rate based on the previous rate and current number of credits queing for transmission at the source node for that flow ($q_{\text{Src}(c)}$). The primal-dual approach well describes additive-increase multiplicative-decrease transport protocols, like TCP [10].

3.4 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 [11].

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

$$\dot{q}^c_i(t) = \left(f_c(t)1_{i=\text{Src}(c)} + \sum_j y^c_{ji}(t) - \sum_j y^c_{ij}(t)\right)_{q^c_i(t)\geq 0} \quad (19)$$

$$\dot{w}^c_{iJ}(t) = \left(\sum_{j\in J} y^c_{ij}(t) - x^c_{iJ}(t)\right)_{w^c_{iJ}(t)\geq 0} \quad (20)$$

where $(x)_{y>z}$ equals x is $y>z$ or 0 otherwise. Similarly, flow rate evolution in the fluid-model is given by

$$\dot{f}_c(t) = \gamma\left(U'_c(f_c(t)) - q_{\text{Src}(c)c}\right)_{f_c(t)\geq 0}. \quad (21)$$

We next prove that the algorithm presented in Section 3.3 stabilizes the system with flow rates that maximize the optimization problem (5). Let us first define an active link

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 $y^c_{ij}(t)>0$, there exists $t', t<t'<t+T$ such that $y^c_{ij}(t')>0$.*

Intuitively, 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 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 active links (and the corresponding nodes) whose average traffic is at least M/T. We then have:

Theorem 2. *If $M > \max_c f_c^*$ then, starting from any vectors $\mathbf{f}(0), \mathbf{q}(0), \mathbf{w}(0)$, 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 .*

Proof. 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 hence the system is stable. The details are given in the appendix. □

3.5 Comparison with MORE

In this section we compare the performance of our algorithm with the MORE algorithm presented in [5, 20]. Our algorithm converges to the optimal solution of the optimization problem (5) (as shown in Theorem 2), hence MORE at best is as good as our algorithm. We first show under what conditions MORE is guaranteed to give the optimal solution. We then also 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 $|\mathrm{SRC}(S)| = 1$ 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 $\sum_{j \in J} y_{ij} \le \alpha_i C_{iJ}$. We also omit c as there is only one flow in the system.

We start with the MORE optimization problem, as given in [20], and we introduce $f = 1/\left(\sum_{i \in \mathcal{N}} z_i\right)$, $y_{ij} = f\, x_{ij}$ and $\alpha_i = f\, z_i$. The optimization from [20, Eq.(1)-(4)] is then equivalent to

$$\min \quad 1/f \tag{22}$$

$$s.t. \quad \sum_{j \in \mathcal{N}} y_{ij} - \sum_{j \in \mathcal{N}} y_{ji} = f 1_{\{i=\mathrm{Src}\}} \tag{23}$$

$$\alpha_i C_{iJ} \ge \sum_{j \in J} y_{ij}, \tag{24}$$

$$\sum_{i \in \mathcal{N}} \alpha_i = 1, \tag{25}$$

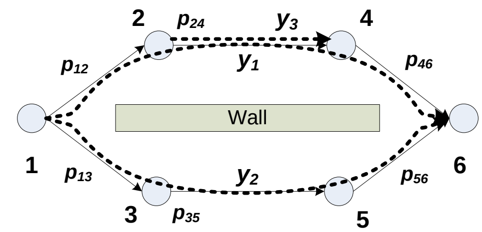

Figure 2: A network with 6 nodes. Due to a wall in between, nodes 2 and 4 do not interfere with nodes 3 and 5. The set of activation profiles is $\mathcal{S} = \{\{(1,\{2,3\})\}, \{(2,3)\}, \{(3,5)\}, \{(4,6)\}, \{(5,6)\}, \{(2,4),(3,5)\}, \{(2,4),(5,6)\}, \{(3,5),(4,6)\}\}$. There are two flows in the 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 ($y_3 = y_{2-4}$).

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 2. Let us first consider a case with a single flow f_1, ($f_2 = 0$), and in which $p_{12} = p_{24} = p_{46} = 0.8$ 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 $f_{\mathrm{MORE}} = 0.267$, 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 both flows active. Since MORE algorithm 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 transport layer on the 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.

4 Practical Issues

In this section we consider two practical issues that concern implementation of the protocol proposed in Section 3 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.

4.1 Finite Generation Size

Previous results assume that generation size used for network coding tends to infinity (see Theorem 1). Practical reasons, such as 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 packets [5, 6]. We next modify our optimization framework to include finite generation size.

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 credits and transmission 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, only constrained on 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 3.3 to this setting is not straightforward. For example, a naive way to modify scheduling rule (17) would be to schedule the oldest generation available

$$g_i^*(c,t) = \min\{g \,|\, (\exists J) w_{iJ}^c(t,g) > 0\}, \tag{26}$$

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 a network with nodes $\{1,2,3\}$ and a single flow, going from 1 to 3 (directly or via node 2). Furthermore, suppose $p_{12} = p_{23} = 1$ and p_{13} is close to 0. Due to the nature of the backpressure algorithm, queue w_{13} will be filled with credits, to prevent 1 assigning more credits to y_{13}. Suppose that the oldest generation that has a credit in w_{13} is generation $g_{13} = \min\{g \,|\, w_{13}^1(t,g) > 0\}$. Since p_{13} is close to 0, queue w_{13} will take long time to get rid of credits from generation g_{13}. At the same time, since $p_{12} = 1$ queues w_{12} and $w_{1\{2,3\}}$ will quickly get rid of all generations older or equal to g_{13}. Thus, whenever node 1 is selected to transmit, it will transmit a packet from generation $g_1^*(t) = g_{13}$, according to (26). Since $w_{12}(t,g_{13}) = w_{1\{2,3\}}(t,g_{13}) = 0$, node 2 doesn't need more packets from generation g_{13} and the packet it receives will be useless. Indeed, it will almost never be received by node 3, yielding end-to-end rate $f \approx 0$. Nevertheless, we still want to keep link 1-3 active because a few packets transmitted over that link still improve overall performance.

Finding a jointly optimal coding and scheduling strategy that maximizes system utility for finite generation sizes is a difficult problem. In addition to the previously mentioned scheduling issue, when the generation size becomes finite, the results from [14] do not hold either. This implies that packets received at the destination will not necessarily 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 (11) and (17) to

$$y_{ij}^c(t,g) = \begin{cases} M & ,q_i^c(t,g) \le q_j^c(t,g) + w_{ij}^c(t,g) \\ 0 & ,q_i^c(t,g) > q_j^c(t,g) + w_{ij}^c(t,g) \end{cases} \tag{27}$$

$$g_i^*(c,t) = \operatorname*{argmin}_g \sum_J w_{iJ}^c(t) x_{iJ}^c(t)\, 1_{\{w_{iJ}^c(t,g)<0\}} \tag{28}$$

$$x_{iJ}^c(t,g) = \begin{cases} C_{iJ}(t), & c = c_i^*(t), g = g_i^*(c,t), \\ 0, & \text{otherwise} \end{cases} \tag{29}$$

An explanation how this policy is derived is given in the

Appendix. We demonstrate its performance, for finite generation sizes, by simulations in Section 5.

4.2 Rate Adaptation For 802.11-compatible Scheduling

As shown in [13], finding the optimal scheduling rule (13) is an NP-hard centralized optimization problem. Some recent works [21, 13, 22] explore decentralized implementations of similar problems. Applying these ideas to our setting is out of scope of this paper, and left for future work. Instead, we will consider a more realistic, suboptimal scheduling process and we will show how our algorithm can be applied as a distributed heuristic.

Like in most existing wireless mesh networks deployment, we will assume that nodes always transmit packet at the full power $P_i(t) = \{0, P^{MAX}\}$. 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 $p_{i_1,i_2} > 0, (\exists j \in J_2) p_{i_1,j} > 0$ 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 is not under our control. At every time t, 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.

Assuming previously defined schedule $S(t)$, the (12) - (17) simplifies to

$$(c_i^*(t), R_i^*(t)) = \underset{c, R_i}{\operatorname{argmax}} \sum_{K \subseteq J_i} w_{iK}^c(t) C_{iK}(R_i), \quad (30)$$

which can be easily solved in a distributed manner, separately at each node.

We note that, for an arbitrary scheduling process $\{S(t)\}_t$, the distributed routing (11) and rate adaptation (30) algorithm does not necessarily minimize the optimization problem (5). The optimal algorithm will depend on the characteristics of $\{S(t)\}_t$ and it is difficult to characterize. We present (11) and (30) 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 5 that in the case of random, 802.11-compatible scheduling, the heuristic (30) outperforms a conventional, single-path routing approach.

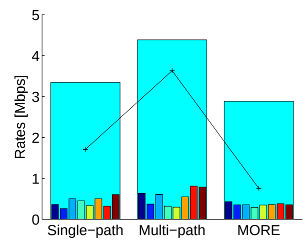

Figure 3: End-to-end rate allocation example: eight flows among randomly selected source-destination pairs. On the y axis we give rates. Small bars denote rates per flow. Large bars denote total rate. Black line gives utilities with an arbitrary scaling (to fit the figure).

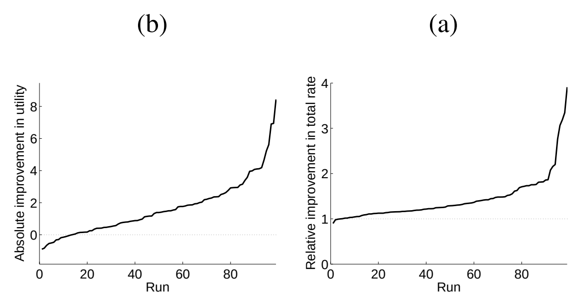

Figure 5: Cumulative performance improvement of our algorithm over MORE: (a) Improvement in utility ($\sum_c \log(f_c^m) - \sum_c \log(f_c^s)$); (b) Improvement in total rate ($\sum_c f_c^m / \sum_c f_c^s$). In all cases we run 100 experiments and we sort them by performance improvement.

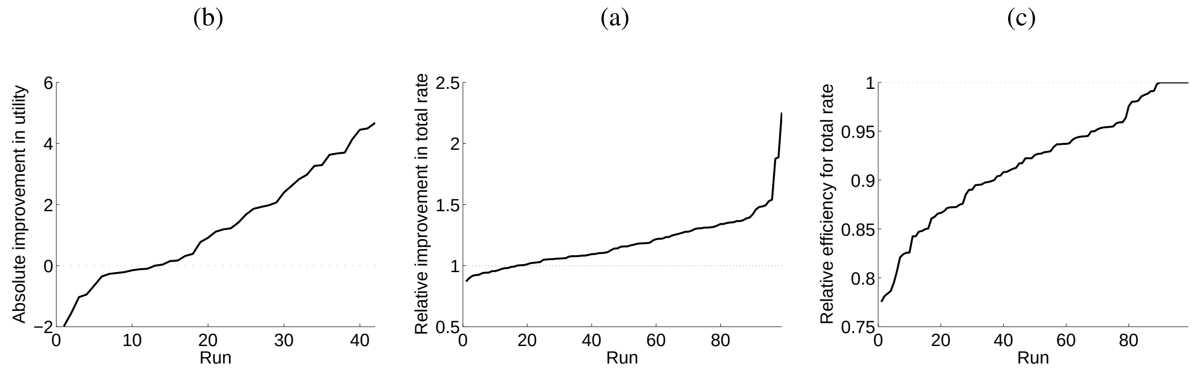

Figure 4: Cumulative performance improvement of multi-path over single-path: (a) Absolute improvement in utility ($\sum_c \log(f_c^m) - \sum_c \log(f_c^s)$); (b) Relative improvement in total rate ($\sum_c f_c^m / \sum_c f_c^s$); (c) Relative efficiency for total rate ($\sum_c f_c^{finite} / \sum_c f_c^{infinite}$) for multi-path routing due to finite generation size of $G = 32$. In all cases we run 100 experiments and we sort them by performance improvement. In many cases, for single-path routing, some flow had zero rates for the duration of the simulation, due to slow convergence. Since this would yield to inifinite utility difference, we don't plot these runs.

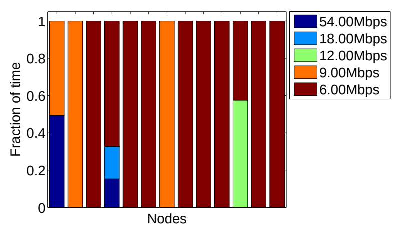

Figure 6: Optimal distribution of PHY rates for roofnet network with 802.11a cards, for one random selection of 8 source-destination pairs. In case of 802.11b cards, almost all nodes are assigned the highest, 11 Mbps rate.

5 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 practical algorithms that can be applied in 802.11-like mesh networks, where 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 4.2, assuming $I(t)$ is randomly selected among backlogged nodes.

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 [11].

In contrast, MORE does not integrate flow control or flow scheduling with the routing algorithm. When simulating the MORE algorithm, defined in [5, 20], we assumed that each source had a large backlog of packets to transmit, and that each relay performed FIFO scheduling among packets from different flows.

We used the roofnet network topology based on 802.11b cards, given in [4], for our simulations. Transmission probabilities between each pair of nodes for different transmission rates are given in [1]. We used $U_c(\cdot) = \log(\cdot)$, hence the rate allocation that maximizes (5) is the proportionally fair rate allocation [19].

We looked at two 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

[2]Since in the simulations we use random and not the optimal schedul-

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.

We developed a discrete-event simulator that implements the three routing, flow and rate control algorithms. We ran simulations to obtain end-to-end rate allocations. Figure 3 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 multiple paths instead of a single one, where we benefit from broadcast and multi-path diversity.

We then ran the previous experiment with 100 random traffic matrices and compared the performances of the different algorithms with respect to the two performance metrics.

Single vs. Multiple Paths We start by illustrating the benefits of the multi-path routing over the single-path routing in Figure 4. We first look at the network utility. The rate allocation obtained by the optimal algorithm (Section 3.3) always maximizes the utility. However, this is not the case for the distributed heuristic (Section 4.2). In our simulations we saw that in about 90% of the runs, the distributed heuristic for multi-path routing achieves higher utility than does single-path routing. In only about 10% 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 multi-path routing algorithm over the single-path one. We see that the advantage is significant even in number of flows is large (see next paragraph for explanation).

Decentralized Heuristic vs. MORE We next compare our decentralized heuristics with MORE. The results are depicted in Figure 5. 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.

ing, the resulting rate allocation does not necessarily have the highest utility.

From these results we can see that in many cases MORE behaves worse then the single-path routing. This resonates with the findings of [5] where the benefits of multi-path decrease as the number of flows increase (for an explanation, see Section 3.5). In addition, MORE is only a routing protocol whereas our single-path routing algorithm also includes more intelligent flow control and flow scheduling.

Effects of Finite Generation Size Figure 4, (c) illustrates the impact of a finite generation size. The performance drop is due to imperfect scheduling (Section 4.1) and occasional linear dependency of received packets. We can see that in most of the cases the efficiency loss is less than 5%. [3]

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]. Furthermore, we use the SNR data from roofnet topology and measurements from [23] to analyze the approximatively optimal performance in the same network topology with 802.11a cards. The results are depicted in Figure 6. As expected, the optimal PHY rate selection is no longer uniform, due to the large number of available rates. This demonstrates a need for an intelligent PHY rate selection algorithm in this framework.

6 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, nor scheduling, and in fact show that the performance benefit drops as number of flows increases. An optimization framework for opportunistic routing that minimizes power consumption is presented in [18], and shows significant benefits

[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.

over [4]. Nevertheless, [18] does not consider network coding, rate maximization, nor the TCP-like primal-dual rate adaptation. Cross-layer design for network coding with unicast or multi-cast sessions is considered in [8]; however, [8] considers only stability and not any form of rate maximization.

Several theoretical analysis of linear network coding algorithms for unicast sessions have been performed [14, 9]. Network coding for unicast sessions is used also in COPE [24]. Compared to COPE, we perform encoding operations only between packets of the same flow; in that respect our approach is orthogonal to COPE.

Our work is an example of cross-layer optimization. Cross-layer design in wireless is a widely research topics (see [12, 25] and references therein). Optimizing network performance in terms of network utility is originally proposed in [19]; see [10] for an overview. Our primal-dual approach is similar to [10], where it is shown that it can capture different versions of TCP.

7 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].

In the case when scheduling is determined by a MAC, 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 for our examples that for 802.11b, using 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] R. Ahlswede, N. Cai, S. R. Li, and R. W. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, 2000.

[8] 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.

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

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

[11] 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.

[12] 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.

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

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

[15] 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.

[16] 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.

[17] 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.

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

[19] 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.

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

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

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

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

[24] 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.

[25] 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.

Appendix

Before proving Theorem 2 we first introduce the following lemma

Lemma 1. *The following equalities and inequalities hold*

$$\sum_{i,j\in\mathcal{N}}\sum_{c\in\mathcal{C}} y_{ij}^{c*}(\mu_i^{c*}-\mu_j^{c*}) = \sum_{c\in\mathcal{C}} f_c^*\mu_i^{c*}, \quad (31)$$

$$\xi_{iJ}^{c*}x_{iJ}^{c*} = \xi_{iJ}^{c*}\sum_{j\in J} y_{ij}^{c*}, \quad (32)$$

$$\sum_{i\in\mathcal{N},J\in\mathcal{P}(\mathcal{N})}\sum_{c\in\mathcal{C}} \xi_{iJ}^{c*}x_{iJ}^{c} \le \sum_{i\in\mathcal{N},J\in\mathcal{P}(\mathcal{N})}\sum_{c\in\mathcal{C}} \xi_{iJ}^{c*}x_{iJ}^{c*}, \quad (33)$$

$$\sum_{i\in\mathcal{N},J\in\mathcal{P}(\mathcal{N})}\sum_{c\in\mathcal{C}} \xi_{iJ}^{c*}x_{iJ}^{c} \le \sum_{i,j\in\mathcal{N},c\in\mathcal{C}} \xi_{ij}^{c*}y_{ij}^{c*}, \quad (34)$$

$$\sum_{i,j\in\mathcal{N},c\in\mathcal{C}} y_{ij}^{c*}(q_i^c(t)-q_j^c(t)) = \sum_{c\in\mathcal{C}} f_c^* q_{\mathrm{Src}(c)}^c, \quad (35)$$

$$\sum_{i\in\mathcal{N},J\in\mathcal{P}(\mathcal{N})} w_{iJ}^c\sum_{j\in J} y_{ij}^c = \sum_{i,j\in\mathcal{N}} w_{ij}^c y_{ij}^c, \quad (36)$$

$$\sum_{i\in\mathcal{N},J\in\mathcal{P}(\mathcal{N})}\sum_{c\in\mathcal{C}} w_{iJ}^c(t)x_{iJ}^c(t) \ge \sum_{i,j\in\mathcal{N},c\in\mathcal{C}} w_{ij}^c(t)y_{ij}^{c*} \quad (37)$$

Proof. Equalities (31) and (32) follow directly from (6) and (7). From (4) we further have $\sum_{c\in\mathcal{C}} x^c_{iJ} = C_{iJ}$. Together with (10) we can derive

$$\begin{aligned}\sum_{i,J,c} \xi^{c*}_{iJ} x^c_{iJ} &\le \sum_{i,J,c}\left(\sum_J \xi^{c*}_{iJ} C_{iJ}\right)\\ &\le \sum_{i,J,c}\left(\sum_J \xi^{c*}_{iJ} C^*_{iJ}\right)\\ &= \sum_{i,J,c} \xi^{c*}_{iJ} x^{c*}_{iJ}\end{aligned}$$

which proves (33). From (32) and (33) we derive (34). Since by definition $q^c_{\mathrm{Dst}(c)}(t) = 0$ we have

$$\sum_{i,j\in\mathcal{N},c\in\mathcal{C}} y^{c*}_{ij}(t)(q^c_i(t) - q^c_j(t)) = \sum_{c\in\mathcal{C},j\in\mathcal{N}} y^{c*}_{\mathrm{Src}(c)j} q^c_{\mathrm{Src}(c)}$$

from which we derive (35). Equality (36) follows from the definition of w^c_{ij}. Also, by definition of scheduling (12)-(17), we have

$$\begin{aligned}\sum_{i,J,c} w^c_{iJ}(t) x^c_{iJ}(t) &\ge \sum_i \max_c \sum_J w^c_{iJ}(t) C_{iJ} (\forall C_{iJ}) &(38)\\ &\ge \sum_{i,J,c} w^c_{iJ}(t) x^{c*}_{iJ} &(39)\end{aligned}$$

which yields (37). □

Proof of Theorem 2: We will follow the idea of the proof of Theorem 2 from [11]. First, let us define Lyapunov function

$$\begin{aligned}W(\mathbf{f},\mathbf{q},\mathbf{w}) &= \frac{1}{2\gamma}\sum_{c\in\mathcal{C}}(f_c - f^*_c)^2 + \frac{1}{2}\sum_{i\in\mathcal{N},c\in\mathcal{C}}(q^c_i - \mu^{c*}_i)^2\\ &+ \frac{1}{2}\sum_{i\in\mathcal{N},J\in\mathcal{P}(\mathcal{N})}\sum_{c\in\mathcal{C}}(w^c_{iJ} - \xi^{c*}_{iJ})^2. &(40)\end{aligned}$$

We want to show that the derivative $\dot W \le 0$. For brevity, we define $f^c_i(t) = f_c(t) 1_{\{i=\mathrm{Src}(c)\}}$. The derivative of W is

$$\begin{aligned}&\dot W(\mathbf{f}(t),\mathbf{q}(t),\mathbf{w}(t)) &(41)\\ &= \sum_c (f_c(t) - f^*_c)(U'_c(f_c(t)) - q^c_{\mathrm{Src}(c)}(t))_{f_c(t)\ge 0}\\ &+ \sum_{i,c}(q^c_i(t) - \mu^{c*}_i)\left(f^c_i(t) + \sum_j y^c_{ji}(t) - \sum_k y^c_{ik}(t)\right)_{q^c_i(t)\ge 0}\\ &+ \sum_{i,J,c}(w^c_{iJ}(t) - \xi^{c*}_{iJ})\left(\sum_{j\in J} y^c_{ij}(t) - x^c_{iJ}(t)\right)_{w^c_{iJ}(t)\ge 0}.\end{aligned}$$

As in (10)-(13) from [11] we have that, when $q^c_i(t) < 0$ the derivative $\dot q^c_i(t)$ is by definition positive; also $\mu^{c*}_i \ge 0$. The same holds for $w^c_{iJ}(t)$ and $f_c(t)$ and we can upper-bound

$$\begin{aligned}\dot W(\mathbf{f}(t),\mathbf{q}(t),\mathbf{w}(t)) &\le \sum_c (f_c(t) - f^*_c)(U'_c(f_c(t)) - q^c_{\mathrm{Src}(c)}(t))\\ &+ \sum_{i,c}(q^c_i(t) - \mu^{c*}_i)\left(f^c_i(t) + \sum_j y^c_{ji}(t) - \sum_k y^c_{ik}(t)\right)\\ &+ \sum_{i,J,c}(w^c_{iJ}(t) - \xi^{c*}_{iJ})\left(\sum_{j\in J} y^c_{ij}(t) - x^c_{iJ}(t)\right) &(42)\end{aligned}$$

Let us further add and substract $U'_c(f^*_c) = \mu^{c*}_{\mathrm{Src}(c)}$, as in [11], to obtain

$$\begin{aligned}\dot W(\mathbf{f}(t),\mathbf{q}(t),\mathbf{w}(t)) &\le \sum_c (f_c(t) - f^*_c)(U'_c(f_c(t)) - U'_c(f^*_c))\\ &+ \sum_{i,c}\mu^{c*}_i\left(\sum_k y^c_{ik}(t) - \sum_j y^c_{ji}(t) - f^{c*}_i\right)\\ &+ \sum_{i,c} q^c_i(t)\left(\sum_j y^c_{ji}(t) - \sum_k y^c_{ik}(t) + f^{c*}_i\right)\\ &+ \sum_{i,J,c}(w^c_{iJ}(t) - \xi^{c*}_{iJ})\left(\sum_{j\in J} y^c_{ij}(t) - x^c_{iJ}(t)\right).\end{aligned}$$

Due to concavity of U_c we have $(f_c(t) - f^*_c)(U'_c(f_c(t)) - U'_c(f^*_c)) \le 0$. Next, let us pick any set of link rates $\{y^{c*}_{ij}\}_{i,j,c}$ that correspond to the optimal flow allocation

$\{f_c^*\}_c$. We expand f_c^* using (31) and (35) to obtain

$$\dot{W}(\mathbf{f}(t),\mathbf{q}(t),\mathbf{w}(t)) \le \sum_{i,j,c}(\mu_i^{c*}-\mu_j^{c*})(y_{ij}^c(t)-y_{ij}^{c*})$$
$$+ \sum_{i,j,c}(q_i^c-q_j^{c*})(y_{ij}^{c*}-y_{ij}^c(t))$$
$$+ \sum_{i,J,c}(w_{iJ}^c(t)-\xi_{iJ}^{c*})\left(\sum_{j\in J} y_{ij}^c(t)-x_{iJ}^c(t)\right)$$

Let us denote with $z_{ij}^{c*}=\mu_i^{c*}-\mu_j^{c*}-\xi_{ij}^{c*}$. Then, from (34), (36) and (37) we have

$$\dot{W}(\mathbf{f}(t),\mathbf{q}(t),\mathbf{w}(t)) \le \sum_{i,j,c}(y_{ij}^c(t)-y_{ij}^{c*})\times \quad (43)$$
$$\left[(\mu_i^{c*}-\mu_j^{c*}-\xi_{ij}^{c*})-(q_i^c(t)-q_j^c(t)-w_{ij}^c(t))\right] \quad (44)$$
$$= \sum_{i,j,c}(y_{ij}^c(t)-y_{ij}^{c*})(z_{ij}^{c*}-z_{ij}^c(t)) \quad (45)$$
$$\overset{(a)}{=} \sum_{i,j,c} y_{ij}^c(t)z_{ij}^{c*}-(y_{ij}^c(t)-y_{ij}^{c*})z_{ij}^c(t) \quad (46)$$
$$\overset{(b)}{\le} -\sum_{i,j,c}(y_{ij}^c(t)-y_{ij}^{c*})z_{ij}^c(t) \overset{(c)}{\le} 0. \quad (47)$$

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 when $z_{ij}^c(t)>0$ then $y_{ij}^c(t)=M>\max_c f_c^*\ge \max_{i,j,c} y_{ij}^{c*}$.

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

$$\mathcal{Q}(t)=\left\{(\mathbf{q},\mathbf{w})\,|\,\sum_{i,j,c}(y_{ij}^c(t)-y_{ij}^{c*})(z_{ij}^{c*}-z_{ij}^c(t))\right\} \quad (48)$$

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 (45) that $\mathcal{E}\subseteq\mathcal{Q}$. We can further apply LaSalle's invariance principle as in [11] 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. □

Derivation of (27)-(29): Let us write modified queue evolution equations $\dot{q}_i^c(t,g)=\left(f_i^c(t,g)+\sum_j y_{ji}^c(t,g)-\sum_j y_{ij}^c(t,g)\right)_{q_i^c(t,g)\ge 0}$ and $\dot{w}_{iJ}^c(t,g)=\left(\sum_{j\in J} y_{ij}^c(t,g)-x_{iJ}^c(t,g)\right)_{w_{iJ}^c(t,g)\ge 0}$. Note that (19) and (20) do not hold anymore. Consequently, we cannot claim that (42) follows from (41) and the proof of Theorem 2 cannot be applied.

We first consider $\dot{q}_i^c(t,g)$. We see from (27) that $y_{ik}^c(t,g)>0$ only if $q_i^c(t,g)>0$. Thus, the exact queue evolution described with $\dot{q}_i^c(t,g)=f_i^c(t,g)+\sum_j y_{ji}^c(t)-\sum_k y_{ik}^c(t)$. 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 (41) we can write

$$\dot{W}(\mathbf{f}(t),\mathbf{q}(t),\mathbf{w}(t)) \le \quad (49)$$
$$\le \sum_c (f_c(t)-f_c^*)(U_c'(f_c(t))-q_{\mathrm{Src}(c)}^c(t)) \quad (50)$$
$$+ \sum_{i,c}(q_i^c(t)-\mu_i^{c*})\left(f_i^c(t)+\sum_j y_{ji}^c(t)-\sum_k y_{ik}^c(t)\right) \quad (51)$$
$$+ \sum_{i,J,c}(w_{iJ}^c(t)-\xi_{iJ}^{c*})\left(\sum_{j\in J} y_{ij}^c(t)-x_{iJ}^c(t)\right) \quad (52)$$
$$+ \sum_{i,J,c} w_{iJ}^c(t)x_{iJ}^c(t)\,1_{\{w_{iJ}^c(t,g^*(c,t))<0\}}. \quad (53)$$

Intuitively (53) 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 (50) + (51) + (52) ≤ 0, we want to minimize (53), which is indeed done in (28) and (29). It is also easy to see from (53) that the naive policy (26) may yield an unbounded drift.