EFFICIENT AND FAIR ROUTING FOR MESH NETWORKS

ANDREA LODI*, ENRICO MALAGUTI*, AND NICOLÁS E. STIER-MOSES**

* *DEIS, University of Bologna,*
Viale Risorgimento 2, 40136 Bologna, Italy; {andrea.lodi,enrico.malaguti}@unibo.it

** *Graduate School of Business, Columbia University*
Uris Hall, Room 418, 3022 Broadway Ave., New York, NY 10027, USA; stier@gsb.columbia.edu

Abstract. Inspired by the One Laptop Per Child project, we consider mesh networks that connect devices that cannot recharge their batteries easily. We study how the mesh should retransmit information to make use of the energy stored in each of the nodes effectively. The solution that minimizes the total energy spent by the whole network may be very unfair to some nodes because they bear a disproportionate burden of the traffic. A Nash equilibrium—achieved when nodes minimize the energy they spend—does not model the situation well because, as retransmissions consume battery without increasing the node's utility, it predicts that nodes refuse to participate. Actually, there are wireless communication protocols, peer-to-peer networks and other systems that provide incentives or impose penalties to encourage nodes to be active and to participate.

We explicitly aim at the solution that minimizes the total energy spent by nodes among those that satisfy a fairness constraint. Although this does not guarantee that the solution is at equilibrium, nodes do not have a big incentive to deviate from the proposed solution since they do not view the situation as extremely unfair to them. This is consistent with the recommendation of Beccaria and Bolelli (1992) who proposed to optimize social welfare keeping user needs as constraints. We propose a distributed and online routing algorithm and compare it to an offline, centralized approach. The centralized approach, besides being unrealistic in terms of information requirements, is also NP-hard to solve. For both reasons, we focus on the former and evaluate it by conducting an extensive set of computational experiments that evaluate the efficiency and fairness achieved by our algorithm.

Keywords. Wireless network, least-energy routing protocol, fairness.

1. Introduction

The *One Laptop per Child* (OLPC) project strives to "create educational opportunities for the world's poorest children by providing each child with a rugged, low-cost, low-power, connected laptop with content and software designed for collaborative, joyful, self-empowered learning." The focus of the project is on education and collaboration. To implement these goals, laptops need to be connected to each other and to the Internet; hence, laptops are equipped with a wireless network interface. This interface allows laptops to connect to a base station or to connect with each other to form a *mesh network*. (The reader is referred to the url laptop.org for extensive information on the OLPC project.)

As described in the OLPC's Wiki (OLPC 2008b), one of the deployment scenarios of the laptops is "a group of kids sitting under a tree." In this configuration, many children that are close-by, all with their own laptop, can collaborate by exchanging information using the wireless capabilities. Under this scenario, there is no outside connectivity. Another possibility that is described in the

Date: Received: Aug 2008; Revised: Apr 16, 2009, Jun 22, 2009 and Jul 13, 2009
Forthcoming on the special issue of *Mathematical Programming*, *Series B*, on Combinatorial Optimization and Integer Programming (2010).

document is to add a *school server* to the simple mesh to form a *school mesh*. The school server takes the role of an access point and provides Internet connectivity to the whole mesh. Laptops that want to communicate with the outside world relay packets to the school server.

Many of the intended users do not have access to a power source at home or in the locations where laptops are to be used. Therefore, to recharge their batteries users rely on mechanical devices that generate electricity using human or animal activity, solar cells, wind towers, and energy derived from other natural resources (OLPC 2008a). For example, users may need to go to specific locations to recharge or they may need to crank a charging device to generate power. Either way, they are likely to be conscientious in how they use their limited battery power.

Laptops participate in the mesh primarily because they want to collaborate and because they need to send or receive information. Instead of sending information directly to the recipient, laptops usually communicate with nearby neighbors that retransmit the information until it reaches its destination. The energy that a laptop spends in retransmissions does not benefit it directly. (In contrast, obviously the laptop that originates the transmission benefits from retransmissions because it uses less energy to reach a laptop that is closer to it.) From a system's perspective, retransmissions enable the network to be more robust and more efficient since the connectivity increases and the power needed to transmit grows super-linearly with the distance between the parties. A large body of research has been devoted to the design of distributed algorithms that implement routing schemes that achieve high throughput and efficiency.

From the point of view of the laptop users, the retransmission of traffic generated by others creates a tension since users have to decide whether to use energy to help the network or to keep the energy for themselves. The current implementation of OLPC deals with this problem in two ways. First, the wireless cards do not consume much energy (current power consumption in mesh-only mode is around 800mW and it is going to be reduced to a half in the next release of the hardware), minimizing the amount of energy that laptops contribute to the system. Second, the wireless radio works while the laptop is in a suspended state. In this way, users are not required to keep all the components of the laptop powered-on to retransmit others' traffic. Both of these features encourage users to contribute power to the network in the form of retransmissions because doing so is not too expensive. Nevertheless, as the example of Section 1.2 shows, depending on the physical location of the laptops, some will end up contributing more than others, which leads to an unfair (or non-equitable) utilization of the scarce resource. Indeed, laptops that are located near the center of the network are more likely to retransmit while nodes close to the periphery will most likely benefit from retransmissions without having to invest their own resources.

The main purpose of this article is to balance the system and the users viewpoints in settings such as those discussed above, where users do not want their batteries depleted by retransmissions but have an incentive to cooperate because they need the network. The network needs to route packets efficiently but does not want to hurt any user unfairly so users continue to cooperate because otherwise not only transmissions will be more expensive but also there will be less users available. We propose an efficient algorithm that chooses a routing scheme that minimizes the total energy spent among the routing schemes that are fair. By designing a protocol that is more fair to users, we give them more incentive to participate in the mesh which is a win-win situation to users and the network simultaneously. The analysis that we propose applies to the OLPC project and to other real world settings as well. Consider as an example the pervasive diffusion of smart phones (mobile phones with advanced capabilities, often including a wireless connection). The market of smart phones grew continuously in the last years, and according to the technology analyst house Canalys, smart phones represented around 13% of the total mobile phone market in the third quarter of 2008 (Canalys 2008). It is not hard to imagine a near future where mesh wireless networks will connect personal devices, such as smart phones, by relaying traffic via multi-hop paths. Other examples, which are already implemented, include temporary networks created during meetings or gatherings, and sensor networks installed for disaster recovery or for geographic surveys.

To embark in this study, we consider measures of efficiency and fairness that allow us to determine, for a given routing procedure, the corresponding benefits for the whole network and for the individual users. Equipped with these definitions, we design an algorithm for routing traffic between laptops in a mesh, which mediates between the two conflicting goals of efficiency and fairness. This algorithm is online and distributed, meaning that it does not require prior information about the demand and it does not require any central computation, respectively.

Most of the literature on wireless networks that focuses on energy consumption has been devoted either to minimizing the total energy spent by the network or to studying the games that arise when users behave in a certain manner. The second stream is related to the study of the efficiency of solutions at equilibrium, which can be quantified by the *price of anarchy*—the worst-case ratio between the cost of an equilibrium and that of the best solution that can be implemented through a central control. An unacceptably high price of anarchy hints that mechanisms with incentives or penalties must be used to induce efficient outcomes. The main contribution of our study is to explicitly consider fairness issues related to participation decisions: a user may leave if it feels that its energy is being used unfairly by others. Moreover, although most articles that deal with competition use the solution concept of Nash equilibrium, it is not clear that users have enough time and wisdom to learn the appropriate behavior. In practice, protocols need not work at equilibrium. For example, both Ethernet and TCP can be exploited by selfish users by not backing-off exponentially or by increasing window-sizes more aggressively than required. But most users do not tweak their communication protocols to squeeze more bandwidth out of them. Because of this, we consider solutions that are fair but not necessarily at equilibrium; the motivation being that when networks are regarded as fair, users have less incentives to act selfishly, henceforth encouraging cooperation.

Unfortunately, theoretical bounds on the relation between fair and efficient solutions are not possible since very simple examples show that fair solutions can be arbitrarily inefficient, independently of the definition of fairness. For instance, consider an example in which a node s wants to transmit to a node t. There is also a third node w located between the first two. Since energy grows super-linearly with distance, the energy needed to transmit from s to w is much lower than the energy needed to transmit directly to t. Although it is clearly more efficient to use w as an intermediate hop, since w has nothing to gain, it should not be forced to spend its energy to help s and t communicate. Hence, under an arbitrary definition of fairness, a fair solution must transmit directly even knowing that it causes an increase in energy compared to the previous alternative. Notice that the ratio between the energy spent under each routing scheme (using w or not) is as large as the ratio between the power needed to transmit directly and the power needed to transmit via w. In summary, the price the system pays for being fair—the worst-case increase in energy needed to get a fair routing—is unbounded.

Obviously, the previous instance is not realistic so the price quoted above is a pessimist estimate that does not necessarily reflect reality. One option to get a more realistic estimate would be to restrict mesh networks to have certain topologies or to make other modeling assumptions. Instead, this paper adopts a computational perspective to estimate how much extra energy is needed to route packets fairly. Since the online algorithm that we propose strives to find a fair solution, the previous example implies that its competitive ratio has to be unbounded too. For the set of random instances that we consider, the empirical competitive ratio—defined as the ratio between the energy consumption in the solution returned by the algorithm and that of an optimal, offline solution—is of the order of 1.1 (see Section 6.1.1). Therefore, the price for achieving a fair routing is approximately 10% more than optimal, in the worst-case.

1.1. **Related Literature.** There is a large number of articles that study routing procedures for wireless networks without centralized control. Because energy is a scarce resource, it is frequently assumed that nodes are selfish and strategize their decisions about how they will spend energy. This led many researchers to propose game-theoretical models to represent the decision-making process faced by nodes, and to use Nash equilibrium as a solution concept. Usually, routing schemes are classified as efficient or not depending on the performance of the resulting equilibria. If central control were available the choice would be to implement a routing scheme to achieve the most efficient solution, typically referred to as a system optimum. The lack of central control prevents this solution from being implemented. In some systems, equilibria are naturally close to system optima and then it is not necessary to elicit a particular behavior from the participants. However, equilibria frequently lead to inefficient solutions. The role of the system designer is to engineer mechanisms that induce cooperation by providing incentives to nodes.

A first group of papers is aimed at studying the efficiency of equilibria when nodes are selfish and there is no possibility of centralized control. In some cases nodes can adjust the transmitting power depending on the node they want to reach in their neighborhood. Ji and Huang (1998) propose a non-cooperative game that models nodes that wish to communicate with a single destination. They propose an algorithm that achieves a Nash equilibrium. Lee, Mazumdar, and Shroff (2005) consider a similar framework where each node has a utility function that quantifies the communication quality experienced by it. The article develops a distributed algorithm that converges asymptotically to the system optimum. Kesselman, Kowalski, and Segal (2005) compute the price of anarchy in the broadcast and convergecast games (all to one communications). Some articles study the properties of the resulting graphs. Chun et al. (2004) consider a network formation game—a game where selfish nodes connect to form routing networks—and analyze the topology, performance, and resilience of the graph resulting from a Nash equilibrium. Eidenbenz, Anil Kumar, and Zust (2006) analyze a game in which nodes adjust the transmitting power to get a desired connectivity to the rest of the network.

When the resulting equilibrium is inefficient, the system designer may consider introducing a mechanism to elicit coordination so nodes behave in a prescribed manner, or to detect and isolate misbehaving nodes. One possibility is to create a market of micro-payments. Nodes are compensated when they cooperate and forward others' traffic, but have to pay when they receive help from others. Buttyán and Hubaux (2003) associate a counter to each node that is decreased when the node originates a packet and increased when the node forwards a packet. If the node wants to send its own packets, the value of its counter must remain positive. The article implements this protocol in a network simulator to study its properties. Anderegg and Eidenbenz (2003) propose a payment-based protocol designed in such a way that an equilibrium coincides with the system optimum.

Another option is a reputation-based system, where nodes store the reputation of their neighbors and use this information to decide how much traffic to forward from and to them. A misbehaving node (e.g., one that does not forward packets) may be cut out from the network because nodes will not want to communicate with it. Marti et al. (2000) design mechanisms to detect nodes that are overloaded, selfish, malicious, or broken. Routing protocols avoiding such nodes are experimentally able to improve the throughput. Buchegger and Boudec (2002a, 2002b) and Mahajan et al. (2005) present mechanisms that detect and punish nodes when they do not follow the prescribed behavior, and hence achieve efficient outcomes. He, Wu, and Khosla (2004) propose a reputation-based protocol, where the reputation of a node is propagated to the network through a secured protocol. Results on the implementation of the proposed protocol into a network simulator show that it can successfully identify selfish nodes and punish them when they misbehave. Milan, Jaramillo, and Srikant (2006) and Levin (2006) look at models to tell between packet collisions and misbehaving nodes, and punish nodes only when necessary. Karakostas and Markou (2008) consider a protocol

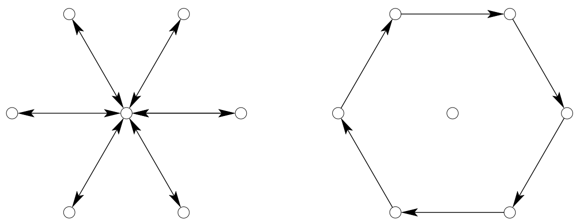

FIGURE 1. A motivating example. *Left*: Most efficient routing. *Right*: Most fair routing.

that prohibits sending information to misbehaving nodes and investigate necessary and sufficient conditions for the existence of an equilibrium. The authors use a utility function that measures the difference between the flow sent by a node and that forwarded by it. This idea is close to the measure of unbalance that we develop in Section 3.

1.2. **A Motivating Example.** The following example illustrates the main problem that arises when one tries to minimize the energy spent collectively by the network.

Example 1. *Consider the network of seven users depicted in Figure 1, where the distance between all neighboring users is equal. Suppose that the central user can reach all other users but the outer users can only reach their three nearest neighbors. The outer users wish to send a unit of traffic to the user furthest away from them, and the central user does not need to communicate with anybody.*

The routing pattern shown on the left of the figure is the most efficient because it transmits all the information along shortest paths. The drawback is that the central user is in a very unfair situation since it forwards all transmissions generated by others, while it is not actively communicating with anybody. In the long term, a user in this situation may decide to leave the network because the routing pattern is unfairly draining its battery. Instead, the routing pattern on the right of the figure is less efficient—because all communications need three hops instead of two—but it is fair. All users contribute the same to the "common good" except the central user, which is not contributing because it is not using the network.

As mentioned before, the goal of this paper is to design routing patterns that are not overly unfair. We follow the recommendation put forward by Beccaria and Bolelli (1992) and find routing patterns that minimize a global criterium (efficiency) with individual needs as constraints (fairness).

1.3. **Structure of the Paper.** The paper is organized as follows. Section 2 introduces some basic definitions, our model and the assumptions we consider throughout the paper. In Section 3, we introduce a specific utility measure to quantify the fairness under a given routing scheme. Section 4 presents our routing algorithm, which is distributed and online. This is compared to an offline, centralized approach that is introduced in Section 5. Offline solutions, besides being unrealistic in terms of information requirements, are shown to be NP-hard to compute, which justifies looking at simpler approaches. Section 6 evaluates the online algorithm proposed in Section 4 by conducting computational experiments. The resulting solutions are compared to the offline problem to evaluate the efficiency and fairness achieved by our algorithm. Finally, Section 7 discusses some implementation issues and Section 8 wraps up the presentation.

2. Description of the Model

We model mesh networks abstractly so the analysis is independent of specific architectures. Because our goal is to derive insights, we do not look at implementation details; instead, we adopt a macroscopic model in which transmitted information is represented as a flow on a network. Flows provide an abstraction that is useful to model the traffic of packets. In particular, we do not consider issues related to interference and conflicts among transmitters, which depend on the specific protocol that is adopted, and thus are out of the scope of this paper.

The main elements of the network are the nodes—representing laptops—which generate information that has to be transmitted to recipients. Nodes retransmit when it is necessary to help others communicate; they do this in exchange for the future help they may get from others for their own communication needs. To communicate with each other, nodes use the radio equipment in their wireless network interfaces to transmit and to receive information, both of which require energy from their reserves. To simplify the presentation, we are going to disregard the energy used when not actively transmitting or receiving because it is many orders of magnitude smaller. For instance, some protocols switch off the radio when possible to save energy.

Network nodes are represented by vertices of a simple graph $G = (V, A)$. The set $A \subset V \times V$ consists of directed arcs (i, j) that indicate that transmissions of node i can reach j directly. Initially, each node has C_i units of energy available representing the initial load of the battery. Every arc $(i, j) \in A$ is associated with a weight $p_{i,j}$ that denotes the energy that is going to be used to send a unit of information (e.g., a bit or a packet) from i to j. This energy equals the transmitting power $\pi_{i,j}$ used on arc $(i, j) \in A$ multiplied by the time needed to transmit one unit of information.

We consider two cases regarding transmit power: it is either fixed (which allows nodes to transmit within a fixed range but always spending the same amount of energy) or it is tunable (which allows nodes to perform power control and use only the amount of energy needed to communicate with a peer). In the case of fixed transmit power, nodes transmit at the same power irrespective of the recipient; hence, $\pi_{i,j}$ is a constant for all $(i, j) \in A$ (we assume that $\pi_{i,j} = 1$ without loss of generality). In this case, nodes can communicate directly with all nodes at distance $d_{i,j} \leq d_{\max}$, where $d_{i,j}$ is the physical distance between the two nodes. In the case of power control, $\pi_{i,j}$ is the power needed to transmit intelligible information from i to j (more power is not necessary and with less power the signal-to-noise ratio is too low). As it is usual in models of wireless networks (e.g., Rappaport 1996), we assume that $\pi_{i,j}$ is proportional to $(d_{i,j})^\alpha$ where $\alpha \in [2, 8]$ is a constant that depends on the environment. We let $\pi_{\max}$ be the maximum transmitting power, meaning that two nodes i and j can communicate directly if $\pi_{i,j} \leq \pi_{\max}$. To simplify the presentation, we assume the time needed to transmit one unit of information is equal to one, thus the numerical values of $p_{i,j}$ and $\pi_{i,j}$ coincide for all $(i, j) \in A$. Similarly, by considering the appropriate units, we have that $\pi_{i,j} = (d_{i,j})^\alpha$. Finally, nodes require a constant energy ρ to receive a unit of information.

We encode the communication needs of nodes in the mesh network as an origin-destination (OD) matrix. An entry $(i, j) \in V \times V$ quantifies the amount of information $r_{i,j}$ to be transmitted between that pair of nodes. Let us define $K := \{(i, j) \in V \times V : r_{i,j} > 0\}$ as the pairs that need to communicate with each other. We assume that all nodes will either send or receive a positive amount of traffic, i.e., $\sum_{j \in V} (r_{i,j} + r_{j,i}) > 0$ for all $i \in V$. This is not a strong requirement because a node that does not need to communicate will probably switch the radio off to save energy. We assume that the wireless communication protocol in use can compute shortest paths between nodes of the network, either exactly or approximately.

Flows denote how information travels across the network. Note that the information transmitted between a pair of nodes can be split into multiple paths if needed. We let $x_{i,j}^{s,t} \geq 0$ be the flow corresponding to OD pair $(s, t) \in K$ that is transmitted along an arc $(i, j) \in A$. A flow is feasible when the following constraints are satisfied. First, flow-conservation constraints must be satisfied at all nodes $i \in V$, for all OD pairs $(s, t) \in K$:

$$\sum_{(i,j) \in A} x_{i,j}^{s,t} - \sum_{(j,i) \in A} x_{j,i}^{s,t} = \begin{cases} r_{s,t} & \text{when } i = s \\ -r_{s,t} & \text{when } i = t \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

Second, nodes cannot use more energy than what is available in each node $i \in V$. Hence, we require that $E_i \leq C_i$, where the energy spent by node i when routing according to flow x equals

$$E_i := \sum_{\substack{(j,i)\in A \\ (s,t)\in K}} \rho\, x^{s,t}_{j,i} + \sum_{\substack{(i,j)\in A \\ (s,t)\in K}} p_{i,j}\, x^{s,t}_{i,j}\,. \tag{2}$$

Note that, when it is obvious to what solution x we are referring to, we do not make the dependence between E_i and x explicit for simplicity of notation. We do the same for other parameters we introduce below.

The primary goal of a routing scheme is minimizing the total energy $E := \sum_{i\in V} E_i$ that is spent by the nodes. We refer to an algorithm that achieves this goal as being *efficient*. The secondary goal, discussed in Section 3, is to be fair to participating nodes. Notice that although the algorithm we will present in Section 4 is online and hence considers time, we defer the discussion of the timing issues until then. For the time being, we just remark that we are not concerned by momentary variations of the mentioned objective functions; only values a-posteriori matter.

3. Fairness Measures

In this section, we explore alternative measures of fairness of a routing pattern. Mainly, we want to quantify how likely it is that nodes participate and cooperate with the network, which they will only do if they have a benefit. Nodes extract positive utilities from being connected to the network because they can communicate with other nodes which are too far to reach directly. They face negative utilities when spending energy in retransmissions that do not benefit them. Consequently, we distinguish between the energy a node spends for its own traffic (either sending or receiving information), and the energy it spends to forward information for others.

There are several possible ways of measuring the fairness (or equity) among the participants of a system. In the context of flow control for telecommunication networks, Jaffe (1981) proposed to call a routing scheme 'fair' when the minimum utility extracted by a node is maximum. Subsequently, the so-called *max-min flow control* was largely adopted by the networking literature; see for example the classical book by Bertsekas and Gallager (1992). Bhargava, Goel, and Meyerson (2001) consider measures of fairness in the context of resource allocation, and discuss different approaches that can be unified under a concept called 'majorization.' The basic idea is that the fairest allocation of resources should maximize the utility of the user with lowest utility, then the sum of the utilities of the two users with lowest utility, and so on. Jahn et al. (2005) considered solutions where the minimum and maximum utilities are as close as possible, where the maximum and the minimum are taken among users that are competing with each other. This measure captures the 'envy' that one user may feel with respect to another user. Furthermore, Jahn et al. also considered solutions in which the minimum utility is as close as possible to the maximum possible utility that could have been achieved by that user. In the latter case, the measure quantifies the *regret* that users may feel a-posteriori.

Following Jaffe (1981), we will strive to find routings in which the node in the most unfair situation is as good as possible because otherwise this node will have a strong incentive to leave the network. We quantify the utility extracted by each node for a given routing scheme by comparing the energy a node spends for the network with the energy the network spends for the node. The following example motivates this comparison.

Example 2. *Consider a network defined by nodes $i = \{1, \ldots, 5\}$, placed on a line at position i, with demands $r_{3,1} = r_{3,5} = r_{2,4} = r_{4,2} = 1$. Power control is used and the distance between nodes is $d_{i,j} = |i - j|$. The maximum power is such that nodes can transmit to nodes at distance at most 2, and $\rho = 0$ and $\alpha \geq 2$.*

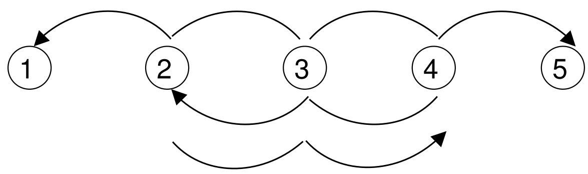

FIGURE 2. Instance used in Example 2.

In essence, this instance has two possible solutions: either all flow is routed directly or all flow is relayed using neighbors, as shown in Figure 2. Because the energy that is needed to transmit to a node at distance two is more than four times the energy used to transmit to an immediate neighbor, the former is at least twice as expensive. The drawback of the latter is that, e.g., node 3 has the burden of retransmitting all traffic between nodes 2 and 4. Although this may be considered as unfair to node 3 at first sight, one should notice that nodes 2 and 4 spend as much energy helping node 3 as node 3 spends for them, alleviating this concern.

The definition of fairness we adopt is based on the two energy consumptions described in the example. Indeed, we define the unbalance $b_i := E_i^{\text{out}} - E_i^{\text{in}}$ of a node $i \in V$ as the difference between the energy E_i^{out} that node i spends for other nodes and the energy E_i^{in} that other nodes spend for node i. Nodes with a high unbalance experience an unfair situation because they are contributing more to the network than the benefits they receive. When a node i retransmits information corresponding to a communication between two other nodes, both the sender and receiver benefit from the retransmission (although not necessarily in an equal manner). To quantify that, we let $\eta^o \in [0,1]$ (resp. $\eta^d = 1-\eta^o$) be the benefit experienced by the origin (resp. destination) per unit of energy spent in a retransmission. Hence, the total benefit arising from retransmissions by others equals the unitary benefit times the total energy spent by others, as can be seen in the following formulas.

$$E_i^{\text{out}} = \sum_{\substack{(s,t)\in K: s\neq i\\(i,j)\in A}} p_{i,j} x_{i,j}^{s,t} + \sum_{\substack{(s,t)\in K: t\neq i\\(j,i)\in A}} \rho x_{j,i}^{s,t} + \sum_{\substack{(i,t)\in K\\(i,j)\in A}} \eta^d p_{i,j} x_{i,j}^{i,t} + \sum_{\substack{(s,i)\in K\\(j,i)\in A}} \eta^o \rho x_{j,i}^{s,i} \tag{3}$$

$$E_i^{\text{in}} = \sum_{\substack{(i,t)\in K\\(\ell,j)\in A: \ell\neq i}} \eta^o p_{\ell,j} x_{\ell,j}^{i,t} + \sum_{\substack{(i,t)\in K\\(\ell,j)\in A}} \eta^o \rho x_{\ell,j}^{i,t} + \sum_{\substack{(s,i)\in K\\(\ell,j)\in A: j\neq i}} \eta^d \rho x_{\ell,j}^{s,i} + \sum_{\substack{(s,i)\in K\\(\ell,j)\in A}} \eta^d p_{\ell,j} x_{\ell,j}^{s,i}. \tag{4}$$

With this interpretation, the flow depicted in Figure 2 is completely fair, on top of spending the least possible amount of energy. Indeed, let us consider that transmissions only benefit the source node (i.e., $\eta^o = 1$ and $\eta^d = 0$). Nodes 1 and 5 have zero unbalance because they do not transmit information. Nodes 2 and 4 retransmit a unit of information for node 3 and vice-versa, so the unbalance of these two nodes is also nil. Finally, node 3 retransmits two units for nodes 2 and 4 but they retransmit two units for node 3, so the unbalance is also nil.

Clearly, a high unbalance represents an undesirable situation for a node that we want to prevent. To facilitate the comparison across nodes with different demand requirements, we normalize the unbalance b_i with its absolute maximum E_i^{out} (which would occur if the node does not transmit any traffic of its own). To transform penalties to utilities, we subtract the relative unbalance from 1. We summarize the calculation of the utility in the following definition. Note that other functional forms would have been possible but we found that ours is simple and works well.

Definition 1. *For a given flow x, the* fairness coefficient *of a node is*

$$\Phi_i := 1 - \frac{b_i}{E_i^{out}} = \frac{E_i^{in}}{E_i^{out}} \geq 0, \tag{5}$$

where we set $\Phi_i = 1$ *by convention when* $E_i^{out} = E_i^{in} = 0$*, and* $\Phi_i = +\infty$ *when* $E_i^{out} = 0$ *and* $E_i^{in} > 0$.

The fairness coefficient we just defined encodes the utility extracted by each node. For instance, a high value indicates a fair situation for the node because the node is receiving more from the network than what it is contributing. This implies a high desire of the node of being in the network. On the contrary, a low fairness coefficient indicates an unfair situation. This means that it is contributing to the network more that it is receiving from the network. In this situation, the node could decide that it has no interest to participate in the network and disconnect. This is highly undesirable, since it can reduce the network efficiency and connectivity. Ideally, all nodes would experience high utilities to reduce the likelihood that they disconnect. This motivates the following definition.

Definition 2. *For a given flow* x*, we refer to the minimum fairness coefficient among all nodes* $\Phi := \min_{i \in V} \Phi_i$ *as the* fairness value *in the network.*

The following property implies that not all nodes will be able to extract an arbitrary high utility; in other words, there is always a node that has to contribute more than others in the system.

Proposition 1. *For any flow,* $\Phi \in [0, 1]$.

Proof. The nonnegativity follows by definition. To prove the upper bound note that $\sum_{i \in V} E_i^{\text{in}} = \sum_{i \in V} E_i^{\text{out}}$ because both express the total work that all nodes do for other nodes. Hence, there must exist some $i \in V$ such that $E_i^{\text{in}} \leq E_i^{\text{out}}$ as required. □

The previous proof also implies that when $\Phi = 1$, every node's contribution and benefit match, as it happened in Example 2. This fact shall provide the intuition needed to prove a complexity result in Section 5.

Corollary 2. *A flow has* $\Phi = 1$ *if and only if* $E_i^{in} = E_i^{out}$ *for all* $i \in V$.

Even though this situation can be considered fair for all nodes, $\Phi = 1$ does not guarantee that the solution is Pareto-efficient, meaning that there may be another solution (possibly with $\Phi < 1$) in which all nodes consume less than or equal energy than in this solution. The following section discusses the concepts of Pareto-efficiency and domination in detail.

3.1. **Meaningful Paths.** Even though there are various ways in which information can be routed, not all of them are equally efficient and fair. If not done carefully, an effort to make a routing more fair can result in solutions that are unnecessarily inefficient. Indeed, to balance the energy spent by nodes, the routing algorithm can make some nodes retransmit information unnecessarily. For this reason, we will restrict our consideration to *Pareto-efficient* paths. The following definition will allow us to formalize this restriction.

Definition 3. *For a fixed routing* x*, consider a path* P *that carries traffic, and another path* P' *with the same origin and destination. We say that* P' dominates P *if* $E_i(P) \geq E_i(P')$ *for all* $i \in V$ *and the inequality is strict for at least one node. Here,* $E_i(\cdot)$*, with a slight abuse of notation with respect to* (2)*, denotes the energy spent by node* i *to route the traffic on the path given as argument. If there exists a path that dominates* P*, we say that* P *is* dominated.

In words, a path dominates another if all nodes spend less energy using the first path than using the second. Note that paths not included in one another cannot dominate or be dominated. Indeed, referring to the set of nodes visited by path P by $V(P)$, consider a pair of paths P and P' such that $V(P) \not\subset V(P')$ and $V(P') \not\subset V(P)$. Nodes in $V(P') \setminus V(P)$ obviously would spend less in P than in P' because P does not go through them, and viceversa. Obviously, if a path dominates another, it makes more sense to use it because all nodes are better off. Hence, to guarantee that the routing is not unnecessarily inefficient, we allow a path to be used only when no other path dominates it. For example, this consideration prevents routing information along cycles because the subpath with the cycles removed dominates it. Using the language of multi-criteria optimization, the paths used by the algorithm will be in the *efficient frontier*. (We refer the reader to, e.g., Fudenberg and Tirole (1991, Chapter 1, Section 2.1.4) for an introduction to Pareto-efficiency and domination.) The following definition is the converse of Definition 3, where we refer to non-dominated paths as being *meaningful*.

Definition 4. *For a fixed routing x, consider a path P that carries traffic. We say that P is* meaningful *if no other path P' from the same origin to the same destination dominates P.*

We now show a condition that can be used to verify if a path is meaningful.

Proposition 3. *A path $P = (i_1, \ldots, i_k)$ is meaningful if and only if $p_{i_j,i_{j+1}} < p_{i_j,i_\ell}$ for all $1 \leq j \leq k-2$ and $j+2 \leq \ell \leq k$.*

Proof. To prove the forward implication, suppose that there exists a path P' that dominates P. We already know that $V(P') \subseteq V(P)$ because otherwise P' cannot dominate P. Note that $E_i(P)$ and $E_i(P')$ are zero for all $i \notin V(P)$. Furthermore, $(p_{i,\mathrm{succ}_P(i)} - p_{i,\mathrm{succ}_{P'}(i)})r = E_i(P) - E_i(P') \geq 0$ for all $i \in V(P)$, where $\mathrm{succ}_Q(i)$ refers to the successor of i along path Q, and r is the flow routed along P. Let i_j be the last node of path P such that $(i_1, \ldots, i_j)$ is also the initial subpath of P'. From the previous expression, $p_{i_j,\mathrm{succ}_{P'}(i_j)} \leq p_{i_j,\mathrm{succ}_P(i)}$, which contradicts the claim. To prove the backward implication, suppose that there exist j and ℓ satisfying that $p_{i_j,i_{j+1}} \geq p_{i_j,i_\ell}$. Defining $P' = (i_1, \ldots, i_j, i_\ell, \ldots, i_k)$, we see that P' dominates P. □

Instead of normalizing the fairness coefficient by dividing b_i with E_i^{out} in (5), we could have used the total energy E_i spent by the node. The fairness coefficient then becomes

$$\Phi_i := 1 - \frac{b_i}{E_i} = \frac{E_i^{\mathrm{net}}}{E_i}, \tag{6}$$

where E_i^{net} denotes the total energy that the network (including node i) spends to transmit i's traffic. Even though this is a legitimate alternative, it has the drawback—for the case of power control—that a node could increase its utility and be in a fairer situation by spending more energy to send its own traffic (because both the numerator and denominator of (6) increase). If we were to adopt this definition, nodes would have an incentive to choose a first hop of higher cost than necessary, which is adverse to both the node and the network. With the adopted measure, this does not happen because the energy spent in a node's own traffic is not part of the fairness coefficient.

As it was mentioned earlier, our algorithm guarantees that meaningful paths are used. Allowing dominated paths would impact the outcomes generated by our methodology in a profound way. Indeed, they would provide the possibility of artificially increasing the fairness level of a routing by making minimal changes to the chosen routes. Making use of dominated paths may create a perception of unfairness that can hinder collaboration; nodes can easily detect that their energy is not used efficiently (e.g., in the case of cycles or using Proposition 3). Instead, if a path is meaningful, at least a node experiences a real benefit from an energy increase of some other nodes.

Unfortunately, even forbidding dominated paths might not fully solve the problem of artificially increasing the fairness value by using unnecessarily long paths. Indeed, let us highlight that the fact that a path is meaningful holds with respect to a fixed routing. These definitions concern unilateral deviations by nodes and not by groups of nodes that may get together to reroute flow in a more efficient manner. Hence, even if a flow routes traffic along meaningful paths only, there may exist another routing under which *all* nodes spend less energy. As an example, consider the instance represented in Figure 3. There are 6 nodes placed on two parallel lines. Let the communication demands be $r_{3,1} = r_{4,6} = 1$, $\alpha \geq 2$, $\eta^o = 1$ and $\eta^d = 0$. The distance between two consecutive

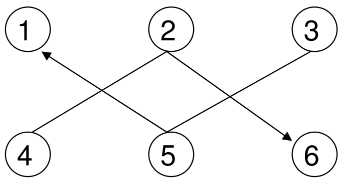

FIGURE 3. Meaningful paths in a dominated routing.

nodes on a line is 1, while the length of any diagonal arc, as shown in the figure, is larger than 1. According to Definition 4, paths $3 \to 5 \to 1$ and $4 \to 2 \to 6$ are meaningful. However, if flow were routed along paths $3 \to 2 \to 1$ and $4 \to 5 \to 6$, which are also meaningful, all nodes would spend less energy. Obviously, the fairness value at each node in the previous two solutions is different. It is possible that the more expensive routing is more fair. In this example, the solution that uses paths $3 \to 5 \to 1$ and $4 \to 2 \to 6$ provides more utility to nodes 3 and 4.

The issue prompted by the instance shown in Figure 3 seems to be, however, unavoidable if the fairness coefficient of a node depends on both the energy it spends for the network and the energy the network spends for it. Actually, it seems that this problem is unavoidable whenever the fairness coefficient of a node is a function of the latter term because creating spurious work for a node increases its fairness coefficient.

A possible solution to the problem presented above is to adopt a fairness coefficient that disregards the contribution of the network to the node. In this spirit, the most natural alternative is to consider the ratio of the energy a node spends for its own traffic to the energy it spends overall. For example, applying this alternative measure to Example 2 (with $\rho = 0$ and $\eta^o = 1$), we would conclude that the solution of Figure 2 has fairness $\Phi = 1/2$, since nodes 2, 3 and 4 spend half of their energy in retransmissions. This measure captures that nodes prefer to spend energy in their own transmissions rather than in retransmitting others' traffic. However, such a definition cannot capture that a mutual exchange benefits both parties, which is the basis of network cooperation. A second drawback—which applies in the case of power control—is that a node would have the incentive to spend more energy in its transmissions as it happened with the alternative normalization discussed after Proposition 3. Running computational experiments with this definition of fairness coefficient, we observed that nodes quickly tend to avoid retransmitting, thus leading to very inefficient solutions.

As we have argued, out of all the different alternatives we considered, we chose the one that consistently produces routings of good quality in terms of energy consumption while choosing paths that are reasonable both from the nodes' perspective as much as from the system's point of view. Nevertheless, our solutions are not robust to coalitions that deviate to improve performance, but none of the other alternative were robust to this either. To tackle this more complicated issue, one needs to change the approach and look at cooperative game theory.

4. A Distributed, Quick and Fair Algorithm

This section proposes a distributed, online algorithm for routing in mesh networks that we call Distr Online Routing. The goal of the algorithm is computing efficient and fair routing tables, for all nodes and all time periods. The algorithm is distributed because each traffic demand is treated independently and no knowledge of the complete traffic matrix is needed. Furthermore, it is online since it only uses the information that is available at a given moment.

We model time as a sequence of k periods, and consider a fixed planning horizon T. The planning horizon represents the time when nodes' batteries will be recharged (e.g., laptops, that are owned by students, are recharged in the morning when they arrive to school and have a charger available).

algorithm DISTR ONLINE ROUTING
input: network, OD matrix over time
output: flow over time
initialization: let $\kappa = p$ and $c = C$, where C = vector of initial battery charges
at each time period:

for all $s \in V$ and $(i, j) \in A$, set $\kappa^s_{i,j} = \begin{cases} \kappa_{i,j} & c_i > \mathcal{B}_i \\ p_{i,j} & c_i \leq \mathcal{B}_i \text{ and } i = s \\ +\infty & c_i \leq \mathcal{B}_i \text{ and } i \neq s \end{cases}$
during this period route traffic originating in s along shortest paths with respect to κ^s
update remaining battery c
remove all nodes such that $c_i \leq 0$ because they ran out of battery
update $\kappa_{i,j}$ for all arcs (i, j) using (7)

end

FIGURE 4. Summary of algorithm DISTR ONLINE ROUTING.

We impose a horizon because it will allow us to evaluate the behavior of DISTR ONLINE ROUTING by comparing the outcome to the offline version of the problem described in Section 6. In practice, though, a horizon is not needed and the algorithm can work without using it. We assume that a dynamic OD matrix encodes the traffic demand during each time period. The values of the matrix for a given period are unknown until that time. Since it is obviously difficult to achieve a high level of fairness in a period without knowing the demand, we only consider the value of fairness after reaching the time horizon. Indeed, it is unlikely that users are concerned about the use of their batteries over short periods of time.

The algorithm routes traffic originating in s along shortest paths with respect to artificial costs κ^s that we assign to arcs. Default artificial costs κ are updated in every period by increasing those of arcs outgoing from nodes with a low fairness coefficient. This has the effect of decreasing the likelihood of having to forward others' transmissions, making the resulting solution more fair while maintaining its efficiency. To prevent a node from running out of battery earlier than necessary we modify κ into κ^s. To that extent, we fix a threshold value for the residual battery of nodes; when a node reaches the threshold, it stops retransmitting information. We assume that the lower-level protocol can compute shortest paths (or approximations thereof) with respect to our metric κ^s. The route selection procedure used by the OLPC project, which is based on a simplified version of the *Hybrid Wireless Mesh Protocol* that was proposed in the 802.11s draft, can do this. Notice that every period is relatively long compared to the time-scale of transmissions, so updating the artificial costs and computing shortest paths constitutes a small overhead in extra battery consumption, effort and time. Furthermore, even if the algorithm is implemented in a distributed manner, it does not impose a big communication overhead. All the information required to run the algorithm locally is how much energy nodes spent for each other. Implementation issues are further discussed in Section 7.

Figure 4 shows algorithm DISTR ONLINE ROUTING. Initially, it sets costs $\kappa_{i,j} := p_{i,j}$ for all arcs. At every time-period $L \in \{1, 2, \ldots, T-1\}$, the algorithm updates κ and, thus, the paths it uses. To make the solution more fair, it increases the cost of arcs that originate in a node i with low fairness coefficient Φ_i. This is achieved by letting the artificial cost at time L be

$$\kappa_{i,j} := f(p_{i,j}, \Phi^L_i) \tag{7}$$

for all arcs $(i, j) \in A$, where Φ^L_i is the fairness coefficient in that time period. Here, $f(p, u) : \mathbb{R}^+ \to \mathbb{R}^+$ is a function such that $\partial f/\partial p > 0$ and $\partial f/\partial u < 0$. When f does not depend on u, the algorithm computes a set of shortest paths on the original graph at every time period. In this case, the algorithm outputs an optimal routing scheme with respect to efficiency although the solution is likely to be very unfair. However, when the dependence of u is strong, the output has a higher level of fairness at the expense of the efficiency of the solution. In our computational study we set $f(p, u) = pu^{-\beta}$, where β is a positive constant that indicates the importance given to the fairness of the final solution.

To help nodes survive as much as possible, the algorithm modifies artificial costs κ when the residual battery level of a node i falls under a threshold value $\mathcal{B}_i$. Using historical energy requirements, nodes could estimate the amount of energy needed until the time horizon. The choice of the threshold value $\mathcal{B}_i$ depends on the predicted total energy needed until the end of the horizon, on risk-tolerances, on the importance attributed to the loss of one node, and on the preference between a node that stops forwarding but generates traffic and a node that forwards until its battery is completely drained.

The algorithm selects the costs κ^s for all traffic originating in node s as follows. If a node i has sufficient energy available, κ is used. If a node i risks not reaching the planning horizon, the algorithm sets the cost to infinity in all outgoing arcs so i is not selected to retransmit traffic. For the traffic originating in i, the algorithm uses the original $p_{i,j}$ to prevent the node from using a longer first hop to balance the fact that it is not retransmitting others' traffic (which would drain its battery faster). Indeed, a node i that has stopped forwarding is likely to have outgoing arcs with small κ since its fairness coefficient is large because the algorithm does not allow it to forward traffic. Because of the distortion of the values of κ from the original costs p, there is a risk of selecting, for traffic originating in i, an inexpensive path with respect to κ that has a very expensive first hop with respect to p. This led us to use p instead of κ for this situation.

The next theorem proves that our algorithm produces meaningful paths. We remind the reader that this does not guarantee that the traffic cannot be transmitted in such a way where, on the whole, all nodes spend less energy. This only indicates that any unilateral deviation is not convenient.

Theorem 4. DISTR ONLINE ROUTING *always selects meaningful paths (with respect to the original costs $p_{i,j}$).*

Proof. Consider a path $P = (i_1, \ldots, i_k)$ that routes flow. Since at the time L when the flow was routed, P was a shortest path, we have that $\kappa_{i_j,i_{j+1}} < \kappa_{i_j,i_\ell}$ for all $1 \leq j \leq k-2$ and $j+2 \leq \ell \leq k$. The update rule in (7) is monotone with respect to p, which means that given two arcs (i, j) and (i, k) from i, we have that $p_{i,j} < p_{i,k}$ if and only if $\kappa_{i,j} < \kappa_{i,k}$ at any period of time L. The result follows from Proposition 3. □

5. MODELING THE CENTRALIZED, OFFLINE PROBLEM

In this section, we consider the centralized, offline version of the routing problem. Although this solution cannot be implemented in practice because it requires information that is not likely to be available to the nodes, this study provides us with a benchmark for the online situation described before. We assume that the full OD matrix is known *a priori* (although in real applications it is revealed over time), and propose a linear programming solution that routes the necessary demand while balancing the objectives of efficiency and fairness. Following the recommendation of Beccaria and Bolelli (1992), we find the routing scheme that minimizes a global criteria (efficiency) with individual needs as constraints (fairness).

Suppose we have a requirement of reaching a target value of ϕ for the value of fairness. We encode the fairness requirement with the constraint $\Phi \geq \phi$. In other words, we impose that the flow x satisfies that $\Phi_i \geq \phi$ for all $i \in V$, which can be written as linear constraints. The following linear program (LP) computes an offline solution.

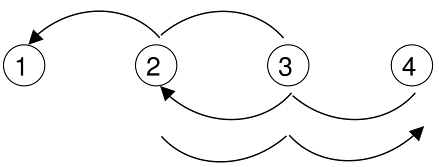

FIGURE 5. Solution x^{SO} corresponding to Example 3.

$$\min \quad \sum_{i \in V} E_i \tag{8a}$$
$$E_i \leq C_i \qquad i \in V \tag{8b}$$
$$\phi E_i^{\mathrm{out}} \leq E_i^{\mathrm{in}} \qquad i \in V \tag{8c}$$
$$\text{and (1), (2), (3), (4)}$$
$$x_{i,j}^{s,t} \geq 0 \qquad (i,j) \in A, (s,t) \in K. \tag{8d}$$

The objective function represents the total energy used by nodes when routing flow according to x. Constraint (8b) makes sure that nodes do not overuse their batteries, and (8c) represents the fairness constraint. Flow conservation is guaranteed by (1), which also ensures that the demand is satisfied. Constraints (2), (3), and (4) define E_i, E_i^{out} and E_i^{in}, respectively.

Definition 5. *We refer to the solution x^{SO} that minimizes the total energy consumption as the* system optimum. *A system optimum can be found by solving problem* (8a)–(8d) *with $\phi = 0$ (i.e., no fairness constraints).*

The solution computed by the LP above should be interpreted as a lower bound for a meaningful routing when fairness constraints (8c) are imposed. Indeed, for fairness requirements that are demanding (high ϕ), the solution of the problem generally routes flow along cycles to satisfy constraints (8c). By taking unnecessary detours, the unbalance of some nodes can be decreased, thus increasing their fairness coefficients and making the flow feasible. The following example illustrates the occurrence of cycles.

Example 3. *Figure 5 shows a network where the most efficient routing includes a cycle. The network is composed of four nodes on a line at a unit distance between each pair of consecutive nodes. The traffic demand is $r_{3,1} = r_{2,4} = r_{4,2} = 1$. Suppose that nodes do not spend energy receiving information ($\rho = 0$), only the sender benefits from retransmissions ($\eta^o = 1$), power control is used, the energy needed to transmit grows super-linearly with distance ($\alpha > 1$), and the maximum distance to transmit is $d_{\max} = 2$. The most efficient solution (shown in the figure) corresponds to flow $x_{3,2}^{3,1} = x_{2,1}^{3,1} = x_{2,3}^{2,4} = x_{3,4}^{2,4} = x_{4,3}^{4,2} = x_{3,2}^{4,2} = 1$, and the total energy equals 6. Replacing in* (1)*, we get that $\Phi = 1/2$ because $\Phi_1 = 1$, $\Phi_2 = 1$, $\Phi_3 = 1/2$, and $\Phi_4 = \infty$.*

With a fairness constraint of $\phi > 1/2$, the previous solution would not have been feasible. To increase the fairness, one can either increase E_3^{in} or reduce E_3^{out}. For example, sending the flow $\Delta/2^{\alpha+1}$ from the OD pair $(3,1)$ along the cycle $(2,4,2)$ increases E_3^{in} and the total energy E spent by the network by Δ, because now nodes 2 and 4 are retransmitting more for node 3. As an alternative, rerouting $\frac{\Delta}{2^\alpha - 2}$ units of flow from path 4-3-2 to path 4-2 reduces E_3^{out}, thus increasing the fairness. However, the increase in fairness when using detours is smaller than that when using cycles for the same increase in the total cost (the cost increase is Δ both with the detour and with the cycle). For that reason the LP will naturally return solutions with cycles.

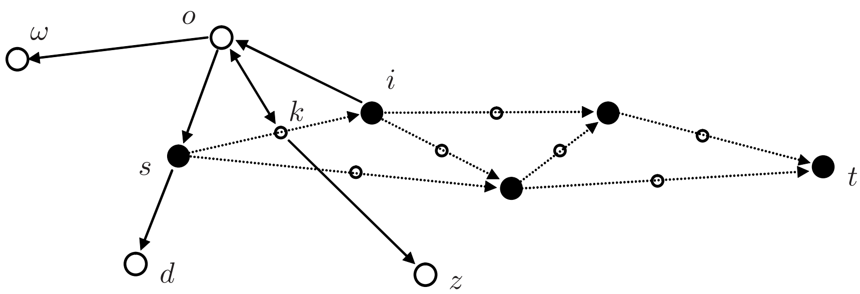

FIGURE 6. Construction of graph G in the proof of Theorem 5. For clarity, we only show a single path from o to z and a single arc from i to o.

As anticipated in Section 3, a routing containing cycles is not meaningful because a path with a cycle is dominated by the same path without the cycle. On the other hand, once the fairness level is fixed, the problem of finding the least-cost routing that only uses meaningful paths turns out to be difficult, as stated in the theorem below.

Theorem 5. *The recognition version of the Hamiltonian s-t-path problem on a directed graph* $\Gamma = (W, B)$ *can be polynomially reduced to checking if there is a routing with* $\Phi = 1$ *that only uses meaningful paths in a related graph.*

Proof. Consider an instance of the Hamiltonian path problem, given by a directed graph $\Gamma = (W, B)$ with a node-set W of size n and an arc-set B of size m, and a distinguished pair of nodes $s, t \in W$. We construct a directed graph $G = (V, A)$ that will be used as an instance for the fair routing problem. The construction is illustrated in Figure 6. Graph G includes all nodes in W and the following additional nodes.

- For each arc $(i, j) \in B$, we add an intermediate node k between i and j. We denote the set of m intermediate nodes by U.
- We add four additional nodes called ω, z, o and d.

Thus, $V = W \cup U \cup \{\omega, z, o, d\}$ and $|V| = n+m+4$. Each arc in Γ gives rise to two arcs in G (there are $2m$ of those arcs). Indeed, each arc $(i, j) \in B$ is split into arcs (i, k) and (k, j). The following list completes the description of A by indicating the outgoing arcs from each node.

- There are arcs (i, o) for all $i \in W \setminus \{s, t\}$ (there are $n - 2$ of these arcs).
- There are arcs (k, o) and (k, z), for all $k \in U$ (there are $2m$ of these arcs).
- There are arcs (o, ω), (o, s) and (o, k) for all $k \in U$ (there are $m + 2$ of these arcs).
- There is an arc (s, d).
- There are no outgoing arcs from nodes d, ω and z.

The cost of all arcs is one. Note that this graph has $5m + n + 1$ arcs and any meaningful path between s and t that does not visit o has cost bounded from above by $2n - 2$ (a path achieving this bound corresponds to a Hamiltonian path in Γ). To complete the definition of an instance of the fair routing problem, we assume that $\rho = \eta^d = 0$ and $\eta_o = 1$, and make the positive entries of the OD matrix be:

- $r_{s,t} = 1$,
- $r_{o,d} = 2n - 3$,
- $r_{o,z} = m - n + 1$, and
- $r_{i,\omega} = 1$ for all $i \in W \cup U \setminus \{s, t\}$.

Notice that, with the exception of s-t, all meaningful paths connecting the OD pairs have length equal to 2, i.e., a single node forwards all traffic along the path. The decisions to be made regarding the constructed instance are how to route flow from s to t and which nodes $k \in U$ to select as intermediate nodes for the OD pair o-z.

We have to prove that a Hamiltonian s-t-path in Γ exists if and only if there is a routing with $\Phi = 1$ that only uses meaningful paths. Let us start with the forward implication. Suppose that there exists a Hamiltonian path in Γ, and denote by P the corresponding path in G. Note that P is meaningful because any two non-adjacent nodes in P cannot be connected since each original arc was subdivided in two. Let us consider the solution to the fair routing problem that routes all the flow from s to t along P and, one unit of flow from o to z using each of the intermediate nodes $k \in U$ that do not belong to P. By construction, all nodes in $W \cup U \setminus \{s,t\}$ retransmit a unit of flow, so $E^{\text{in}} = E^{\text{out}}$ for them. Node s is balanced too since there are $2n-3$ intermediate nodes in P. Finally, node o is balanced too since it is the origin of $n+m-2$ units of flow that are transmitted with the help of s and some nodes in U, and it retransmits the same amount of flow. Consequently, we have found a routing along meaningful paths that satisfies $\Phi = 1$ as required.

To see that the converse also holds, consider a solution with $\Phi = 1$ that only uses meaningful paths. Corollary 2 implies that all nodes are fully balanced. In particular, the network has to spend $2n-3$ units of energy forwarding the unit of traffic originating in node s. This flow cannot go through node o because in that case o will work more for the network than the network will work for it. When flow is routed along a meaningful path, the maximum energy per unit of flow that the network can contribute to s is $2n-3$, and the upper bound is achieved only when flow is routed along a path corresponding to a Hamiltonian path in Γ. Putting this upper bound together with the fact that the network spends a total of $2n-3$ units of energy retransmitting traffic of s implies that the energy spent per unit of flow in *any used path* has to be exactly $2n-3$. Thus, we conclude that all the flow is routed along Hamiltonian paths. In particular, a Hamiltonian path exists. □

Theorem 5 above implies that finding a solution that routes flow along meaningful paths with a given fairness level is NP-complete. Note, though, that the case of $\phi = 0$ is polynomially solvable through the Linear Program (8).

Corollary 6. *Deciding if there exists a routing that only uses meaningful paths and that has a fairness value bigger than or equal to ϕ is NP-complete, for $0 < \phi \leq 1$.*

Proof. Since the Hamiltonian s-t-path problem is NP-complete, the previous reduction implies that finding a routing on meaningful paths with $\Phi = 1$ is NP-complete too. To extend this result to any $0 < \phi < 1$, we must make $r_{s,t} = 1$, $r_{o,d} = (2n-3)/\phi$, $r_{o,z} = (m-n+1)/\phi$, and $r_{i,\omega} = 1/\phi^2$ for all $i \in W \cup U \setminus \{s,t\}$ in the proof of Theorem 5. Proceeding as in the previous proof, the forward implication consists on verifying that the fairness coefficients for all nodes are larger than ϕ. The reverse implication works without modification. □

A natural question is whether avoiding cycles leads to solutions that make use of meaningful paths. The answer to this question is negative: in an optimal solution, cycles are generally replaced by simple, but not meaningful, paths that tend to be unnecessarily long. (The issue of forbidding cycles will be further discussed in Section 6.1.1.) Nevertheless, note that finding acyclic and fair routings is hard too. The proof of the following corollary relies on essentially the same construction used in Theorem 5. (Actually, the proof is slightly simpler because one does not need to split the original arcs in two.)

Corollary 7. *Deciding if there exists a routing that only uses acyclic paths and that has a fairness value bigger than or equal to ϕ is NP-complete, for $0 < \phi \leq 1$.*

5.1. **Empirical Evaluation of the Competitiveness of the Online Algorithm.** This section proposes measures to evaluate the online algorithm described in Section 4. To estimate its performance, we compare the solutions it returns to optimal, offline solutions. As we consider two objectives, we will compare the optimal solutions with respect to each. Let us assume that for a given instance of the problem, algorithm Distr Online Routing has obtained a routing x^{OL}, with cost $E(x^{\mathrm{OL}})$ and fairness $\Phi(x^{\mathrm{OL}})$.

To evaluate the online algorithm, we compare x^{OL} to the solution x^{LP} of (8a)–(8d) with target fairness equal to $\phi = \Phi(x^{\mathrm{OL}})$. Clearly, $E(x^{\mathrm{LP}}) \le E(x^*) \le E(x^{\mathrm{OL}})$, where x^* is the most efficient routing for fairness requirement ϕ that only uses meaningful paths. Although the reference (offline) solution x^* is difficult to compute, the bounds we obtain to approximate it are enough to assert the empirical effectiveness of the proposed algorithm.

Conversely, we are also interested in computing the most fair solution for a maximum energy expenditure of $E(x^{\mathrm{OL}})$. This comparison indicates how well Distr Online Routing does in terms of fairness. Such a solution, denoted by x^{NLP}, can be computed by solving this *Nonlinear Program* (NLP):

$$\max \quad \{\phi : \text{ subject to (1), (2), (3), (4), (8b), (8d) and} \tag{9a}$$

$$\sum_{i \in V} E_i \le E(x^{\mathrm{OL}}) \tag{9b}$$

$$\phi E_i^{\mathrm{out}} \le E_i^{\mathrm{in}}, \qquad \text{for all } i \in V\} \tag{9c}$$

The complexity of system (9) is stated in the following theorem:

Theorem 8. *Problem* (9) *can be solved in polynomial time.*

Proof. Note that for any fixed value of ϕ, the previous NLP reduces to an LP. We already know that $\Phi \in [0, 1]$, which does not depend on the instance. We can use binary search to find the optimal value of ϕ within its feasible range. This works because $\sum_{i \in V} E_i$ is a monotone non-decreasing function of ϕ (this follows from the definition of constraints (8c)). Thus, if the problem is infeasible for a given value ϕ^* during the binary search algorithm, all LPs with $\phi > \phi^*$ are infeasible too. □

As before, we have that $\Phi(x^{\mathrm{OL}}) \le \Phi(x^{**}) \le \Phi(x^{\mathrm{NLP}})$, where x^{**} is the optimal solution to the NLP with constraints that ensure that paths are meaningful.

6. Computational Experiments

The objective of this section is to asses the experimental behavior of the proposed algorithm through a set of computational experiments that we designed and performed. For that purpose we are going to evaluate it comparing the resulting fairness and total energy consumption to those of the offline, centralized algorithm discussed in Section 5. In addition we also compare the resulting number of hops for different solutions as a proxy for the delay of the transmission. With this in mind, we define $L_{\mathrm{avg}}(x)$ and $L_{\max}(x)$ as the average and maximum hop-length, respectively, in a solution x. We will provide evidence that our online algorithm performs very well compared to the offline variants with respect to all objective functions. In particular, we address the following issues:

(1) Solutions of Distr Online Routing are significantly more fair than system optima.
(2) Although solutions returned by Distr Online Routing use more energy than system optima, the amount of extra energy is not dramatic. Hence, the cost of increased fairness is very small.
(3) The design parameter β used by Distr Online Routing allows the system designer to control the tradeoff between total energy consumption and fairness.
(4) The performance of Distr Online Routing, both with respect to energy consumption and fairness, is comparable with the benchmarks given by the optimal solutions to the LP and NLP problems introduced in the previous section.

(5) The end-to-end transmission delay for the solutions returned by Distr Online Routing is comparable to that in a system optimum.
(6) Our algorithm is particularly beneficial to nodes in central locations that would unfairly spend too much energy under a system optimum because all nodes would tend to use them for retransmitting their traffic. These central nodes experience a significant cost reduction.

We start the study with instances inspired by one of the deployment scenarios of the OLPC project, namely a "group of kids sitting under a tree." (see the deployment scenarios of the OLPC project (OLPC 2008b)). This scenario is general enough to represent other real world applications, like those mentioned in Section 1. Laptops are placed randomly in a square, meant to represent a village. Because laptops collaborate with each other, we look at all-to-all OD matrices and the benefits of a transmission is shared equally between the origin and the destination. In a second part, we look at what OLPC calls a school mesh, which adds an access point to the Internet to the previous scenario. Since an access point is assumed to have a permanent source of energy, we modify the definition of fairness slightly. For these instances, we also modify the OD matrix to reflect the possibility of connecting to the Internet. We provide more details in the corresponding sections.

6.1. **Simple-Mesh Instances.** We kept the number of design variables to a minimum to understand the major effect that our algorithm induces on the fairness level and energy consumption, as opposed to blurring results with more variability that comes from other design factors. A simple mesh is an instance in which laptops only want to communicate with each other. To construct a mesh, we randomly place n nodes on a unit square using a uniform distribution. We consider values of n equal to 10, 20 and 30 to study the dependence between our algorithm, the instance size and the density of the nodes. For each combination of parameters, we construct 20 independent instances to make sure that the reported results are significant.

We use a uniform all-to-all demand matrix, meaning that each node i communicates with every node $j \neq i$. The total demand for each OD pair is normalized to 1. Although we do not report the results here, we obtained similar conclusions for other types of demand matrices such as random uniform demands, all-to-one, and more complicated ones that try to mimic realistic scenarios. The initial battery levels in the nodes were set to large values so capacity constraints are not binding in these experiments. The motivation for this is that the periods of the online algorithm are in a different time scale than the time needed to deplete a battery. For that reason, in this study we preferred an analysis without the complication of feasibility issues. Finally, we set the benefit extracted by the origin and destination nodes of the transmission to $\eta^o = \eta^d = 1/2$.

For this computational study, costs are updated using the function $f(p, u) = pu^{-\beta}$ (see (7)), where β is a positive constant that indicates the importance given to the fairness of the final solution. We consider a planning horizon T of 50 periods, in which the traffic rate is constant. (This can model, e.g., Fixed Bit Rate traffic sources such as programs that send voice-over-IP). We run Distr Online Routing for each value β in the set $\{0.3, 0.5, 0.8, 1, 1.5, 2, 3, 4, 6, 8\}$ and find the corresponding solution x^{OL}. We compare the objective values $E(x^{\mathrm{OL}})$ and $\Phi(x^{\mathrm{OL}})$ to those of the system optimum and to those of problems LP and NLP, described in Section 5.1. However, recall that benchmark solutions only provide bounds; computing them requires information not known in advance. Moreover, these solutions may be inefficient because they may contain cycles and dominated paths.

We first focus on the case in which nodes transmit using fixed power. We assume that sending a unit of information on an arc $(i, j) \in A$ requires that nodes i and j use $p_{i,j} = 1$ and $\rho = 1/3$ units of energy to send and to receive that information, respectively. We set the maximum transmitting power $\pi_{\max}$ to allow every node to reach all other nodes that are not further away than 4/10 of the diagonal of the square where nodes are located. We chose this threshold because higher values

TABLE 1. System optima, no power control.

n	$E(x^{\mathrm{SO}})$	$\Phi(x^{\mathrm{SO}})$	$L_{\mathrm{avg}}(x^{\mathrm{SO}})$	$L_{\mathrm{max}}(x^{\mathrm{SO}})$
10	185.883	0.296	1.55	2.88
20	755.733	0.225	1.49	2.95
30	1698.134	0.199	1.46	2.95

TABLE 2. Performance of DISTR ONLINE ROUTING, no power control.

n	β	$\Phi(x^{\mathrm{OL}})$	$\mathrm{gap}(x^{\mathrm{OL}})$	$\mathrm{gap}(x^{\mathrm{LP}})$	$\Phi(x^{\mathrm{NLP}})$	$L_{\mathrm{avg}}(x^{\mathrm{OL}})$	$L_{\mathrm{max}}(x^{\mathrm{OL}})$
10	0.5	0.424	0.1	0.0	0.437	1.55	3.12
	1.0	0.432	0.6	0.2	0.454	1.56	3.41
	3.0	0.458	2.2	0.8	0.513	1.58	3.82
	6.0	0.469	3.4	1.1	0.557	1.60	3.88
20	0.5	0.544	0.1	0.0	0.566	1.49	3.15
	1.0	0.562	0.6	0.2	0.601	1.50	3.60
	3.0	0.615	3.6	0.9	0.733	1.55	4.65
	6.0	0.633	7.1	1.3	0.855	1.60	5.00
30	0.5	0.650	0.0	0.0	0.697	1.47	3.05
	1.0	0.665	0.3	0.0	0.730	1.47	3.20
	3.0	0.738	2.5	0.4	0.856	1.50	4.35
	6.0	0.775	5.8	1.7	0.940	1.55	5.05

of $\pi_{\max}$ allow each node to communicate directly to its destination with a unit cost and maximum fairness, while networks may be disconnected for lower thresholds because nodes will have only a few neighbors to communicate. Note that by keeping this value constant across instances with different number of nodes, we obtain networks with different densities. Out of the 60 random networks that we generated, only 3 of them were disconnected (all had 10 nodes); we discarded them to prevent problems of infeasibility.

Secondly, we consider the case in which nodes perform power control. We assume that transmitting power increases quadratically with distance ($\alpha = 2$), and that the maximum power $\pi_{\max}$ is such that all nodes can reach all other nodes. Finally, we set ρ equal to a third of the energy needed to send a unit of information on an arc of length 1/10.

6.1.1. *Analysis of the Experiments with Fixed Power.* Considering the instances we generated, we run DISTR ONLINE ROUTING and compute solutions to the system optimum problem and to the relaxations described in Section 5. Tables 1 and 2 summarize these solutions. Notice that all numbers in the tables refer to the averages of the corresponding quantities over all instances. Table 1 reports, for each instance of size n, the objective functions at system optima: the energy consumption $E(x^{\mathrm{SO}})$, the fairness value $\Phi(x^{\mathrm{SO}})$, the average hop-length $L_{\mathrm{avg}}(x^{\mathrm{SO}})$ and the maximum hop-length $L_{\mathrm{max}}(x^{\mathrm{SO}})$.

Table 2 reports the characteristics of solutions returned by DISTR ONLINE ROUTING, for each instance size and each value of $\beta \in \{0.5, 1.0, 3.0, 6.0\}$. (We do not report the other values of β because results are similar.) For a given solution x, the value $\mathrm{gap}(x) = 100(E(x) - E(x^{\mathrm{SO}}))/E(x^{\mathrm{SO}})$ measures the relative gap between solution x and the system optimum with respect to energy consumption. Each line reports the fairness value $\Phi(x^{\mathrm{OL}})$ and $\mathrm{gap}(x^{\mathrm{OL}})$. Solution x^{LP} is computed by solving problem (8) with the constraint that the solution is at least as fair as x^{OL}. This quantifies how much more efficient one can be with the same fairness value. As anticipated in Section 5, forbidding cycles in system (8) is not only NP-hard but also computationally ineffective. Indeed, in our computational experience, the optimal solution with no cycles spends only slightly more energy than $E(x^{\mathrm{LP}})$ with the same fairness because it uses detours to achieve an effect that

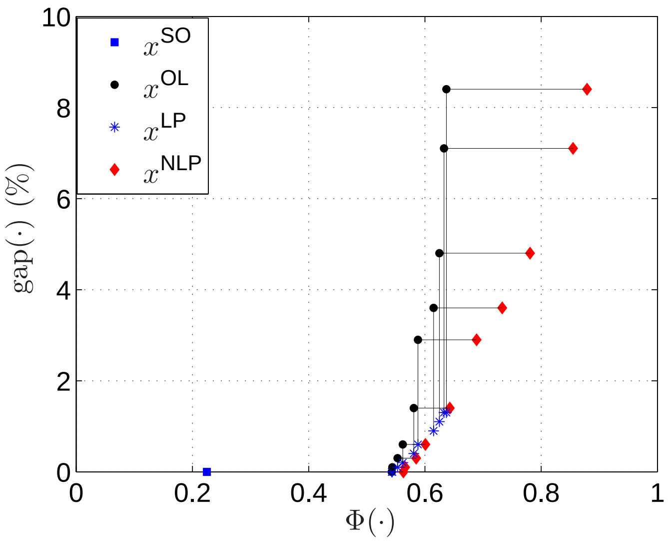

FIGURE 7. Tradeoff between energy consumption and fairness, no power control.

is similar to using cycles. Since $E(x^{\text{LP}})$ (or potentially the energy spent in a solution without cycles) is only used as a lower bound, the significant increase of computational effort involved in computing the solution without cycles does not pay off. Similarly, solution x^{NLP} is computed by solving problem (9) with the constraint that energy consumption is at most that of x^{OL}. This quantifies how fair one can be without spending more energy than in the online solution. The last two columns are similar to those in Table 1.

Recall that energy has to be minimized while fairness has to be maximized. With respect to the energy, the lowest possible value is the energy of the system optimum x^{SO} (e.g., on average 185.883 for networks of 10 nodes), while the largest is the energy of x^{OL} (e.g., for the case of $\beta = 3.0$, this represents an increase of 2.2% for the same group of instances). Between the two extremes, x^{LP} (with and increase of 0.8% over the system optimum) uses less energy than a solution that only uses meaningful paths x^{EMP}. The latter solution would be the best offline solution, but we cannot compute it. Although we do not know its energy consumption, it is between x^{LP} and x^{OL} (which are 1.4% apart for the group we use as an example). Concerning fairness, we have that x^{NLP} is the most fair solution (with a fairness of 0.513 for networks of 10 nodes and $\beta = 3.0$; recall that this routing could even be infeasible). The solution denoted by x^{FMP} maximizes fairness and only uses meaningful paths, but we cannot compute it either. These are followed by x^{OL} (with a fairness value of 0.458) and the least fair is x^{SO} (with a value of 0.296).

The relation among the objective values considered in this section is:

$$E(x^{\text{OL}}) \geq E(x^{\text{EMP}}) \geq E(x^{\text{LP}}) \geq E(x^{\text{SO}})$$

$$\Phi(x^{\text{NLP}}) \geq \Phi(x^{\text{FMP}}) \geq \Phi(x^{\text{OL}}) \geq \Phi(x^{\text{SO}}),$$

for energy and fairness, respectively.

To complement the previous relations, Figure 7 depicts averages of solutions computed by DISTR ONLINE ROUTING for networks with 20 nodes, for the various values of β. Circles represent solutions x^{OL} and the square represents the system optimum x^{SO}. Stars correspond to x^{LP} in Table 2, while diamonds correspond to x^{NLP}.

Both Table 2 and Figure 7 capture the tradeoff between the energy consumption and the fairness of solutions. As it can be seen, DISTR ONLINE ROUTING always improves the value of fairness with respect to the system optimal solution. This improvement in fairness is higher with larger values of β because this signals the algorithm that fairness is more important. In addition, for the same value of β, the increase in fairness is larger for larger networks. This is because the larger networks are more dense and, consequently, there are more feasible routing patterns for the same value of $\pi_{\max}$. Although there is a price to pay for obtaining higher fairness, for the instances we considered, the additional energy consumption is always, averaging over random instances with the same characteristics, smaller than 7.1%. (This is the maximum over the column gap(x^{OL}) in Table 2; the largest gap(x^{OL}) among the random networks considered in the experiments is 12.6%.) The energy increase is due to both making the solution fairer and to the lack of complete information when the online algorithm starts.

This worst-case bound is computational and provides an empirical analysis of the average quality of solution x^{OL} provided by the online algorithm with respect to an optimal solution. We have used the solution to the offline problem x^{SO}, as it is typically done in the literature of online algorithms when measuring the competitive ratio (Borodin and El-Yaniv 1998). Indeed, when evaluating an online algorithm, typically one quantifies the worst-case ratio between the solution provided by the online algorithm to the optimal offline one. Another option, had we chosen to do a comparison with a solution that explicitly considers fairness, would have been to consider the most efficient routing that only uses meaningful paths x^{EMP} as the offline solution. The previous bound is also related to the concept of price of anarchy in the literature of algorithmic game theory (Nisan et al. 2007), defined as the worst-case ratio between an equilibrium of a game and the solution that minimizes cost. Here, the solution x^{OL} plays the role of an equilibrium because it is a solution that users do not have strong incentives to reject, and the solution maximizing welfare is given by x^{SO} because it minimizes energy consumption disregarding the issue of fairness and the possibility of rejection by the users.

It is important to remark that, for small values of β, Distr Online Routing increases the fairness level significantly compared to the system optimum without increasing the required energy by a large extent. In other words, among the feasible solutions of approximately minimum cost, the algorithm selects one with a large fairness value. Moreover, the information, coordination and computation requirements are much milder than those that allow one to compute a system optimum because the algorithm is online, distributed and simple. To see this, let us look, e.g., at the row corresponding to $n = 30$ and $\beta = 0.5$ in Table 2: for instances with 30 nodes, our algorithm provides solutions with an average fairness of 0.650, without increasing the energy consumption. In comparison, the average fairness of system optima is 0.199 (see Table 1).

Comparing columns gap(x^{OL}) and gap(x^{LP}) in Table 2, we see that Distr Online Routing computes good solutions for the fairness level of x^{OL}. Indeed, the gap between x^{OL} and x^{SO} is not much larger than that between x^{LP} and x^{SO}. This can also be observed in the figure by comparing the curve formed by stars to that formed by circles. Our solutions use at most 5.8% more energy than necessary. (Recall that all gaps are expressed as a percentage of the energy in the system optimum.) The number 5.8% is the difference between gap(x^{OL}) and gap(x^{LP}) for $n = 20$ and $\beta = 6.0$. Comparing x^{OL} with x^{NLP}, we see that when the total energy consumption is constant, in the worst case there is a difference of 0.222, which is the difference between $\Phi(x^{\mathrm{OL}})$ and $\Phi(x^{\mathrm{NLP}})$, again for $n = 20$ and $\beta = 6.0$. When we consider small values of β, x^{OL} is very close to x^{LP} and x^{NLP}, meaning the our algorithm is nearly optimal. For example, for $\beta = 0.5$, the largest difference between $E(x^{\mathrm{OL}})$ and $E(x^{\mathrm{LP}})$ is 0.1%, and the largest difference between $\Phi(x^{\mathrm{OL}})$ and $\Phi(x^{\mathrm{NLP}})$ is 0.047.

Finally, we measure the quality of solutions with respect to the average and maximum number of hops, which are a proxy for transmission delay. The average number of hops for the solutions returned by Distr Online Routing is very close to those under a system optimum. The average increases with β because more fair solutions tend to use detours to reduce the unbalance of nodes.

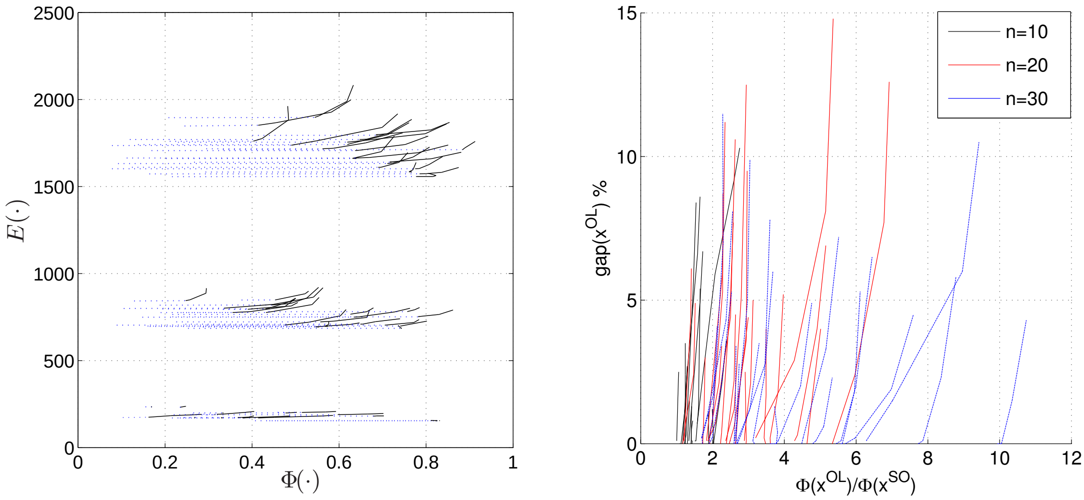

FIGURE 8. Tradeoff between efficiency and fairness, no power control.

These detours tend to be longer than the most direct routes used by system optima. Actually, the increase in the number of hops with respect to system optima is proportional to the increase in energy. For the same reason, the maximum number of hops also increases with β. Although the rate of increase is bigger than that of the average, the maximum number of hops is never higher than 72% more than that of the system optimum ($n = 30$ and $\beta = 6$). This bound provides an alternative way to measure the fairness of a flow (Roughgarden 2002; Jahn et al. 2005; Correa et al. 2007). In essence, the parameter β of the algorithm gives the network designer the power to select between efficient but relatively unfair solutions (although more fair than a system optimum), and fair solutions that are more inefficient. This effect is depicted clearly in Figure 8. This figure complements the graphs and tables presented earlier by showing all instances separately, and thus allowing one to see the variability.

The left graph in Figure 8 shows the energy consumption and fairness value in the case of no power control. The left-most points represent system optima, which are connected with a dotted line to the solutions corresponding to the lowest β. Then, continuous lines connect solutions obtained for increasing values of β. Note that, for most instances, the solution with lowest β is much more fair than the system optimum, without increasing the energy (the dotted lines are almost horizontal). The three clusters that appear clearly in the figure correspond, from bottom to top, to networks of 10, 20 and 30 nodes. Some of the curves corresponding to networks with 10 nodes look like a point because those instances admit few feasible solutions and more fair solutions are hard to find. Larger networks are more dense making this very unlikely.

The right graph in the figure represents the same information but in relative terms. The vertical axis displays the gap while the horizontal axis displays the fairness ratio between x^{OL} and x^{SO}. The curves, whose styles depend on the size of the network, connect solutions of the same instance for increasing values of β. As we indicated previously, our algorithm tends to improve more on the fairness value when instances are large (there are many dotted lines at the right of the picture and there are many continuous lines at the left). Considering all instances we generated, gap(x^{OL}) is always less than 15%, and quite often under 10%, while the fairness of x^{OL} solutions is often at least twice those of system optima.

6.1.2. *Analysis of the Experiments with Power Control.* Tables 3 and 4 and Figure 9 summarize the solutions we obtained for the case when nodes can control the transmission power. They are all similar to those in the previous section: tables report averages over 20 instances and the figure

TABLE 3. System optima, with power control.

n	$E(x^{\text{SO}})$	$\Phi(x^{\text{SO}})$	$L_{\text{avg}}(x^{\text{SO}})$	$L_{\max}(x^{\text{SO}})$
10	18.369	0.269	2.46	5.45
20	59.087	0.241	3.27	8.00
30	112.603	0.219	4.04	10.35

TABLE 4. Performance of DISTR ONLINE ROUTING, with power control.

n	β	$\Phi(x^{\text{OL}})$	$\text{gap}(x^{\text{OL}})$	$\text{gap}(x^{\text{LP}})$	$\Phi(x^{\text{NLP}})$	$L_{\text{avg}}(x^{\text{OL}})$	$L_{\max}(x^{\text{OL}})$
10	0.5	0.431	5.8	1.9	0.584	2.40	6.10
	1.0	0.536	11.6	4.6	0.777	2.35	6.25
	3.0	0.759	22.8	11.0	1.000	2.26	6.50
	6.0	0.861	29.9	14.4	1.000	2.23	6.75
20	0.5	0.415	6.3	1.4	0.645	3.39	9.55
	1.0	0.535	12.6	3.6	0.832	3.33	9.65
	3.0	0.741	26.3	9.3	1.000	3.25	10.20
	6.0	0.845	39.9	13.2	1.000	3.16	10.20
30	0.5	0.381	6.1	1.0	0.631	4.31	12.80
	1.0	0.505	13.5	3.1	0.842	4.24	12.80
	3.0	0.737	30.0	9.5	1.000	4.08	13.00
	6.0	0.847	46.4	13.7	1.000	3.94	13.15

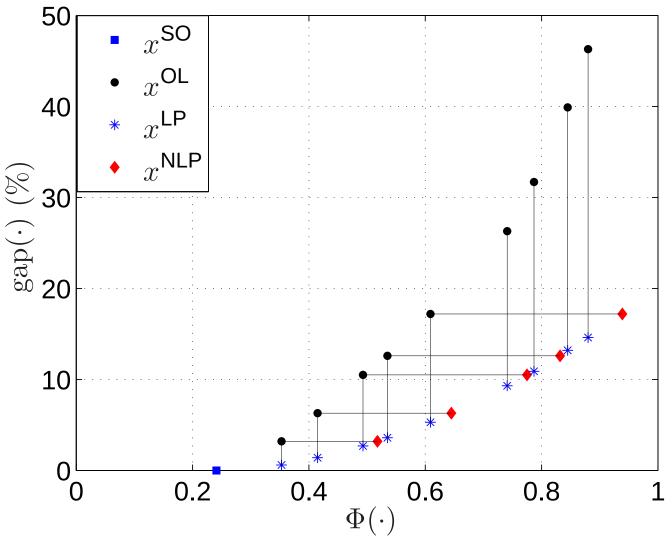

FIGURE 9. Tradeoff between energy consumption and fairness, with power control.

represents average values for networks with 20 nodes. Some diamonds in the figure are not displayed because there are solutions with fairness equal to one and lower energetic cost (see below).

The case with power control is considerably more difficult because the energy needed increases super-linearly with distance making it less efficient to retransmit using nodes that are not the nearest neighbor. Nevertheless, DISTR ONLINE ROUTING still improves the fairness value with respect to that in the system optimum. This improvement increases with bigger values of β, and it does not seem to depend on the network size for the same value of β. Although the cost to pay for increased fairness can be much larger than when power control is not used, the increase itself is not very large. Indeed, the relative gap to the system optimum never exceeds 46.4%, which corresponds to $n = 30$ and $\beta = 6$. However, as opposed to the case of no power control, a small improvement of fairness requires a noticeable increase in energy consumption. For example, for $n = 20$, to improve from $\Phi = 0.241$ at a system optimum to $\Phi = 0.415$ requires spending 6.3% more energy. This is because as costs depend on the length of arcs, it is unlikely to have many shortest paths between a pair of nodes.

Comparing the gaps and curves in the figure, Distr Online Routing still delivers good solutions, albeit less efficient than without power control. Although the circles increase faster than the stars and diamonds, all curves have the same rate of increase. The worst-case gap between x^{OL} and x^{LP} is 32.7% ($n = 30$ and $\beta = 6.0$). A value of 1.0 in column $\Phi(x^{\text{NLP}})$ means that there is a solution x^{NLP} that uses at most $E(x^{\text{OL}})$ energy with fairness equal to one. These solutions correspond to the missing diamonds in Figure 9.

Concerning the end-to-end latency of transmissions, averages of number of hops for solutions returned by Distr Online Routing are very close to those of system optima. Contrary to the case of no power control, the average number of hops decreases with the fairness value. This fact can be interpreted as follows: paths tend to be composed of many short hops at a system optimum because this reduces the energy consumption (without power control, it was more efficient to use the longest arcs that were present in the network). In this situation, central nodes end up forwarding traffic that originates from nodes far from the network center. A more fair solution reduces the contribution of central nodes to the network by using longer arcs that skip those nodes. An indirect effect of this is a reduction in the number of hops. Moreover, we observe that the maximum number of hops under a solution x^{OL} is slightly larger than in a system optimum. This value increases with β, but never exceeds 28% of the same parameter at x^{SO} ($n = 20$ and $\beta = 6$). Recall that the case without power control was more unfair.

Figure 10 is the counterpart of Figure 8, and displays the same curves for the case of power control. The left graph shows the clusters formed by networks of the same size. As we already mentioned, it is more expensive to achieve a higher value of fairness when power control is used. The graph in the right shows that all instances are clustered together when one looks at relative values. This implies that, when power control is used, bigger instances are not easier; regardless of the network size one needs to pay the same relative price in efficiency to gain the same relative fairness. Nevertheless, for all instances, one can obtain a solution with at least twice the fairness of the corresponding system optima without paying more than 30% more.

6.1.3. *Analysis of the Instances.* In the previous sections, we have studied empirically the performance of Distr Online Routing on randomly generated instances with uniform traffic. In addition, the example given right before Section 1.1 shows that there are networks in which a fair solution does not exist under any reasonable definition of fairness—i.e., these networks are *structurally unfair*—which is independent of the ability of our algorithm to find a good compromise between efficiency and fairness. A natural question to ask is whether there are networks such that a simultaneously efficient and fair routing exists but our algorithm fails to find it. To answer this question we are going to classify instances in different scenarios, based on features that influence the fairness and efficiency achieved by the algorithm. We separate the discussion between topological and traffic demand characteristics.

a. *Topology of the Network.* We center this discussion around two dimensions related to the network topology: size and density. When either dimension is high (i.e., big or dense), there tend to be more paths present that join any OD pair of the demand matrix. In that case, it is likely that several disjoint (or at least partially so) routes that are efficient can be chosen to reach a fair outcome. For instance, the experiments of the previous subsections confirm

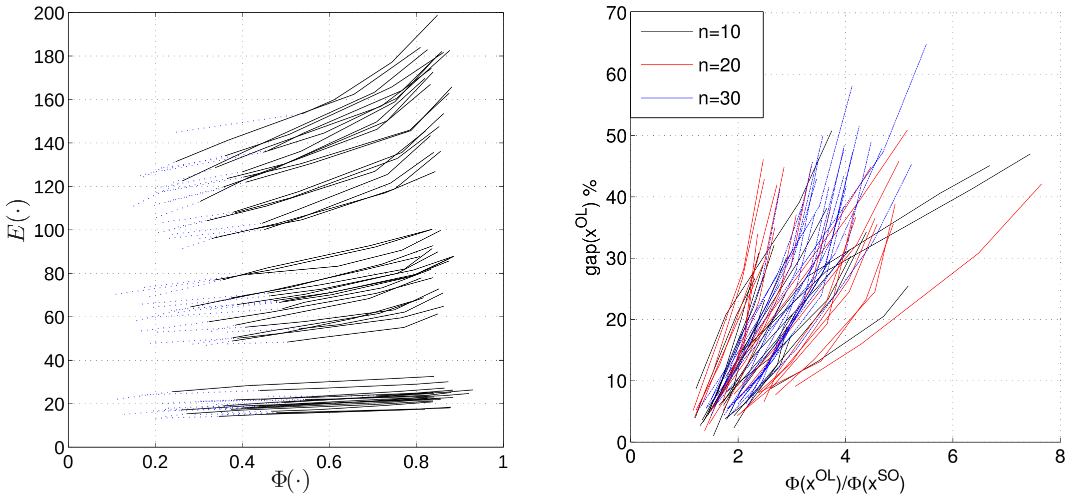

FIGURE 10. Tradeoff between efficiency and fairness, power control.

that when networks are larger, DISTR ONLINE ROUTING finds these alternative paths and reach fairer solutions. Because in the definition of fairness we adopt a max/min criterion (remember that the definition given in Section 3 considers the minimum fairness coefficient among the users as network fairness), having a small number of paths connecting even a single OD pair would make it more difficult to achieve a fair solution.

When a single efficient path exists for an OD pair, like in the worst-case example in Section 1, sparsity seems to introduce a structural difficulty when combined with an unbalanced traffic demand. When the alternatives are not sufficient, fair solutions are unlikely to exist or are extremely inefficient, independently of our algorithm.

Concerning settings where a fair routing exists but DISTR ONLINE ROUTING is unable to find it, remember that DISTR ONLINE ROUTING computes shortest paths according to the modified costs k, thus improving the values of the users' fairness coefficient "on average" (as opposed to the fairness definition which considers the user in the worst condition). As an example, suppose that node h has the lowest fairness coefficient of the network, i.e. $\Phi_h = min_{i \in V} \Phi_i = \Phi$. The only way to improve the fairness of the network is increasing Φ_h, which is indirectly done by DISTR ONLINE ROUTING by increasing the costs of the arcs leaving node h. However, if node h is enclosed by a set of nodes with large fairness coefficient, it may still happen that the shortest paths for some OD pair visit h, and fairness is not improved, unless the value of the coefficient β is set to a vary large value, but in this case the original costs of the arcs are completely lost and DISTR ONLINE ROUTING can be very inefficient. Although this situation can be easily reproduced in networks designed on purpose, we did not observe such a behavior in the randomly generated networks considered in our computational study. Finally, one may observe that a possible way to overcome the potential problem described in this paragraph is to compute paths that maximize the minimum fairness of involved nodes (instead of shortest path). However, such a choice would lead to completely inefficient routings.

b. *Traffic Demand Matrix.* Both the spatial and the temporal dimensions in the traffic demand matrix structure may influence the performance of our algorithm. Regarding spatial imbalances, there can be a large variability in the quantity of information transmitted between nodes. As discussed in the examples given in Section 1, spatial variability may preclude the existence of a fair solution. In this case, the instance is structurally unfair, which is independent of the performance of our algorithm.

Second, let us consider the temporal dimension. Instead of being constant as we have considered so far, the traffic demand may vary over time, possessing arbitrarily complicated statistical properties ranging from being independent and identically distributed in each period to being non-stationary, auto-correlated, etc. This represents another type of imbalance that may lead Distr Online Routing to return solutions that are not fair and efficient. This happens because our algorithm may not be able to learn the current demand pattern fast enough to set the weights in arcs and create an equitable situation. If changes are sudden, demand may have already changed, invalidating the weights that are being used to route flow. The main issue is that, like most online algorithms, our algorithm cannot compare very well to an all-powerful competitor that knows the future realizations of a random process, and the more variability in this random process, the bigger this is a problem.

We have performed a set of computational experiments to assess the effectiveness of Distr Online Routing when the demand is stochastic. Remember that in our previous experiments we considered a constant total demand equal to 1 divided in a planning horizon of 50 periods. In a first set of experiments, we take an all-to-all OD matrix where the demand per period between each OD pair is taken from a uniform distribution with *expected* value equal to 1/50 (hence the expected total demand is 1). The results we obtain are very similar (almost the same) to those reported in the previous sections. In a second set of experiments we consider non-stationary distributions and partition the time horizon into three different macro-periods. We keep the total expected demand equal to 1 but make the expected value per period equal to 1/6, 1/2 and 1/3, respectively for each macro-period. This simulates situations with dynamic traffic characteristics, e.g., evolving during the day. Again, we obtain results that are very close to those reported in the previous sections.

Finally, we also do experiments with a reduced number of periods (i.e., the number of times Distr Online Routing updates costs). For both stationary and non-stationary demands, we observe that the quality of solutions gets worse. As an extreme example, when the number of periods is three, solutions are inefficient and just slightly fairer than a system optimum. Comparing these solutions to those returned by our algorithm when a larger number of periods is used, they are worse with respect to both objectives. Our conclusion is that although these instances posses fair and efficient solutions (which can be found when more time periods are used), our algorithm fails to find them because it does not have enough time to learn the topology and demand in the network and adapt.

Section 4 showed that Distr Online Routing always selects meaningful paths. However, only using meaningful paths does not guarantee that a routing is efficient. For instance, the example depicted in Figure 3 uses meaningful paths but the routing is dominated. It is reasonable to ask whether it may happen that for a particular traffic configuration, our algorithm returns flows that are dominated for all time periods. Although it is not difficult to design a traffic demand matrix that forces Distr Online Routing to use paths that form a dominated routing, we could not identify any traffic pattern for which this happens for more than a limited time. Indeed, after few periods some paths become expensive and flow is rerouted, achieving better efficiency and a heightened value of fairness. This support the conclusion of the previous paragraph: enough time is required to learn.

6.2. **An 'On-Off' Algorithm.** In the previous sections we have shown that Distr Online Routing performs very well for the artificial instances that we have designed. Our instances are purposely

Table 5. Performance of the algorithm 'On-Off'.

		no power control			*power control*		
n	δ	$\Phi(x^{\mathrm{SO}})$	$\Phi(x^{\mathrm{OL}})$	$\mathrm{gap}(x^{\mathrm{OL}})$	$\Phi(x^{\mathrm{SO}})$	$\Phi(x^{\mathrm{OL}})$	$\mathrm{gap}(x^{\mathrm{OL}})$
10	0.4	0.296	0.341	1.7	0.269	0.489	16.2
	0.3		0.352	1.6		0.534	21.0
	0.2		0.353	2.0		0.581	27.4
	0.1		0.356	1.2		0.637	36.9
	0.05		0.352	1.0		0.663	42.6
20	0.4	0.225	0.375	3.3	0.241	0.520	20.3
	0.3		0.393	3.9		0.561	27.5
	0.2		0.393	3.7		0.605	37.6
	0.1		0.385	4.5		0.655	50.1
	0.05		0.353	4.3		0.684	59.3
30	0.4	0.199	0.393	1.4	0.219	0.524	23.5
	0.3		0.423	1.9		0.569	31.2
	0.2		0.454	3.2		0.617	41.8
	0.1		0.468	5.1		0.650	58.5
	0.05		0.450	5.9		0.680	69.1

kept as simple as possible in order to asses the performance in a neutral setting; more realistic instances will be considered in the next section. A natural question, however, is whether an algorithm that forbids the nodes that already contributed "too much" to the network from forwarding would achieve a high level of fairness. This might seem plausible, especially in a settings with uniform communication requirements as ours.

To answer this question we implemented the algorithm 'On-Off' that works as follows. At every time-period, the algorithm computes the average energy $\overline{E}_{net}$ spent by nodes since the beginning. This variable includes the energy spent for themselves and for other network nodes. If the energy E_i spent so far by node i exceeds $\overline{E}_{net}$ by more than a given threshold δ, the node stops forwarding traffic, unless this would disconnect the network. In other words, if $E_i > \overline{E}_{net}(1+\delta)$ for a value of $\delta > 0$ set a-priori, node i stops forwarding others' traffic. A node will restart forwarding traffic again when $E_i \leq \overline{E}_{net}(1+\delta)$. We disregard nodes that do not forward traffic at the moment, and route all traffic along the shortest paths that are available.

By choosing a small value for the parameter δ, one can try to make all nodes spend approximately the same energy. Intuitively, this would have the effect of making the energy spent by all nodes similar, thereby generating large values of fairness. Since central nodes will have to forward more packets, one can expect to obtain more efficient but less fair solutions when choosing large values of δ.

Table 5 reports the results obtained with this algorithm: for the same networks as in the previous sections, we report the fairness level $\Phi(x^{\mathrm{OL}})$ and the increase in energy consumption $\mathrm{gap}(x^{\mathrm{OL}})$ achieved for values of δ in the set $\{0.05, 0.1, 0.2, 0.3, 0.4\}$. The left part of the table considers the case without power control, and the right part considers the case with power control. The fairness values $\Phi(x^{\mathrm{SO}})$ of system optima are copied from Tables 1 and 3, respectively.

As one might have expected, the fairness of solutions delivered by this algorithm is higher than under optimal solutions. However, the performance of this algorithm is much lees satisfactory than that of Distr Online Routing. In the case of no power control, the algorithm is not very responsive to changes to parameter δ, and Distr Online Routing obtains a larger fairness for the same energy increase ($\mathrm{gap}(x^{\mathrm{OL}})$). In the case of power control, the algorithm is more sensitive to the value of δ, and one can actually obtain fairer solutions by reducing it. However, the increase of

Table 6. Performance of Distr Online Routing in a school mesh setting

β	$\Phi(x^{\mathrm{OL}})$	$\mathrm{gap}(x^{\mathrm{OL}})$
0.5	0.403	5.7
1.0	0.488	10.7
3.0	0.650	22.9
6.0	0.744	32.8

energy ($\mathrm{gap}(x^{\mathrm{OL}})$) is very large for the increase of fairness. For networks of 30 nodes with $\delta = 0.05$, on average we have that $gap(x^{\mathrm{OL}}) = 69.1$ and $\phi = 0.680$, while Distr Online Routing with $\beta = 6.0$ achieves $gap(x^{\mathrm{OL}}) = 46.4$ and $\phi = 0.847$ for the same networks. We also observed that for values of δ larger than 0.4, the algorithm tends to find solutions closer to the SO when δ increases (i.e., the algorithm is less effective in computing alternative routings). These numbers put the results of the previous sections in perspective by providing evidence that not any algorithm can achieve the same level of fairness for the same increase in energy consumption.

6.3. **School-Mesh Instances.** In the experiments of the previous sections, we decided to keep the instances as simple as possible to not blur our analysis with a large amount of variables. This section is intended to show that our analysis is still valid in a setting that is closer to a real-world situation than the previous sections. We study a more realistic setting involving randomized traffic and special nodes that act as access points to external resources. These instances model what the OLPC project calls a *school mesh* and the access points mentioned above are normally referred to as *school servers*. Laptops communicate by forming a mesh network as before. Access points, which we assume connected to the power grid, can access external resources such as servers, an Internet connection, etc. Some laptops can reach the access points directly; the rest can establish a communication indirectly using the mesh network.

We consider instances with 20 laptops and 2 access points. Laptops perform power control, and the maximum transmitting power $\pi_{\max}$ allows every laptop to reach all other laptops that are not further away than 1/4 of the edge of the square where they are located. We assume that the power needed to transmit increases quadratically with distance ($\alpha = 2$). The OD matrix is not all-to-all, as in the previous section, but randomly constructed. With a probability of $p_t = 0.6$, a laptop i wishes to transmit to other laptops in the mesh. In that case, with an independent probability of $p_r = 0.7$, i needs to talk to j for all $j \neq i$. In addition, we consider that each laptop has to receive traffic from one of the two access points, representing Internet use. The rate of each OD pair (laptop-laptop and access point-laptop) is a random value with uniform distribution in the interval $[0.5, 1]$. Since the traffic routed through access points to the laptops originates outside the mesh and benefits the receiver only, we set $\eta^o = 0$, and $\eta^d = 1$.

Following standard user choice models, we assign laptops to access points following a Logit model (Ben-Akiva and Lerman 1985). Under this model, the probability that laptop j connects to access point i is proportional to $e^{-p_{i,j}}$. Because energy is not a scarce resource for access points, they communicate with directly-reachable laptops without retransmissions. In addition, we do not consider the energy spent by access points when computing energy consumption, nor include this energy in any fairness calculation (in practice, we set $p_{i,j} = 0$ for all existing arcs (i, j) where i is an access point).

As done previously, we generate 20 random replications of the described setting, and report average values. The locations of the 20 laptops and the 2 access points follow a random distribution in the unit square. On average, 7.1 laptops are outside the reach of any access point, and thus need intermediate nodes to communicate with them. Table 6 reports the fairness level $\Phi(x^{\mathrm{OL}})$ and the increase in energy consumption $\mathrm{gap}(x^{\mathrm{OL}})$ achieved by Distr Online Routing for values of β in the set $\{0.5, 1, 3, 6\}$. As a comparison, the average fairness of a system optimum $\Phi(x^{\mathrm{SO}})$ is 0.253.

The results presented in Table 6 confirm that the Distr Online Routing also works well in a more realistic experiment. As before, the developed algorithm can compute fair solutions with a limited increase in the total energy consumption, and the design parameter β allows the system designer to control the tradeoff between total energy consumption and fairness. The increase in energy consumption ranges from 5.7% to 32.8%, and the fairness level ranges from 0.403 to 0.744, which represents a noticeable improvement with respect to that of the system optimum. Finally, let us highlight that we considered different mixes for the traffic rates between laptops and between access points and laptops. For each mix, we obtained very similar results.

7. Implementation Issues

The objective of this research was to derive insights on how exposed to unfairness nodes in mesh networks can be. Because of their unstructured nature, it is quite difficult to agree on what constitutes a typical mesh network, including topology, size, demand pattern, etc. For this reason, and because we did not want to make our analysis dependent of a specific architecture, we adopted a macroscopic flow model that is general enough to include most transmission paradigms in use in the real world. Although the implementation *per se* is out of the scope of this paper, this section briefly discusses some issues related to it. The objective is to show that our ideas can be implemented in a real protocol without extensive modifications and, more importantly, that it can be done without introducing a significant communication overhead.

To implement Distr Online Routing, each node is required to compute its fairness coefficient. To evaluate (5), nodes need to know how much energy they spent on retransmissions, which is measurable locally. In addition, each node needs the amount of energy that the rest of the nodes contributed to the transmission of its traffic. Thus, at the end of every phase and before the cost update, each node can inform which nodes it has helped by sending a "receipt" for the corresponding amount of energy. Note that since few bytes are enough to encode this information, this is a relatively small overhead compared to the overall traffic. Following the spirit of this paper, we do not consider that nodes may lie and report incorrect values. This is a potential problem because the proposed mechanism is not incentive compatible, but this issue is problematic for TCP/IP protocols as well and most users do not seem to be exploiting it to gain advantage.

The frequency of cost updates—which is proportional to the communication overhead—depends on the traffic pattern and on the duration of the time horizon. Since when considering fairness measures, we can expect time horizons to be of the order of tenths of minutes to a few hours (e.g., in the laptop scenario described in the paper, no one would get its battery drained by an unfair network in few minutes), this would correspond to an update frequency of a few minutes. Related to the time-scale of transmitting packets, this is a very infrequent event representing a negligible overhead. In any case, the actual frequency must be set experimentally according to the specific application and engineering characteristics of the system.

Finally, the integration of Distr Online Routing to a real protocol depends on how the protocol performs the routing. Most wireless protocols discover routes using distributed algorithms that compute exact or approximate shortest path in the network (see, e.g., Anderegg and Eidenbenz (2003) and references therein). A typical protocol works as follows: when a node s wants to communicate with a node t, s broadcasts a *route request* packet to all its neighbors. Each node i that receives a route request estimates the energy that was needed to send the packet in the last hop, appends this information to the packet, and forwards the packet to all its neighbors. If a node i receives a route-request packet from multiple sources (i.e., there are alternative routes from s to i in the network), it may drop one of them if it had previously received a route-request packet from the same origin. When one or more route request packets reach the destination node t, it will have discovered the shortest path from s. To adapt this framework to our needs, nodes modify their (energy) costs according to their current fairness coefficient, as specified in Figure 4.

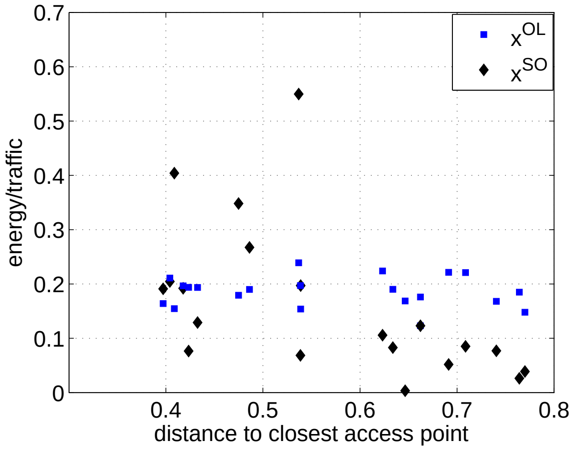

FIGURE 11. Energy consumption divided by traffic received, as a function of the distance to the closest access point.

8. CONCLUSIONS

All the evaluations of DISTR ONLINE ROUTING we have performed were solved almost instantaneously, regardless of the value of n, so we do not expect problems to arise when solving larger instances. On the other hand, computing some of the benchmarks needed to evaluate the solutions provided by the online algorithm was more time-consuming. In practice, though, these benchmarks are not required in an actual implementation of the proposed routing scheme. Recall that Example 1 showed a network in which the central node was in an unfair situation because of its location. We have presented an algorithm that provides solutions that are more fair to all nodes. In this way, nodes have a smaller incentive to act selfishly and to stop cooperating with the network.

Coming back to one of the school-mesh instances described in the computational study, Figure 11 provides a clear proof of the benefits of our approach. For each node, the figure plots the distance to the closest access point versus the energy it spends divided by the traffic it receives (this ratio measures the energy consumption per bit of information received). Diamonds indicate the energy spent by a system optimal routing, which achieves a fairness of $\Phi(x^{\mathrm{SO}}) = 0.271$. Squares show the energy spent by the same nodes in the solution found by DISTR ONLINE ROUTING, which achieves a fairness of $\Phi(x^{\mathrm{OL}}) = 0.735$. Minimizing energy and disregarding fairness makes some laptops close to the access points spend much more energy than necessary. Instead, the variability of the normalized energy consumption of the solution returned by the algorithm we have proposed is reduced significantly. This picture captures the essence of our approach: by explicitly considering fairness when routing traffic, one can equalize the energy expenditures without increasing their average too much.

In closing, we want to highlight that the ideas about efficiency and fairness in wireless routing that we have introduced are applicable with a greater generality than just the OLPC project. We have only chosen to focus in this project because it is a concrete application. Nevertheless, Section 6.1.3 has provided some evidence that our algorithm performs better when networks are larger. A line for future research involves the computation of faster benchmarks that can be used to show the performance of our approach when networks are significantly larger. This can be applied to the study of some of the applications mentioned in the introduction such as ad hoc networks of smart phones.

Acknowledgements

The first and second authors are partially supported by MIUR, Italy, and by the EU project *ARRIVAL* (contract n. FP6-021235-2). Part of this work was conducted while the second author was visiting Columbia University supported by a Marco Polo grant from the University of Bologna. Thanks are due to Alessandra Giovanardi for introducing us to the world of mesh networks and to Alberto Caprara, Claudia D'Ambrosio, Antonio Frangioni, George Karakostas and Leo Liberti for stimulating conversations regarding the various subproblems and their complexity. We are indebted to two anonymous referees for constructive comments and remarks that enabled us to improve our paper significantly.

References

Anderegg, L. and S. Eidenbenz (2003). Ad hoc-VCG: a truthful and cost-efficient routing protocol for mobile ad hoc networks with selfish agents. In *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking (MobiCom)*, pp. 245–259. ACM Press.

Beccaria, G. and A. Bolelli (1992). Modelling and assessment of dynamic route guidance: The Margot project. In L. Olaussen and E. Helli (Eds.), *Proceedings of the 3rd IEEE Vehicle Navigation & Information Systems Conference*, pp. 117–126. Oslo, Norway.

Ben-Akiva, M. E. and S. Lerman (1985). *Discrete Choice Analysis: Theory and Application to Travel Demand*, Volume 9 of *Transportation Studies Series*. MIT Press, Cambridge, MA.

Bertsekas, D. and R. Gallager (1992). *Data Networks*. Prentice Hall, Englewood Cliffs, NJ.

Bhargava, R., A. Goel, and A. Meyerson (2001). Using approximate majorization to characterize protocol fairness. In *Proceedings of the 2001 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, pp. 330–331. ACM Press.

Borodin, A. and R. El-Yaniv (1998). *Online computation and competitive analysis*. Cambridge University Press, Cambridge, UK.

Buchegger, S. and J.-Y. Le Boudec (2002a). Nodes bearing grudges: towards routing security, fairness, and robustness in mobile ad hoc networks. In *Proceedings of the 10th Euromicro Workshop on Parallel, Distributed and Network-based Processing*, pp. 403–410.

Buchegger, S. and J.-Y. Le Boudec (2002b). Performance analysis of the CONFIDANT protocol. In *Proceedings of the 3rd ACM International Symposium on Mobile Ad Hoc Networking & Computing (MobiHoc)*, Lausanne, Switzerland, pp. 226–236. ACM Press, New York, NY.

Buttyán, L. and J.-P. Hubaux (2003). Stimulating cooperation in self-organizing mobile ad hoc networks. *Mobile Networks and Applications 8*(5), 579–592.

Canalys (2008). Canalys research release 2008/112. `http://www.canalys.com/pr/2008/r2008112.htm`.

Chun, B., R. Fonseca, I. Stoica, and J. Kubiatowicz (2004). Characterizing selfishly constructed overlay routing networks. In *Proceedings of 23rd Annual IEEE INFOCOM*, Vol. 2, pp. 1329–1339.

Correa, J. R., A. S. Schulz, and N. E. Stier-Moses (2007). Fast, fair, and efficient flows in networks. *Operations Research 55*(2), 215–225.

Eidenbenz, S., V. S. Anil Kumar, and S. Zust (2006). Equilibria in topology control games for ad hoc networks. *Mobile Networks Applications 11*(2), 143–159.

Fudenberg, D. and J. Tirole (1991). *Game Theory*. MIT Press, Cambridge, MA.

He, Q., D. Wu, and P. Khosla (2004). SORI: a secure and objective reputation-based incentive scheme for ad hoc networks. In *Wireless Communications and Networking Conference*, Vol. 2, pp. 825–830.

Jaffe, J. (1981). Bottleneck flow control. *IEEE Transactions on Communications 29*(7), 954–962.

Jahn, O., R. H. Möhring, A. S. Schulz, and N. E. Stier-Moses (2005). System-optimal routing of traffic flows with user constraints in networks with congestion. *Operations Research 53*(4), 600–616.

Ji, H. and C.-Y. Huang (1998). Non-cooperative uplink power control in cellular radio systems. *Wireless Networks 4*(3), 233–240.

Karakostas, G. and E. Markou (2008). Emergency connectivity in ad-hoc networks with selfish nodes. In E. S. Laber et al. (Eds.), *LATIN 2008: Theoretical Informatics*, Búzios, Brazil, Volume 4957 of *Lecture Notes in Computer Science*, pp. 350–361. Springer, Heidelberg.

Kesselman, A., D. Kowalski, and M. Segal (2005). Energy efficient communication in ad hoc networks from user's and designer's perspective. *Mobile Computing and Communications Review 9*(1), 15–26.

Lee, J. W., R. R. Mazumdar, and N. B. Shroff (2005). Downlink power allocation for multi-class wireless system. *IEEE/ACM Transactions on Networking 13*(4), 854–867.

Levin, D. (2006). Punishment in selfish wireless networks: a game theoretic analysis. In *Proceedings of the First Workshop on the Economics of Networked Systems (NetEcon)*, Ann Arbor, MI, pp. 9–14.

Mahajan, R., M. Rodrig, D. Wetherall, and J. Zahorjan (2005). Sustaining cooperation in multihop wireless networks. In *Proceedings of the 2nd Symposium on Networked Systems Design & Implementation*, Volume 2, pp. 231–244. USENIX Association, Berkeley, CA.

Marti, S., T. J. Giuli, K. Lai, and M. Baker (2000). Mitigating routing misbehavior in mobile ad hoc networks. In *Proceedings of the 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, pp. 255–265. ACM Press, New York, NY.

Milan, F., J. J. Jaramillo, and R. Srikant (2006). Achieving cooperation in multihop wireless networks of selfish nodes. In *Proceedings of the Workshop on Game Theory for Communications and Networks (GameNets)*, Pisa, Italy. ACM Press, New York, NY.

Nisan, N., T. Roughgarden, É. Tardos, and V. V. Vazirani (2007). *Algorithmic Game Theory*. Cambridge University Press, UK.

OLPC (2008a). Battery and power. `http://wiki.laptop.org/go/Battery_and_power`.

OLPC (2008b). Networking scenarios. `http://wiki.laptop.org/go/Networking_scenarios`.

Rappaport, T. (1996). *Wireless Communications: Principles and Practices*. Prentice Hall, Englewood Cliffs, NJ.

Roughgarden, T. (2002). How unfair is optimal routing? In *Proceedings of the 13th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, San Francisco, CA, pp. 203–204. SIAM, Philadelphia, PA.