The Complexity of Connectivity in Wireless Networks

Thomas Moscibroda
Computer Engineering
and Networks Laboratory
ETH Zurich, Switzerland
moscitho@tik.ee.ethz.ch

Roger Wattenhofer
Computer Engineering
and Networks Laboratory
ETH Zurich, Switzerland
wattenhofer@tik.ee.ethz.ch

Abstract— We define and study the scheduling complexity in wireless networks, which expresses the theoretically achievable efficiency of MAC layer protocols. Given a set of communication requests in arbitrary networks, the scheduling complexity describes the amount of time required to successfully schedule all requests. The most basic and important network structure in wireless networks being connectivity, we study the *scheduling complexity of connectivity*, i.e., the minimal amount of time required until a connected structure can be scheduled. In this paper, we prove that the scheduling complexity of connectivity grows only polylogarithmically in the number of nodes. Specifically, we present a novel scheduling algorithm that successfully schedules a strongly connected set of links in time $O(\log^4 n)$ even in arbitrary worst-case networks.

On the other hand, we prove that standard MAC layer or scheduling protocols can perform much worse. Particularly, any protocol that either employs *uniform* or *linear* (a node's transmit power is proportional to the minimum power required to reach its intended receiver) *power assignment* has a $\Omega(n)$ scheduling complexity in the worst case, even for simple communication requests. In contrast, our polylogarithmic scheduling algorithm allows many concurrent transmission by using an explicitly formulated *non-linear power assignment* scheme.

Our results show that even in large-scale worst-case networks, there is no theoretical scalability problem when it comes to *scheduling* transmission requests, thus giving an interesting complement to the more pessimistic bounds for the *capacity* in wireless networks. All results are based on the *physical model* of communication, which takes into account that the signal-to-noise plus interference ratio (SINR) at a receiver must be above a certain threshold if the transmission is to be received correctly.

I. INTRODUCTION

A most important lesson in wireless multi-hop networking is that concurrent transmissions may cause interference. In a network, if too many devices transmit simultaneously, the interference caused by these transmissions will prevent an intended receiver from receiving the signal, i.e., the message is lost. On the other hand, if too few nodes transmit at the same time, valuable bandwidth is wasted and the overall throughput may suffer. Hence, the classic problem faced by any MAC layer or scheduling protocol is that neither selecting too many nor too few devices for concurrent transmission is acceptable. Instead, it is necessary to find the subtle balance, in which a large number of devices transmit in parallel and yet, the interference does not cause messages to be lost.

The measure that captures and quantifies this equilibrium is *time*. More specifically, assume that we are given a set of directed links between pairs of nodes that indicate communication requests. How much time is required to schedule all these requests? In this paper, we define and study this *scheduling complexity* in wireless networks. Like, for instance, the notion of *network capacity* [12], the scheduling complexity expresses a fundamental law that governs communication in wireless multi-hop networks. While roughly speaking, the capacity of a wireless network describes the maximum amount of information that can be transmitted in the network, the scheduling complexity indicates the minimum amount of time required to transmit over a set of communication links. As it turns out, studying the scheduling complexity of wireless networks reveals previously unknown aspects of communication in wireless networks.

In order to obtain results that capture the inherent characteristics of wireless networks, much care has to be taken when choosing the communication model. Multi-hop wireless networks have often been modeled as graphs. The nodes of this communication graph typically represent the physical devices, two nodes being connected by an edge if and only if the respective devices are within mutual transmission range. In this graph-theoretic model a node is assumed to receive a message correctly if and only if no other node in close physical proximity transmits at the same time. It is therefore not surprising that in graph theory, interference-free concurrent transmissions typically boil down to solving variants of coloring or independent set problems (e.g. [25]).

Clearly, this graph-theoretic notion of interference is a tremendous simplification of the physical reality faced in wireless networks [10], [2]. Particularly, the interference caused by different transmitters may accumulate and is not binary, i.e., does not stop at a any specific border. Moreover, when transmission powers are properly assigned, a node may successfully receive a message in spite of being in the transmission range of other simultaneous transmitters. In fact, a message is successfully received by a node if the *ratio* between the received signal strength on the one hand and the ambient noise plus interference from other nodes on the other hand exceeds a certain hardware-specific threshold. The communication model adopting this notion of signal-to-noise-

plus-interference ratio (SINR) is also known as the *physical model* [12].

In this paper, we adopt this physical model of communication and study the scheduling complexity, i.e. the question of how much time is required in order to successfully transmit messages over a given set of communication links. Formally, we are given an arbitrary (i.e., not randomly distributed) network, and a set of directed links representing communication requests. For each such link we assign a time slot and a power level such that all simultaneous transmissions are successful, i.e., not violating the signal-to-interference plus noise ratio at any receiver. In particular, we want an assignment that uses as few time slots as possible, which is precisely the task faced by MAC layer or scheduling protocols.

In practical scenarios, it is often not known in advance where and when communication requests arise. Certain fundamental types of network structures, however, play a vital role in many applications of wireless networks. Most basically, it is typically required that a message can be routed between any pair of nodes in the network. That is, all nodes should be connected by a *communication backbone*, such as a tree or any other suitable network topology.

In the main part of this paper we therefore study the scheduling complexity of *connecting* a given number of nodes located at arbitrary positions by some communication tree. For this basic *connectivity problem*, we present an algorithm which constructs a spanning tree, and assigns power levels and time slots to each link of the tree such that in polylogarithmic time, all transmissions are received correctly, i.e., without violating the signal-to-interference plus noise ratio at any receiver. In other words, we prove that for the most basic of all network properties, connectivity, the scheduling complexity is *polylogarithmic* in the number of network nodes. This result is astonishing because it shows that even in the harsh SINR model and in large-scale networks, scheduling communication requests (even if they must result in a connected structure!) can be achieved efficiently. Hence, theoretically, there is no fundamental scalability problem in wireless networks when it comes to *scheduling* communication requests.

Our upper bound is intriguing because standard MAC layer protocols and power assignment schemes can perform much worse. In particular, two different power assignment schemes have typically been studied and adopted in wireless networks. In *uniform power assignment schemes*, all nodes transmit with the same transmission power [11]. In *linear power assignment schemes*, it is assumed that if a node x wants to transmit a message to a node that is located in distance d, then x should send with a transmission power of $P_x = \rho \cdot d^\alpha$, for some constant ρ, where α is the so-called path-loss exponent. In other words, the strategy of most MAC layer or power control protocols (e.g. [26], [20], [22], [32], [30], [1], [29]) is to either let all nodes send at the same power, or to adjust the transmission power level of a packet to be proportional to P_{min}, where P_{min} is the minimum power required such that in the absence of interference from other nodes, the SINR at the intended receiver is just enough for decoding the packet.

In this paper, we show that the result of both of these power assignment schemes can be disastrous. In a network consisting of n nodes, we prove that any scheduling approach that uses a *uniform or linear power assignment* potentially requires up to time $\Theta(n)$ in order to schedule links that constitute a connected structure. This is exponentially slower than the scheduling complexity achieved by our polylogarithmic algorithm. In fact, we even show that *any protocol* that uses *uniform* or *linear power assignment* may need up to time $\Theta(n)$ until every node has been able to transmit successfully once. This holds even if the receiving node for every sender is selected best-possible, e.g. when each node transmits to its closest neighbor. Note that this result places a strong lower bound on the amount of time required by *any* MAC layer protocol (even an imaginary MAC protocol that optimally schedules the communication requests) if it assigns transmission powers uniformly or linearly.

Besides being of theoretical interest, the bad scheduling complexity achieved by uniform and linear power assignments has practical relevance: It shows that in order to obtain a fast scheduling of sending requests in wireless networks, MAC layer and scheduling protocols should adopt neither uniform nor linear power assignment. Instead, the remedy against this loss of efficiency is a highly non-linear assignment of power levels. That is, our polylogarithmic scheduling algorithm uses an explicitly defined power scheme that lies "in between" uniform and linear power assignment. Particularly, *many different power levels* are required in order to achieve an efficient schedule.

The rest of this paper is organized as follows. In Section II, we formally introduce and define the communication model and the *scheduling complexity*, respectively. The limitations of uniform and linear power assignments are studied in subsequent Section III. We then propose an efficient polylogarithmic scheduling algorithm in Section IV. The performance of a simple and natural, but worst-case inefficient linear power assignment algorithm is analyzed in Section V. An overview of related previous work is given in Section VI and finally, Section VII concludes the paper.

II. SCHEDULING COMPLEXITY

We consider the network nodes $X = \{x_1, \ldots, x_n\}$ to be located arbitrarily (even worst-case) in the Euclidean plane. The Euclidean distance between two nodes $x_i, x_j \in X$, is denoted by $d(x_i, x_j)$. For a (directed) link $f_{ij} = (x_i, x_j)$, $\ell(f_{ij}) = d(x_i, x_j)$ denotes the distance between its endpoints. Finally, the *ball* $B(x_i, r)$ of radius r around node x_i contains all nodes $x_j \in X$ for which $d(x_i, x_j) \leq r$. For simplicity and without loss of generality, we assume that the minimal distance between any two nodes is 1 and we define $\Delta = \log(\ell_{max})$, where ℓ_{max} is the largest distance between two nodes.

With what power level should nodes send in a certain timeslot? Intuitively, if the power level is large, more nodes can be covered, at the same time causing an increase in interference faced by other nodes. In the following, we formally define the notion of a power assignment.

As in [12], we assume without loss of generality that transmissions are slotted into synchronized slots of equal length. In each time-slot t, a node x can either transmit or not transmit a message. A *power assignment* determines the power level chosen by each node in a certain time-slot. Formally, a power assignment ϕ_t is a function $\phi_t : X \mapsto \mathbb{R}^+$ which maps every node in the network to a power level. We denote by $\phi_t(x_i)$ the power level of node x_i in time-slot t. If a node is not scheduled to transmit in this time-slot, then $\phi_t(x_i) = 0$. In case it is clear from the context which time-slot t is meant, we also use the notational short-cut $P_i = \phi_t(x_i)$. A *schedule* $\mathcal{S} = (\phi_1, \ldots, \phi_{T(\mathcal{S})})$ is a sequence of $T(\mathcal{S})$ power assignments, where ϕ_i denotes the power assignment in time-slot i. Finally, we call $T(\mathcal{S})$ the *length* of schedule $\mathcal{S}$. That is, a schedule S of length $T(S)$ determines the power level P_i for every node $x_i \in X$ for $T(S)$ consecutive time-slots.

A major aspect of the model is the description of the circumstances under which a message is received by its intended recipient. As mentioned in the introduction, in this paper we adopt the *signal-to-interference plus noise ratio* (SINR) model (*physical model* [12]) in order to determine whether a transmission is successfully received. In this model, the successful reception of a transmission depends on the received signal strength, the ambient noise level, and the interference caused by simultaneously transmitting nodes. Let P_r be the signal power received by a node x_r and let I_r denote the amount of interference generated by other nodes. Finally, let N be the ambient noise power level. Then, a node x_r receives a transmission if and only if $\frac{P_r}{N + I_r} \geq \beta$, where β denotes the minimum signal to interference ratio that is required for a message to be successfully received.[1]

In wireless networks, the value of received signal power P_r is a decreasing function of the distance $d(x_s, x_r)$ between transmitter node x_s and receiver node x_r. More specifically, the received signal power can be modeled as decaying with distance $d(x_s, x_r)$ as $\frac{1}{d(x_s,x_r)^\alpha}$. The so-called path-loss exponent α is a constant between 2 and 6 and depends on external conditions of the medium, as well as the exact sender-receiver distance. As customary, we assume that $\alpha > 2$ [12].

Let P_i be the power level assigned to node x_i in a time-slot. A message transmitted from a node $x_s \in X$ is successfully received by a node x_r if

$$\frac{\frac{P_s}{d(x_s,x_r)^\alpha}}{N + \sum_{x_i \in X \setminus \{x_s\}} \frac{P_i}{d(x_i,x_r)^\alpha}} \geq \beta. \tag{1}$$

As for notation, $I_r(x_v) = \frac{P_v}{d(x_v,x_r)^\alpha}$ denotes the amount of interference encountered at x_r caused by a node x_v that is concurrently transmitting. Finally, the total interference I_r experienced by a receiver x_r is the sum of the interferences created by all nodes (except the intended sender x_s) in the network, i.e., the *total interference* at x_r is $I_r = \sum_{x_i \in X \setminus \{x_s\}} I_r(x_i)$.

Definition 2.1: Consider a time-slot t and a power assignment ϕ_t. We say that a directed link (x_i, x_j) is *successfully*

scheduled at time-slot t if x_j successfully receives a message from x_i according to the SINR Inequality (1).

Let E_t be the set of all successful links in time-slot t. We want that after as few time-slots as possible, the union of all link-sets E_t forms the desired network topology, e.g., connectivity. We therefore define the scheduling problem for a given network property as follows.

Definition 2.2: The scheduling problem for a network property Ψ is to find a schedule $\mathcal{S}$ of minimal length $T(\mathcal{S})$, such that the union of all successfully transmitted links $\Sigma = \bigcup_{t=1}^{T(\mathcal{S})} E_t$ satisfies property Ψ.

As an example, the scheduling problem when Ψ is *connectivity* translates to finding a schedule $\mathcal{S}$ of minimal length in which all successfully transmitted links strongly connect the network, i.e., there exists a path between all pairs of nodes. Finally, we define the scheduling complexity of a network property Ψ.

Definition 2.3: The *scheduling complexity* of a network property Ψ is the minimal number of time-slots T, such that there always exists a valid schedule $\mathcal{S}$ for Ψ of length $T = T(\mathcal{S})$.

With regard to a scheduling algorithm $\mathcal{A}$, we also refer to the *scheduling complexity of* $\mathcal{A}$ as the number of time-slots required by this algorithm in the worst-case to schedule the desired network property.

The scheduling complexity in wireless networks is a fundamental measure that indicates how quickly communication requests can be established. In this regard, the *scheduling-complexity* complements the notion of *capacity* in wireless networks that has been studied in the past. While *capacity* captures the amount of information that can maximally be sent in a *best-case* scenario (i.e., without assuming worst-case restrictions on the network topology over which messages must be sent), the *scheduling-complexity* of a wireless network describes how fast information can be transmitted in a *worst-case* scenario, i.e., if communication links must satisfy a certain property such as connectivity. Hence, the scheduling complexity describes the theoretically achievable efficiency of any MAC layer or scheduling protocol.

Note that we could define the scheduling complexity of wireless networks in an alternative, but equivalent way. Particularly, the scheduling complexity also captures the minimal *number of non-interfering frequencies* that are required in order to schedule a desired network property in a *single time-slot*. Throughout the paper, we will focus on the *time* aspect of the scheduling complexity, but all our upper and lower bound results immediately apply to frequencies as well.

III. LIMITATIONS OF UNIFORM AND LINEAR POWER ASSIGNMENT

In this section, we prove the deficiency of the power assignment schemes that have been widely studied in the field of wireless networks (and that have also been adopted by most standard MAC layer protocols). We study the scheduling

---

[1]Note that all our results can directly be generalized to the case in which nodes can use multiple non-interfering frequencies simultaneously.
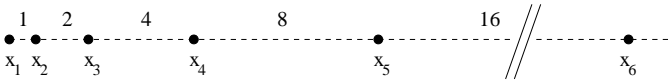
Fig. 1. Example with nodes x_i being located at position $x_i = 2^i$, $i = 1, \ldots, n$.

complexity of the following property Ψ_{min}: *Every node $x \in X$ can send at least one message successfully.* Note that property Ψ_{min} does not restrict to *which* other node a node must send, i.e., nodes can for instance select their nearest neighbors as receivers. In other words, we investigate the following simple and fundamental problem: How much time is required until every node can successfully transmit one message, when the receivers for each sender are selected best possible.

Because of its simplicity, achieving good solutions for this problem (i.e., good scheduling complexity for property Ψ_{min}) seems to be easy and intuitively, one would expect that simple MAC layer protocols achieve a good performance. Surprisingly, however, the opposite is true. We prove in this section that generally accepted power assignment schemes are incapable of achieving a reasonable scheduling complexity even for the simple problem Ψ_{min}. In the worst case, such protocols have the same asymptotic performance as a protocol that schedules each single network node individually.

One way of assigning power levels in a radio network is to let every node transmit at the same power. Such *uniform power assignment* schemes have been widely studied [26], [11], [13] and adopted in practical systems. However, the following theorem states that even for property Ψ_{min}, the scheduling complexity of a uniform power assignment algorithm is linear in n.

Theorem 3.1: Assume that every node v_i has the same transmission power. The scheduling complexity for problem Ψ_{min} in such a uniform power assignment is at least $n \cdot \frac{\beta}{2^\alpha + \beta} \in \Omega(n)$, even in the absence of ambient noise.

Proof: Consider the example given in Figure 1, in which nodes $x_0, \ldots, x_{n-1}$ are placed on a straight line with exponentially increasing distances between them. We prove that in each time-slot, at most $\frac{2^\alpha}{\beta} + 1$ nodes can send successfully if the transmission power is uniform. Assume for contradiction that there are $L = \frac{2^\alpha}{\beta} + 2$ nodes sending successfully in the same time-slot, and let x_s be the right-most of these transmitters. Further, assume that x_s's transmission is successfully received by node x_r. On an exponential line, if x_r is to the left of x_s, it holds that $d(x_i, x_r) \leq d(x_s, x_r)$ for each simultaneously transmitting node x_i. If x_r is on x_s's right, it holds that $d(x_i, x_r) \leq 2d(x_s, x_r)$ for each such x_i. Because all transmission powers P are equal and x_s is the right-most sender, the SINR at x_r is therefore at most

$$\frac{\frac{P}{d(x_s,x_r)^\alpha}}{N + (L-1) \cdot \frac{P}{(2d(x_s,x_r))^\alpha}} \leq \frac{2^\alpha}{L-1} = \frac{2^\alpha \beta}{2^\alpha + \beta} < \beta,$$

which is not sufficiently high for a correct reception of the message at x_r, which yields the contradiction. Because at most $\frac{2^\alpha}{\beta} + 1$ links can be simultaneously scheduled in any time-slot, the algorithm requires at least $n \cdot (\frac{2^\alpha + \beta}{\beta})^{-1}$ time-slots to schedule all nodes at least once, from which the theorem follows. □

The other intuitive and frequently adopted way of assigning power levels when scheduling a set of nodes is the following: Intended senders transmit at a power level that is *proportional* to the minimal power *required* for transmitting over the wireless link (e.g., [20], [29], [1]). In other words, for a pair of sender s_i and receiver r_i, s_i sends with power $P_s = \rho \cdot d(s_i, r_i)^\alpha$, where ρ is a constant which depends on the values of α, β, and the ambient noise N. Since $d(s_i, r_i)^\alpha$ is the minimal amount of power necessary to reach r_i from s_i, it seems natural to let nodes send with a power that is proportional to $d(s_i, r_i)^\alpha$. We call such a power assignment *linear*, because the power assigned to a node depends linearly on the minimal power required for its link. *Linear power assignments* have been assumed in many papers written on topology control (e.g. [24]), in papers that study the issue of energy efficiency in wireless networks (e.g. [29], [1]), and in some MAC layer protocols [20]. However, like in the uniform case, protocols using linear power assignment can perform badly even for the basic requirement Ψ_{min}.

Theorem 3.2: Assume that every node x_i that intends to send a message over a link of length ℓ_i transmits with power $P_s = \rho \cdot \ell_i^\alpha$, for an arbitrary constant ρ which may depend on α, β, or N. The scheduling complexity for problem Ψ_{min} in such a linear power assignment is $n \cdot \min\{1, \beta/2^\alpha\} \in \Omega(n)$, even in the absence of ambient noise.

Proof: Consider again the example given in Figure 1. Let x_i be a transmitting node in an arbitrary time-slot t. In a linear power assignment, it transmits at least with power $P_i = \rho \cdot d(x_i, x_{i-1})^\alpha$, for some constant ρ. As a consequence of this transmission, all nodes $x_j, j < i$ face an interference of at least

$$I_j(x_i) \geq \frac{\rho \cdot d(x_i, x_{i-1})^\alpha}{(2d(x_i, x_{i-1}))^\alpha} = \frac{\rho}{2^\alpha}$$

because the distance $d(x_i, x_j)$ is at most $2d(x_i, x_{i-1})$ in the exponential line. Because at least the same amount of interference is caused by all simultaneous senders x_i, a node x_j faces a total interference of at least $I_j \geq R \cdot \frac{\rho}{2^\alpha}$, where R is the number of sending nodes to the right of x_j. Now, let x_s be the left-most node that sends a message in time-slot t, and let x_r be its receiver. Because the SINR at every x_r must be at least β, i.e.,

$$\frac{\frac{\rho \cdot d(x_s,x_r)^\alpha}{d(x_s,x_r)^\alpha}}{N + R \cdot \frac{\rho}{2^\alpha}} \geq \frac{\rho 2^\alpha}{2^\alpha N + \rho R} \geq \beta.$$

From this, it follows that the maximum number of simultaneous senders R_{max} can be at most $R_{max} \leq \frac{2^\alpha}{\beta}$ and consequently, the algorithm requires at least $n \cdot \min\{1, \beta/2^\alpha\}$

for scheduling all nodes. Note that this result holds even if there is no noise N. □

In reality, both α and β are small constant values. Hence, Theorems 3.1 and 3.2 show that even in the most basic scheduling problem Ψ_{min}, only a *small constant number* of links can be simultaneously scheduled when adopting uniform or linear power assignment schemes. For $\alpha = 4$ and $\beta = 7dB$, for instance, at most 4 links can be scheduled in parallel. It follows that any MAC layer or scheduling protocol that assigns transmission powers according to either of these two policies performs disastrously in the worst-case. In the subsequent section, we show how a *non-linear power assignment* yields much more efficient schedules. Specifically, we prove that in every network (including the one shown in Figure 1), no fewer than $\Omega(n/\log^4 n)$ links can simultaneously be scheduled in each time-slot.

IV. THE COMPLEXITY OF CONNECTIVITY

In this section, we present an algorithm which, for every possible placement of n nodes in the plane, successfully schedules a *strongly connected subgraph* in $O(\log^4 n)$ time-slots. This proves that in wireless networks, the scheduling complexity of strong-connectivity is at most polylogarithmic in the number of nodes.

Theorem 4.1: The scheduling complexity of strong-connectivity in wireless networks is at most $O(\log^4 n)$.

This theorem captures a fundamental characteristic of wireless networks: scheduling a strongly connected topology theoretically remains efficient in every wireless network, even when n becomes large. As shown in Section III, this is in contrast to the $\Omega(n)$ scheduling complexities achieved by uniform and linear power assignment protocols.

Algorithm 1 proceeds in *phases*, each phase corresponding to an iteration of the outermost loop. The purpose of this outer loop is to gradually reduce the number of *active nodes* $x_i \in \mathcal{A}$. Initially, the set of active nodes $\mathcal{A}$ contains all nodes, and whenever a node becomes passive (by being discarded from $\mathcal{A}$), it does not transmit in any subsequent time-slot. At the outset of a phase p (line 5), every active node x_i chooses its closest active neighbor, say x_j, and the directed link $f_{ij} = (x_i, x_j)$ becomes designated to be scheduled in phase p. After breaking cycles of length 2 (i.e. two nodes that are mutually closest neighbors) in Line 7, $\mathcal{F}_p$ is the set of all selected links that are scheduled in phase p. $\mathcal{F}_p$ forms a *nearest neighborhood forest* consisting of a set of trees, from each of which only the root remains active in the next phase $p+1$. This process is repeated until there remains only a single active node. At this point, the scheduled links form a *directed tree* towards a single node, which can then complete the strong connectivity requirement in a single additional time-slot.

The main challenge is how to efficiently schedule the forest $\mathcal{F}_p$. As we have seen in Section III, neither standard linear nor uniform power assignments lead to acceptable solutions. So, the problem is: How do we choose the sending nodes and at what power levels should they send? As we will see, both of these subproblems are strongly interrelated.

Algorithm 1 Polylogarithmic Scheduling Algorithm

Input: An arbitrarily located set of nodes X
Output: A schedule $\mathcal{S}$ satisfying strong-connectivity

1: $\mathcal{A} := X;\quad t := 1;\ \nu > 4N;\ \mu := 3 + 2^{\frac{7}{\alpha}+2} \sqrt[\alpha]{\beta \frac{\alpha-1}{\alpha-2}};$
2: **while** $|\mathcal{A}| > 1$ **do** $\qquad\qquad \{\ast$ *Phase p* $\ast\}$
3: $\quad \mathcal{F}_p := \emptyset;$
4: $\quad$ **for each** $x_i \in \mathcal{A}$ **do**
5: $\qquad$ choose $x_j \in \mathcal{A} \setminus \{x_i\}$ minimizing $d(x_i, x_j);$
6: $\qquad f_{ij} := (x_i, x_j);$
7: $\qquad$ **if** $f_{ji} \notin \mathcal{F}_p$ **then** $\mathcal{F}_p := \mathcal{F}_p \cup f_{ij};$ **fi**
8: $\quad$ **end for**
9: $\quad$ Let $\mathcal{L} = L_0, \ldots, L_{\Delta-1}$, such that L_k is the set of links f_{ij} of length $2^k \le \ell(f_{ij}) < 2^{k+1};$
10: $\quad$ Delete all empty length classes L_k and rename $\mathcal{L}$ such that L_k is the kth largest non-empty length-class;
11: $\quad$ **for each** $x_i \in \mathcal{A}$ with $f_{ij} \in \mathcal{F}_p$ **do** $\mathcal{A} := \mathcal{A} \setminus \{x_i\};$
12: $\quad$ **for** $j = 0$ **to** $\log(4\beta n) - 1$ **do**
13: $\qquad$ Schedule all $f_{ij} \in \mathcal{F}_p \cap \left(\bigcup_{h=0}^{\frac{n}{\log(4\beta n)}-1} L_{h\log(4\beta n)+j} \right)$ using subroutine **Schedule()**;
14: $\quad$ **end for**
15: **end while**
16: $\phi_t(x_i) := N\beta \cdot \ell_{max}^\alpha$ for $x_i \in \mathcal{A}$
17: $\mathcal{S} := \{\phi_1, \ldots, \phi_{t-1}\};$

Subroutine Schedule() :

1: Let $\mathcal{F}$ be the set of links to be scheduled, classified in at most $p = \lceil \frac{n}{\log(4\beta n)} - 1 \rceil$ length classes $\mathcal{L}' = L_1', \ldots, L_p';$
2: **for each** $f_{uv} \in \mathcal{F} \cap L_k'$ **do** $\tau(x_u) := p - k + 1;$
3: **while** $\mathcal{F} \ne \emptyset$ **do**
4: $\quad$ **for each** $x_i \in \mathcal{A}$ **do** $\phi_t(x_j) := 0;$ **end for**
5: $\quad \mathcal{F}_t := \mathcal{F};\quad E_t := \emptyset;$
6: $\quad$ **while** $\mathcal{F}_t \ne \emptyset$ **do**
7: $\qquad$ choose the link $f_{ij}^* \in \mathcal{F}_t$ of minimal length;
8: $\qquad E_t := E_t \cup \{f_{ij}^*\};\quad \mathcal{F}_t := \mathcal{F}_t \setminus \{f_{ij}^*\};$
9: $\qquad \phi_t(x_i) := \nu(4\beta n)^{\tau(x_i)} \ell(f_{ij}^*)^\alpha;\quad \{\ast Schedule\ f_{ij}^* \ast\}$
10: $\qquad$ **for each** $f_{k\ell} \in \mathcal{F}_t$ **do**
11: $\qquad\quad \delta_{ik} := \tau(x_i) - \tau(x_k);$
12: $\qquad\quad$ **if** $\delta_{ik} = 0$ **and** $x_\ell \in B\left(x_i, \mu\ell(f_{ij}^*)\right)$ **then**
13: $\qquad\qquad \mathcal{F}_t := \mathcal{F}_t \setminus \{f_{k\ell}\};$
14: $\qquad\quad$ **else if** $x_\ell \in B\left(x_i, (4n\beta)^{\frac{\delta_{ik}+1}{\alpha}} \cdot \ell(f_{ij}^*)\right)$ **then**
15: $\qquad\qquad \mathcal{F}_t := \mathcal{F}_t \setminus \{f_{k\ell}\};$
16: $\qquad\quad$ **end if**
17: $\qquad$ **end for**
18: $\quad$ **end while**
19: $\quad \mathcal{F} := \mathcal{F} \setminus E_t;\quad t := t + 1;$
20: **end while**

The links in $\mathcal{F}_p$ are classified into at most n different non-empty length classes L_k, such that the length of links in the same class differ by at most a factor of 2 (Lines 9 and 10). The algorithm schedules these links using $\log(4\beta n)$
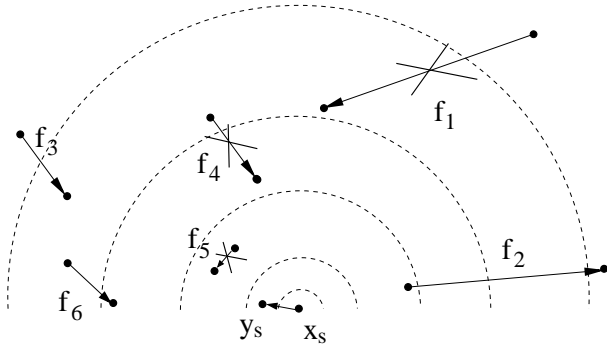
Fig. 2. Illustration of Algorithm 1. In the example, links f_1, f_4, and f_5 are dropped from $\mathcal{F}_t$. Note that f_2 is not dropped because its receiver is outside the critical ball, even though its sender is close from x_s.

calls to a *schedule()* subroutine, which lies at the heart of our algorithm. In the first call to the subroutine, it schedules all links in length classes $L_0, L_{\log(4\beta n)}, L_{2\log(4\beta n)}, \ldots$, in the second call, $L_1, L_{\log(4\beta n)+1}, L_{2\log(4\beta n)+1}, \ldots$, and so forth. In other words, the length classes scheduled in the same subroutine call are always separated by $\log(4\beta n) - 1$ length classes. The idea is that the length of simultaneously scheduled links should be either similar (same length class) or very different, but not in between.

At the outset of the subroutine, the $p = \lceil \frac{n}{\log(4\beta n)} - 1 \rceil$ received length classes are renamed to $L'_1, L'_2, \ldots, L'_p$ and the set $\mathcal{F}$ denotes the links that must be scheduled. Scheduling this forest $\mathcal{F}$ is done as follows. Each outer-loop iteration of the subroutine corresponds to a time-slot t in which a subset of the links of $\mathcal{F}$ (denoted by E_t) is chosen to be scheduled simultaneously. The selection of E_t from $\mathcal{F}$ proceeds as follows. Starting from the shortest link, the algorithm greedily picks links f^*_{ij} (Line 7) from $\mathcal{F}_t$, a set denoting the links from $\mathcal{F}$ that are still eligible to be selected in time slot t. Initially, all links are eligible, $\mathcal{F}_t = \mathcal{F}$, but whenever a link is selected, as shown in Figure 2, links whose *distance-to-receiver vs. length* ratio is too small are removed from $\mathcal{F}_t$ (Lines 12-15). Such links become eligible again in the next time-slot.

As we show in Lemma 4.8, this distance-length threshold avoids that too many links are discarded from $\mathcal{F}_t$ in any time-slot. The specific distance-length ratio of a link f_{ij} depends on the value $\tau(x_i)$, which expresses that length class containing f_{ij} is the $\tau(x_i)^{\text{th}}$ shortest length-class of the phase. For two intended senders x_i and x_u, the value δ_{iu} denotes the difference between $\tau(x_i)$ and $\tau(x_u)$, i.e., the number of length classes separating x_i and x_u. When scheduling the link f^* with sender x_s and y_s in a time-slot, larger links f_i are deleted from $\mathcal{F}_t$ if the intended receiver is located in the ball $B(x_i, (4n\beta)^{\frac{\delta_{si}+1}{\alpha}} \cdot \ell(f^*))$. The intuition is not to schedule a link in $\mathcal{F}$ if its *receiver* is in physical proximity of a much *shorter*, concurrently scheduled link. Note that the radius of the ball depends exponentially on the two links' *relative length classes*. Using a simple distance function instead would either ruin the schedule's complexity or correctness.

Of particular interest is the power assignment adopted in Line 9 of the subroutine. In comparison to a linear power assignment, the transmission power assigned to an intended sender x_i is scaled by a factor of $\nu(4\beta n)^{\tau(x_i)}$, thus disproportionally favoring short links over long ones! That is, nodes transmitting to close nodes "overpower" their receivers much more than transmitters of a long link. Note, however, that the *scaling factor* does not *directly* depend on the length of f^*_{ij} relative to other links.[2] Instead, the power level depends on the *length-class* to which f^*_{ij} is assigned. Because empty length classes were deleted from $\mathcal{L}$ before the subroutine, a node's power scaling factor does not depend directly on its *length*. Rather, it depends on the *relative position of its length class L'_k in $\mathcal{L}'$*. This somehow counter-intuitive power assignment in combination with the greedy procedure for selecting transmitters is the key to our algorithm. It keeps the SINR high for all intended receivers, while still allowing to schedule many links in parallel.

As shown in Section III, neither linear nor uniform power assignment schemes result in an efficient schedule. In contrast, our algorithm's power assignment lies *in between* these two extreme cases. Unlike in linear power assignment schemes, it favors short links over long ones in the sense that they transmit at a higher power than actually required to reach the receiver. But unlike in uniform power schemes, nodes having long links still use a higher absolute transmission power than short ones.

Analysis: We now show that Algorithm 1 is both correct and efficient. We start with a simple lemma that characterizes the length ratio between two links that are to be scheduled in the same execution of the scheduling subroutine.

Lemma 4.2: Let f_{xy} and f_{uv} be two links that are considered in the same subroutine call, and let $\tau(x) \geq \tau(u)$, where x and u are the intended transmitting nodes. Then, the length of f_{uv} is at least $\ell(f_{uv}) \geq \frac{1}{2}(4n\beta)^{\delta_{xu}} \cdot \ell(f_{xy})$.

Proof: By Line 13 of the algorithm, only links in length classes $L_j, L_{\log(4\beta n)+j}, L_{2\log(4\beta n)+j}, \ldots$ are considered in the same execution of the subroutine. The value of δ_{xu} denotes the number of non-empty length classes that separate links f_{xy} and f_{uv}, each incurring at least a doubling of the length. Taking into account that lengths can differ by at most a factor of 2 within a length-class, it follows that $\ell(f_{uv})$ is at least

$$\ell(f_{uv}) \geq \ell(f_{xy}) \cdot 2^{\delta_{xu}\log(4\beta n)-1} = \ell(f_{xy}) \cdot \frac{1}{2}(4n\beta)^{\delta_{xu}}.$$

$\square$

We now show that the schedule obtained by Algorithm 1 is *correct*. In particular, we prove in Theorem 4.6 that all transmissions E_t scheduled in a time-slot t during the algorithm are received successfully by the intended receivers. More specifically, we show that the SINR at every intended receiver is high enough, i.e., larger than β. In the following series of Lemmas 4.3, 4.4, and 4.5, we bound the total interference experienced at a receiver x_r that was caused by

---

[2]In fact, it can be shown that any power assignment that depends directly on the relative length of links cannot lead to an efficient schedule.

simultaneously scheduled links from smaller, the same, or larger length classes, respectively.

Lemma 4.3: Consider a scheduled link f_x with intended sender x_s and receiver x_r. Let I_r^- be the total interference caused at x_r by simultaneously transmitting nodes y_i for which $\tau(y_i) > \tau(x_s)$. It holds that

$$I_r^- \ < \ \frac{\nu}{4}\beta^{\tau(x_s)-1}(4n)^{\tau(x_s)}.$$

Proof: Bounding the interference caused by substantially shorter links is tricky, because our algorithm's non-linear power assignment scheme assigns such nodes a disproportionately large sending power. Consider a link f_y with transmitting node y_i. We begin by showing that the interference $I_r(y_i)$ at x_r caused by y_i is at most

$$I_r(y_i) \ \leq \ \nu(4\beta n)^{\tau(x_s)-1}. \tag{2}$$

Assume for contradiction that inequality (2) does not hold, i.e., assume that

$$I_r(y_i) = \frac{P_i}{d(y_i, x_r)^\alpha} = \frac{\nu(4\beta n)^{\tau(y_i)}\ell(f_y)^\alpha}{d(y_i, x_r)^\alpha} \overset{!}{>} \nu(4\beta n)^{\tau(x_s)-1}.$$

Simplifying the above inequality and defining χ to be the ratio $\chi = \frac{d(y_i, x_r)}{\ell(f_y)}$, it must hold that

$$(4\beta n)^{\tau(y_i)-\tau(x_s)+1} \ > \ \chi^\alpha$$

and consequently $(4\beta n)^{\frac{\delta_{is}+1}{\alpha}} > \chi$. This means that the distance between y_i and x_r is upper-bounded by

$$d(y_i, x_r) \ < \ \ell(f_y) \cdot (4\beta n)^{\frac{\delta_{is}+1}{\alpha}}.$$

However, this establishes a contradiction to the definition of Algorithm 1. In the iteration of the innermost while-loop in which f_y was scheduled, f_x would have been deleted from $\mathcal{F}_t$ if $d(y_i, x_r) < \ell(f_y) \cdot (4\beta n)^{\frac{\delta_{is}+1}{\alpha}}$ in Line 14. Hence, f_x and f_y would not have been scheduled in the same time-slot, which establishes the contradiction.

Because the number of transmitting nodes is at most n, the total interference caused by transmitters of substantially smaller links is

$$I_r^- \ = \sum_{y_i : \tau(y_i) > \tau(x_s)} I_r(y_i) \ \leq \ n \cdot \nu(4\beta n)^{\tau(x_s)-1}$$

from which the lemma follows. $\qquad\square$

The next lemma similarly bounds the interference from nodes that are in the same length-class.

Lemma 4.4: Consider a scheduled link f_x with intended sender x_s and receiver x_r. Let I_r^0 be the total interference caused at x_r by simultaneously transmitting nodes y_i for which $\tau(y_i) = \tau(x_s)$. It holds that

$$I_r^0 \ \leq \ \frac{\nu}{4}\beta^{\tau(x_s)-1}(4n)^{\tau(x_s)}.$$

Proof: By Lemma 4.2, we know that for each link f_i, $f_i \neq f_x$, with transmitting node y_i and $\tau(y_i) = \tau(x_s)$, it holds that $2\ell(f_x) \geq \ell(f_i) \geq \frac{1}{2}\ell(f_x)$. In the algorithm, after scheduling a link f_k, all links in the same length-class whose

intended receivers are within distance $\mu\ell(f_k)$ of the sender are deleted from $\mathcal{F}_t$ in line 13 of the scheduling subroutine. More precisely, around each transmitting node y_i, there can be no other scheduled sender y_j from the same length class within distance at least $\mu\ell(f_i) - \ell(f_j) \geq (\mu-2)\ell(f_i) \geq \frac{\mu-2}{2}\ell(f_j)$. This means that disks D_i of radius $\frac{\mu-2}{4}\ell(f_i)$ centered at all transmitting nodes y_i from the same length class do not overlap. The area of each such disk is $A(D_i) = (\frac{\mu-2}{4}\ell(f_i))^2\pi$.

Consider rings R_k of width $\frac{1}{2}(\mu-3)\ell(f_x)$ around x_r, consisting of all nodes y_i for which $\frac{1}{2}k(\mu-3)\ell(f_x) \leq d(y_i, x_r) \leq \frac{1}{2}(k+1)(\mu-3)\ell(f_x)$. Because the distance from x_r to any scheduled transmitter (in the same length class) except x_s is at least $\frac{1}{2}(\mu-3)\ell(f_x)$, the first such layer R_0 does not contain any other scheduled sender y_i. Consider all transmitters y_i in R_k. All corresponding disks D_i must be entirely located in an "extended" ring of area

$$
\begin{aligned}
A(R_k) \ = \ & \left[\left(\frac{(k+1)(\mu-3)\ell(f_i)}{2} + \frac{(\mu-2)\ell(f_i)}{4}\right)^2 \right. \\
& \left. - \left(\frac{k(\mu-3)\ell(f_i)}{2} - \frac{(\mu-2)\ell(f_i)}{4}\right)^2\right]\pi \\
< \ & \frac{1}{4}\left[\left(k+\frac{3}{2}\right)^2 - \left(k-\frac{1}{2}\right)^2\right](\mu-2)^2\ell(f_i)^2\pi \\
= \ & \left(k+\frac{1}{2}\right)(\mu-2)^2\ell(f_i)^2\pi.
\end{aligned}
$$

Each transmitter y_i in R_k has distance at least $\frac{1}{2}k(\mu-3)\ell(f_x)$ from x_r and sends with a power at most $\nu(4\beta n)^{\tau(y_i)} \cdot (2\ell(f_x))^\alpha$. Using the fact that the disks D_i do not overlap, we can bound the interference at x_r from nodes in ring R_k using a standard area argument.

$$
\begin{aligned}
I_r^0(R_k) \ = \ & \sum_{y_i \in R_k} I_r(y_i) \\
\leq \ & \frac{A(R_k)}{A(D_i)} \cdot \frac{\nu(4\beta n)^{\tau(y_i)} \cdot (2\ell(f_x))^\alpha}{(\frac{1}{2}k(\mu-3)\ell(f_x))^\alpha} \\
\leq \ & \frac{16(k+\frac{1}{2})\nu(4\beta n)^{\tau(x_s)} \cdot 2^{2\alpha}}{k^\alpha(\mu-3)^\alpha} \\
\leq \ & \frac{24\nu(4\beta n)^{\tau(x_s)} \cdot 2^{2\alpha}}{k^{\alpha-1}(\mu-3)^\alpha}.
\end{aligned}
$$

Summing up the interferences over all rings yields

$$
\begin{aligned}
I_r^0 \ < \ & \sum_{k=1}^{\infty} I_r^0(R_k) \leq \frac{24\nu(4\beta n)^{\tau(x_s)} \cdot 2^{2\alpha}}{(\mu-3)^\alpha}\sum_{k=1}^{\infty}\frac{1}{k^{\alpha-1}} \\
< \ & \frac{24\nu(4\beta n)^{\tau(x_s)} \cdot 2^{2\alpha}}{(\mu-3)^\alpha} \cdot \frac{\alpha-1}{\alpha-2} \\
< \ & \frac{\nu}{4}\beta^{\tau(x_s)-1}(4n)^{\tau(x_s)},
\end{aligned}
$$

where the second-to-last inequality follows from a standard bound for Riemann's zeta-function and the last one from plugging in the definition of μ. $\qquad\square$

Finally, we bound the interference created by links that are in higher length classes than f_x.

Lemma 4.5: Consider a scheduled link f_x with intended sender x_s and receiver x_r. Let I_r^+ be the total interference caused at x_r by simultaneously transmitting nodes y_i for which $\tau(y_i) < \tau(x_s)$. It holds that

$$I_r^+ \leq \frac{\nu}{4}\beta^{\tau(x_s)-1}(4n)^{\tau(x_s)}.$$

Proof: In $\mathcal{F}$, every sender has a link to its closest neighbor and hence, $\ell(f_y) \leq d(y_i, x_r)$ for all links f_y with intended transmitter y_i. The interference at x_r caused by y_i is therefore at most

$$
\begin{aligned}
I_r(y_i) &= \frac{P_i}{d(y_i, x_r)^\alpha} \leq \frac{\nu(4\beta n)^{\tau(y_i)}\ell(f_y)^\alpha}{\ell(f_y)^\alpha}\\
&= \nu(4\beta n)^{\tau(y_i)} \leq \nu(4\beta n)^{\tau(x_s)-1}.
\end{aligned}
$$

Summing up over all nodes concludes the proof, i.e.,

$$I_r^+ = \sum_{y_i:\tau(y_i)<\tau(x_s)} I_r(y_i) \leq \frac{\nu}{4}\beta^{\tau(x_s)-1}(4n)^{\tau(x_s)}.$$

$\square$

Having thus bounded the interference caused by links in all different length classes in Lemmas 4.3, 4.4, and 4.5, establishing the correctness of the algorithm is now easy. In particular, we can show that every transmitted message is successfully received throughout the algorithm.

Theorem 4.6: Consider an arbitrary time-slot t. All scheduled transmissions E_t in t are received successfully by the intended receivers. That is, the computed schedule is correct.

Proof: Consider a link f_x with intended sender x_s and receiver x_r, that is scheduled for transmission in time-slot t, i.e., $\phi_t(x_s) > 0$. The transmission power of x_s is $P_x = \nu(4\beta n)^{\tau(x_s)}\ell(f_x)^\alpha$. By Lemmas 4.3, 4.4, and 4.5, we know that the total interference faced at x_r is at most

$$I_r \leq I_r^- + I_r^0 + I_r^+ \leq \frac{3\nu}{4}\beta^{\tau(x_s)-1}(4n)^{\tau(x_s)}.$$

Hence, defining $X := \beta^{\tau(x_s)-1}(4n)^{\tau(x_s)} > 1$, the SINR at the x_r is lower-bounded by

$$SINR \geq \frac{\frac{\nu(4\beta n)^{\tau(x_s)}\ell(f_x)^\alpha}{\ell(f_x)^\alpha}}{N + \frac{3\nu}{4}\beta^{\tau(x_s)-1}(4n)^{\tau(x_s)}} \geq \frac{4\beta X}{1+3X} > \beta.$$

In view of Inequality (1), this proves that every transmission scheduled by Algorithm 1 is successfully received. $\square$

Proving correctness (i.e., the absence of collisions) of the computed schedule is only one side of the story. In order to obtain an efficient schedule, we must also guarantee that many links can simultaneously be scheduled, such that all communication requests can quickly be satisfied. In Lemma 4.8, we derive a lower bound on the amount of *progress* achieved in every single time-step of the algorithm. Specifically, we show that at each time t, at least a $\Omega(\frac{1}{\log n})$ fraction of the links in $\mathcal{F}$ that remain to be scheduled *are* indeed scheduled.

Before proving this claim in Lemma 4.8, however, we require the following geometric helper lemma.

Lemma 4.7: Consider a disk C with radius r_c, and disks D_i with centers c_i and radius r_i, $r_i \geq r_c$ for all i. Let κ be the maximal number of such disks D_i such that both of the following properties hold:

- Every D_i overlaps with C in at least one point.
- No disk D_i contains a center c_j for $i \neq j$.

Then, it holds that $\kappa \leq 12$.

Proof: The proof follows a standard geometry argument. Assume for contradiction that there are 13 disks D_i that fulfil both properties stated in the lemma and consider the corresponding centers c_i. There must be a cone of angle $\frac{\pi}{3}$ centered at r_c that contains at least 3 such centers c_1, c_2, c_3. Consider the two senders that are closest to c_r, say c_1 and c_2. Because the cone's angle is $\frac{\pi}{3}$ and $r_i \geq r_c$ for every disk, c_3 must be closer to either c_1 or c_2 than to any point in C. Hence, D_3 either violates the lemma's first or second property. $\square$

Using Lemma 4.7, we can derive the following key lemma that establishes the bound on the algorithm's progress.

Lemma 4.8: Consider an arbitrary time-slot t during the execution of Algorithm 1 and let $\mathcal{F}$ be the set of links that remain to be scheduled at the beginning of time-slot t. Let the constant κ be as defined in Lemma 4.7. It holds that for some constant $\chi > 0$,

$$|E_t| \geq \frac{|\mathcal{F}|}{(\log_\alpha n + 2)\kappa + 4(\mu+4)^2} \geq \chi\frac{|\mathcal{F}|}{\log_\alpha n}.$$

Proof: The proof is based on a geometric argument. We show that for every link selected in steps 7-9 of the scheduling subroutine, there can be at most a logarithmic number of longer communication links that are deleted from $\mathcal{F}_t$, i.e., not scheduled in time-slot t. On the other hand, when choosing a link, the algorithm removes no previously selected shorter links from $\mathcal{F}_t$. From this observation, the lemma then follows.

Consider an arbitrary iteration of the inner-most while loop in which the link f^* is selected for transmission in time-slot t, i.e., $E_t = E_t \cup \{f^*\}$. Assume that x_s is the transmitting node of f^*. Because the algorithm considers the links in $\mathcal{F}_t$ in *increasing* order, only links that are *longer* than f^* can be dropped from $\mathcal{F}_t$ in steps 12-15 of the same while-loop iteration. When bounding the number of longer links that can be dropped due to the scheduling of f^*, we distinguish two cases. First, we bound $P^0(f^*)$ which denotes the number of dropped links that are in the same length class as f^* (Line 13). Secondly, we consider $P^+(f^*)$, i.e., the number of links in higher length classes erased from $\mathcal{F}_t$ in the same iteration of the while-loop (Line 15).

We start with $P^0(f^*)$. For each dropped link f_{uv} with $\tau(f^*) = \tau(f_{uv})$, it holds that $2\ell(f^*) \geq \ell(f_{uv}) \geq \ell(f^*)$. Consider for every f_{uv} a disk D_u of radius $\frac{1}{2}\ell(f_{uv})$ around its transmitter x_u. Because $\mathcal{F}_t \subseteq \mathcal{F}_p$ and because $\mathcal{F}_p$ is a nearest neighbor forest, disks D_u do not overlap. Furthermore, the area of each such disk is

$$A(D_u) = \frac{1}{4}\ell(f_{uv})^2\pi \geq \frac{1}{4}\ell(f^*)^2\pi.$$

By the condition given in step 12 of the scheduling subroutine, a link f_{uv} with $\tau(f^*) = \tau(f_{uv})$ is dropped only if $x_v \in B(x_s, \mu\ell(f^*))$. Hence, the transmitting node x_u must be located within distance $\mu\ell(f^*) + \ell(f_{uv}) \leq (\mu+2)\ell(f^*)$ of x_s. That is, all disks D_u corresponding to removed links are entirely contained in a disk D^* centered at x_s with radius $(\mu+4)\ell(f^*)$. Thus, the number of dropped links in $P^0(f^*)$ is bounded by

$$P^0(f^*) \leq \frac{(\mu+4)^2\ell(f^*)^2\pi}{\frac{1}{4}\ell(f^*)^2\pi} = 4(\mu+4)^2.$$

We now turn our attention to the more complicated case $P^+(f^*)$. Recall that the sending node of f^* is x_s and denote by $f_1, \ldots, f_k$ all links that are dropped from $\mathcal{F}_t$ in Lines 14 and 15 of the schedule subroutine for which $\tau(f^*) > \tau(f_i)$, i.e., $\delta_{si} > 0$. For each such link f_i, s_i and r_i denote its intended sender and receiver, respectively. The links f_i are ordered according to the distance $d(x_s, r_i)$, where f_1 is the link whose r_1 is the closest intended receiver from x_s.

By the definition of the algorithm, a link f_i is dropped if and only if $r_i \in B(x_s, (4\beta n)^{\frac{\delta_{si}+1}{\alpha}}\ell(f^*))$. Turning this argument around, we observe that a link f_i whose receiver r_i is at distance more than $(4\beta n)^{\frac{\varphi}{\alpha}}\ell(f^*))$ from x_s, is only dropped if $\delta_{si} + 1 > \varphi$ and consequently, $\delta_{si} \geq \varphi$. In combination with Lemma 4.2, this yields the fact that for a dropped link f_i with

$$d(x_s, r_i) > (4\beta n)^{\frac{\varphi}{\alpha}}\ell(f^*), \qquad (3)$$

the length $\ell(f_i)$ of the link must be at least

$$\ell(f_i) \geq \frac{1}{2}(4\beta n)^{\varphi}\ell(f^*). \qquad (4)$$

In the following, consider an exponentially growing series of disks C_j, $j = 1, 2, \ldots$ of radius $r_j = (4\beta n)^{\frac{j}{\alpha}}\ell(f^*)$ centered at x_s. Furthermore, define a *ring* R_j as the area $C_{j+1} \setminus C_j$, i.e., it holds for every node $x_i \in R_j$ that

$$(4\beta n)^{\frac{j}{\alpha}}\ell(f^*) < d(x_s, x_i) \leq (4\beta n)^{\frac{j+1}{\alpha}}\ell(f^*).$$

A key observation for the proof is that there cannot be many links dropped from rings which are close to one another. This intuition is formalized using two helper lemmas. Lemma 4.9 shows that there can only be a constant number of receivers in the first three rings. In Lemma 4.10 we then prove that if for an arbitrary i, the receiver r_i is located in R_k, $k \geq 3$, there cannot be more than κ other intended receivers from dropped links in the subsequent $\alpha(k-1) - 1$ rings.

Lemma 4.9: It holds that $r_{2\kappa+1}$ is located outside of C_3, i.e., at most 2κ links with receiver in C_3 are dropped from $\mathcal{F}_t$.

Proof: First, consider all links f_i for which $\delta_{si} = 1$. Each such link has length at least $\ell(f_i) \geq \frac{1}{2}(4\beta n)\ell(f^*)$. Since f_i was dropped, its receiver must be located within distance $(4\beta n)^{1/\alpha}\ell(f^*)$ of x_s. For $\beta \geq 1$ and $\alpha > 2$, it holds that $(4\beta n)^{1/\alpha} \leq \frac{1}{2}(4\beta n)$. Now, assume for contradiction that $\kappa+1$ or more links f_i with $\delta_{si} = 1$ exist. Also, draw a disk C of radius $(4\beta n)^{1/\alpha}\ell(f^*)$ around x_s, and disks D_i of radius $\ell(f_i)$

around each corresponding sender s_i. Notice that there are at least $\kappa+1$ disks D_i each of which overlaps with disk C in at least one point (where r_i is located) and no disk D_i contains the center of another disk D_j, because the links f_i form a nearest neighbor forest. However, the possibility of packing $\kappa + 1$ disks D_i in such a way contradicts Lemma 4.7 and hence, it follows that there can be at most κ links f_i with $\delta_{si} = 1$.

Next, we bound the remaining number of dropped links f_i whose receivers r_i are situated in C_3. Each of these remaining links has length at least $\frac{1}{2}(4\beta n)^2\ell(f^*)$ because $\delta_{is} \geq 2$. Moreover, all receivers are located in C_3, that is, $d(x_s, r_i) \leq (4\beta n)^{\frac{3}{\alpha}}\ell(f^*) < \frac{1}{2}(4\beta n)^2\ell(f^*)$. Again, it follows by Lemma 4.7 that the number of dropped links with $\delta_{si} \geq 2$ in C_3 is upper-bounded by κ. $\square$

Lemma 4.10: It holds for all i that if $r_i \in R_k$, $k \geq 3$, then $r_{i+\kappa} \in R_h$ for $h > \alpha(k-1)$. That is, for any $k \geq 3$, there can be at most κ dropped receivers in rings $R_k, \ldots, R_{\alpha(k-1)}$.

Proof: It follows from equations (3), (4), and the definition of a ring, that every dropped link with receiver in rings $R_k, \ldots, R_{\alpha(k-1)}$ must be of length at least $\frac{1}{2}(4\beta n)^k\ell(f^*)$ (otherwise, it would not be dropped). On the other hand, the distance between a receiver in these rings and x_s is at most

$$d(x_s, r_i) \leq (4\beta n)^{\frac{\alpha(k-1)+1}{\alpha}}\ell(f^*) = (4\beta n)^{k-1+\frac{1}{\alpha}}\ell(f^*)$$
$$< \frac{1}{2}(4\beta n)^k\ell(f^*) \leq \ell(f_i),$$

where the second to last inequality holds for $\beta \geq 1$, $\alpha > 2$, and $k \geq 3$. It follows that like in the proof of Lemma 4.9, we can draw a disk D_i with radius $\ell(f_i)$ around each sender s_i having its receiver r_i in rings $R_k, \ldots, R_{\alpha(k-1)}$. Each of these disks must overlap with the disk centered at x_s of radius $(4\beta n)^{k-1+\frac{1}{\alpha}}\ell(f^*)$ and no disk D_i contains the center of another disk. Hence, as illustrated in Figure IV, it follows by Lemma 4.7 that there can be at most κ dropped links with receiver in rings $R_k, \ldots, R_{\alpha(k-1)}$. $\square$

Having proven Lemmas 4.9 and 4.10, we can now bound the total number of dropped links and thus conclude the proof of Lemma 4.8. By Lemma 4.9, we know that at most the first 2κ receivers $r_1, \ldots, r_{2\kappa}$ can be located in C_3. All other receivers must be located in a ring R_k for $k \geq 3$. By applying Lemma 4.10, it follows that the receiver $r_{3\kappa+1}$ cannot be closer to x_s than in ring $R_{\alpha(k-1)} = R_{2\alpha}$, receiver $r_{4\kappa+1}$ cannot be closer than in ring $R_{2\alpha^2-\alpha}$, and so forth. By thus recursively applying Lemma 4.10, it follows that receiver $r_{(j+2)\kappa+1}$ cannot be closer than in ring R_{λ_j}, where λ_j is

$$\lambda_j = 2\alpha^j - \sum_{h=1}^{j-1} \alpha^h > \alpha^j.$$

Because there are at most n different length classes, the last ring from which a receiver (and its link) can be dropped is R_n. Consequently, the total number of links that can be dropped when selecting link f^* is at most $(j_m + 2)\kappa$, where
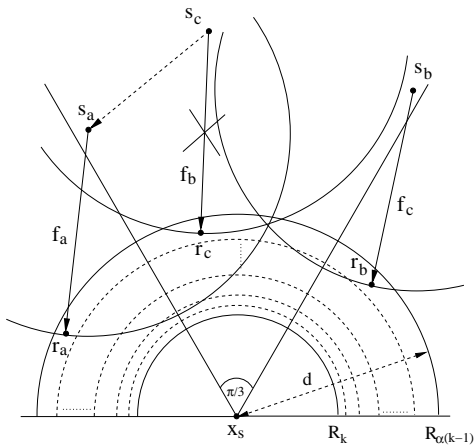
$$\alpha^{j_m} \leq n \implies j_m \leq \log_\alpha n,$$

Fig. 3. Illustration of the proof of Lemma 4.10. Because the length of links f_a, f_b, and f_c is larger than the radius of the disk in which all receivers must be located, at most κ such links can exist. In the example, the closest neighbor of s_c is s_a and not r_c, which yields the contradiction.

which implies $P^+(f^*) \leq (\log_\alpha n + 2)\kappa$.

In summary, for every link that is selected in the inner-most while-loop of the schedule subroutine for scheduling in a time-slot t, at most

$$P^0(f^*) + P^+(f^*) \ \leq \ 4(\mu+4)^2 + (\log_\alpha n + 2)\kappa$$

communication links are dropped from $\mathcal{F}_t$. Therefore, the number of communication links $|E_t|$ that are scheduled in time-slot t is at least

$$|E_t| \ \geq \ \frac{|\mathcal{F}|}{4(\mu+4)^2 + (\log_\alpha n + 2)\kappa + 1},$$

from which Lemma 4.8 follows. $\qquad\square$

Lemma 4.8 provides us with a lower bound on the amount of progress achieved by the algorithm when scheduling the links selected in one *phase*. In particular, it allows us to derive a bound on the time required to schedule the nearest neighbor forest in this phase. However, we also need to bound the number of phases that the algorithm executes before termination. This is done in the following lemma.

Lemma 4.11: Let $\mathcal{A}_p$ denote the set of active nodes at the beginning of phase p during the execution of Algorithm 1. For each p, it holds that $|\mathcal{A}_{p+1}| \leq |\mathcal{A}_p|/2$.

Proof: In line 11 of Algorithm 1 all nodes that have an outgoing link (i.e., that transmit during this phase) are removed from $\mathcal{A}$. Consider the connected components of forest $\mathcal{F}_p$. In each such connected component, there is at most 1 node that has no outgoing link, because each connected component forms a directed tree with a unique sink. The claim follows because each connected component consists of at least two nodes. $\qquad\square$

Finally, we are ready to prove the main theorem of this section containing the claimed correctness and efficiency results of Algorithm 1.

Theorem 4.12: For every network, Algorithm 1 produces a correct schedule $\mathcal{S}$ that induces a strongly connected subgraph. Furthermore, the length of the schedule is $T(\mathcal{S}) \in O(\log^4 n)$.

Proof: As for the scheduling complexity, we start by showing that every subroutine call requires at most $O(\log^2 n)$ time-slots. Let m denote the total number of links that are to be scheduled during a subroutine call, i.e., initially $|\mathcal{F}| = m \leq n$. By Lemma 4.8, at least a $\Omega(\frac{1}{\log n})$ fraction of the links in $\mathcal{F}$ that remain to be scheduled are scheduled in each time-slot in the subroutine. After the first time-slot, at least $\chi\frac{m}{\log_\alpha n}$ nodes have been scheduled. Generally, by repeatedly applying Lemma 4.8, it follows that after the kth time-slot, for $k = \ln m \cdot \log_\alpha n$, the number of links that have not yet been scheduled is at most

$$m \cdot \left(1 - \frac{\chi}{\log_\alpha n}\right)^k \ \leq \ m \cdot e^{-\chi \ln m} \ < \ 1.$$

That is, each invocation of the *schedule()* subroutine requires at most $O(\log^2 n)$ time-slots. The number of subroutine calls in each phase of the algorithm (lines 12-14) is clearly $\log(4\beta n)$.

All that remains to be done to derive the algorithm's scheduling complexity is to bound the number of phases. By Lemma 4.11, the number of active nodes is at least halved in every phase. Therefore, at most $\log n$ phases are required until there remains only a single active node upon which the algorithm terminates. Putting everything together, the algorithm's scheduling complexity is

$$T(\mathcal{S}) \ \leq \ \ln m \cdot \log_\alpha n \cdot \log(4\beta n) \cdot \log n \in O(\log^4 n).$$

By Theorem 4.6, every transmitted message is successfully received. Furthermore, observe that the union of all scheduled links $\mathcal{F}_p$ forms a directed tree towards a single node (the one node that remains active at the end) in the network. This node can then connect the network with a single transmission. Hence, the union of all scheduled links is strongly connected, i.e., there exists a path between all pairs of nodes. $\qquad\square$

V. Linear Power Assignment Algorithm

Clearly, MAC layer protocols based on uniform or linear power assignment strategies have the practical advantage that their implementation is particularly simple. The lower bound of Section III for uniform or linear power assignment protocols is based on a network in which some communication links are exponentially longer than others. This raises the question whether the performance of uniform or linear power assignment approaches may also deteriorate as badly in case the length of the communication link is less varied. If, for instance, linear power assignment strategies perform well in randomly deployed average-case networks, heuristic protocols featuring simple linear power assignments may be employed in non-critical networks in spite of their bad worst-case scheduling complexity.

In this section, we propose and analyze an algorithm that adopts a *linear power assignment*. Specifically, we show that this algorithm performs poorly only in scenarios in which there

Algorithm 2 Linear Power Assignment Algorithm

Input: An arbitrarily located set of nodes X
Output: A schedule $\mathcal{S}$ in which every node $x \in X$
    can send successfully to its closest neighbor.
1: For each $x_i \in X$, let f_i be the link to its closest neighbor;
2: Let $\mathcal{L} = L_0, \ldots, L_{\Delta-1}$, such that L_k is the set
   of links of length $2^k \leq \ell(f_i) < 2^{k+1}$;
3: $\mu = 6 \sqrt[\alpha]{\frac{2\beta(\alpha-1)}{\alpha-2}}$; $\rho > 4N$; $t = 1$;
4: **for each** $L_i \neq \emptyset$ **do**
5:    Partition the plane in squares of width $\mu \cdot 2^k$;
6:    **for** $j = 1$ **to** 4 **do**
7:        Select a maximal independent set of squares j
          (cf. Figure 4);
8:        **repeat**
9:            For each selected square R, pick one link
              $f_i \in L_k$ whose intended sender x_r is in R;
10:           $\phi_t(x_i) := \rho \cdot \ell(f_i)^\alpha$;
11:           Do not consider link f_i in future iterations;
12:           $t = t + 1$;
13:       **until** all links in active squares have been picked;
14:   **end for**
15: **end for**
16: $\mathcal{S} := \{\phi_1, \ldots, \phi_{t-1}\}$;

---

are links belonging to many different orders of magnitude. For simplicity of presentation, we again consider the simple network property Ψ_{min}, i.e., we want that every node can transmit successfully at least once. By applying a technique similar to the one in Section IV, strong connectivity can be achieved at the cost of an additional $O(\log n)$ factor in the scheduling complexity.

The *diversity* $g(V)$ of a set of nodes is the number of magnitudes of distances [19]. Formally, $g(V)$ is defined as

$$g(V) := |\{m \mid \exists x_i, x_j \in X : \lfloor \log(d(x_i, x_j)) \rfloor = m\}|.$$

In our case, $g(V)$ denotes the number of non-empty length classes of the nearest neighbor forest links. In the example shown in Figure 1, for instance, the diversity is $g(V) = \log(2^n) = n$. In the sequel, we show that Algorithm 2 achieves a scheduling complexity of $T(S) \in O(g(V))$.

The idea of Algorithm 2 is simple: simultaneously schedule links of similar length, while guaranteeing a large enough buffer distance between each pair of transmitting nodes. In each phase of the algorithm, only links belonging to the same length class are scheduled. In order to schedule one such phase for links of length $2^k \leq \ell(f_i) < 2^{k+1}$, the algorithm partitions the plane into grid-cells of width $\mu \cdot 2^k$. In each time-slot, it chooses a maximal independent set of cells and selects one link in each such cell for scheduling.

In the following, we again establish the claimed correctness and scheduling complexity results.

Lemma 5.1: Every node can send successfully in the unique time-slot t in which $\phi_t(x_i) > 0$.

Proof: The proof follows exactly along the lines of the proof of Theorem 4.4 (only using different constants) and is therefore omitted from this paper. □
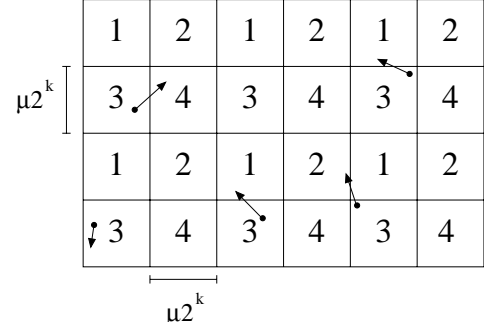

Fig. 4. In line 7 of Algorithm 2, the algorithm picks all squares numbered by j. The example shows an inner-loop iteration for length-class L_k and $j = 3$. The algorithm schedules one unscheduled communication link from each selected square (if there exists one).

Theorem 5.2: The schedule S obtained by Algorithm 2 has length at most $T(S) \in O(g(V))$ and fulfills property Ψ_{min}, i.e., each node can send successfully at least once.

Proof: Correctness follows directly from Lemma 5.1 and from the observation that for every sender x_i, there is a time-slot t for which $\phi_t(x_i) > 0$. As for the length of the schedule, we first observe that there are at most $O(g(V))$ non-empty length classes, i.e., iterations of the outermost loop. Hence, it only remains to prove that a single phase requires only a constant number of time-slots.

Consider the phase in which length-class L_k is scheduled. We first show that the number of potential transmitters in a cell can be at most a constant. Because every transmitting node has a link to its *closest* neighbor, the disks D_i of radius $\frac{1}{2}\ell(f_i) \geq 2^{k-1}$ around each transmitter x_i do not overlap. Consider all nodes located in a cell C. The disks D_i belonging to these nodes are completely contained in a square of side-length $(\mu + 1) \cdot 2^k$. Hence, it follows from the standard area packing argument that the number of links in C is at most $4(\mu + 1)^2 \in O(1)$ in each cell. The proof is now concluded by observing that in a grid, always one fourth of the cells can be scheduled independently as shown in Figure 4. □

In combination with the "growing component" technique used in Section IV, the following theorem can be derived.

Theorem 5.3: Algorithm 2 can be adapted to obtain a scheduling algorithm for the strong-connectivity property with scheduling complexity $O(\min(n, g(V) \cdot \log n))$.

Proof: If $g(V) \cdot \log n < n$, we combine Algorithm 2 with the technique of merging clusters iteratively in each phase, as done in Algorithm 1. Using Lemma 4.11, it requires at most $O(\log n)$ phases (each taking time $O(g(V))$) until the scheduled links form a directed tree towards a single node. If $g(V) \cdot \log n \geq n$, the algorithm can simply schedule each node individually. □

It is interesting to compare the result of Algorithm 2 with the solution given in Section IV. Since the diversity $g(V)$ can be as large as n, the $O(\log^4 n)$ schedule by Algorithm 1 improves the scheduling complexity by an exponential factor.

VI. RELATED WORK

Our work has connections to several different areas that have been the focus of attention in the networking community. In this section, we want to highlight some of these connections.

It is clear that our work has ties to *coloring* problems and more specifically, MAC layer protocols. The scheduling complexity of a wireless network expresses the number of time-slots that are required until a certain set of communication requests can be satisfied. A multiplicity of MAC layers have been proposed for wireless multi-hop networks [5], [26], [20], [22], [14], some of them particularly designed for energy-constrained networks such as sensor networks [30], [32], [23]. The results obtained in Sections III of our paper place strong lower bounds on the amount of time required by most of these MAC protocols in order to schedule a set of requests in the physical model. On the other hand, the algorithm in Section IV theoretically constitutes a close to optimal way of scheduling requests, even when these requests are constrained to satisfy the connectivity property.

The traditional graph-theoretic way of studying scheduling problems results in *coloring* problems. When modeling the network as a graph $G = (V, E)$, the design of a collision-free MAC layer boils down to obtaining a 2-hop coloring of the nodes in the interference graph [25]. The efficiency of such a MAC layer protocol then depends on the number of colors used in the process. Scheduling and coloring problems of this kind have been studied in a variety of papers, e.g., [19], [21], [25], [15]. Algorithmic aspects of *capacity* in wireless networks have also been studied in various graph models, e.g. [16].

One shortcoming of these and related graph-theoretical models is that they ignore the accumulated interference of a large number of distant nodes. Even more severely, these models imply that simultaneous transmissions on proximate links *necessarily* interfere with each other, which is not true in a SINR environment. Specifically, a node x_r may receive a message from x_s in spite of being in the *transmission-range* of other simultaneous transmissions, if the transmission powers of the different senders are properly adjusted. As shown in this paper, such highly *non-linear power assignments* are in fact indispensable ingredients of any fast scheduling algorithm. Studying scheduling in graph-theoretical models simply abstracts away this crucial aspect.

Integrated scheduling-power control problems in the SINR model have been studied in various papers, e.g., [7], [9], [4], [3]. In [4], [3], the impact of power assignments to nodes on the achievable throughput capacity is studied, whereas [7], [9] study the problem of finding a schedule and power control policy that minimizes the total average transmission power in the wireless multi-hop network. None of the above papers provides a bound on the scheduling complexity in wireless networks. Moreover, the proposed algorithms do either not yield provable worst-case guarantees or are based on solutions to complex optimization problems that can only be solved in exponential time in the number of links or nodes in the network.

Not surprisingly, the study of *connectivity* in networks has always been of great interest to the networking community. Using results from percolation theory, [11] investigates the critical power level that is necessary for a *randomly deployed* wireless network to become connected under the assumption that all nodes transmit at the same power level. Ever since, much research effort has been directed towards studying asymptotic connectivity requirements in randomly distributed wireless networks, e.g. [31], [8]. What these papers do not consider, however, is the complexity of actually *scheduling* the communication links that form the connected network.

While the above papers study connectivity in *random networks*, the study of *topology control* has typically focused on arbitrary, possibly worst-case networks. In topology control, the idea is that instead of each node transmitting at a maximum power level, the nodes collaborate to determine a transmission power that results in a network topology which exhibits favorable properties. Not surprisingly, the first topology control protocols focused on guaranteeing the most basic of all network properties [24], i.e., *connectivity*. Subsequently, a lot of effort has been made in developing more subtle network structures that combine desirable properties such as low energy paths, low node degrees, planarity, or sparseness, e.g., [28], [27], [18] or to study heterogenous networks [17].

The problem of topology control is that it assumes a static network model, i.e. without considering the fact that the selected communication links may not be scheduled in a reasonable amount of time. In this regard, topology control is a theoretic notion that abstracts away physical restrictions caused by interference. A recent step in the evolution of topology control has been to explicitly take into consideration the issue of *interference*. Although again based on a *static* graph-theoretical communication model, [6] proves that all classic topology control algorithms fail to actually reduce interference in wireless networks.

VII. CONCLUSIONS

Clearly, our results in this paper are not directly comparable to results on the capacity of wireless networks. Nonetheless, it is intriguing to discuss the connections between these two complementing approaches. The results on the capacity [12] of wireless networks essentially give a negative answer to the possibilities of wireless networks by limiting the throughput that can be achieved per node as the number of nodes in the network grows. In contrast, our result is of a more positive nature. Specifically, our result shows that when using proper MAC layer protocols and power assignment schemes, complex communication requests can theoretically be scheduled efficiently even in large-scale worst-case networks. This implies that unlike for the capacity, there exists only little theoretical limitation to scaling as far as the actual *scheduling* of

transmissions in wireless networks is concerned. Interestingly, however, such a fast solution cannot be achieved using the intuitive uniform or linear power assignment schemes that have been widely adopted by standard MAC layer and scheduling solutions. Instead, we have shown that a more subtle, non-linear handling of the transmission powers at nodes is required in order to achieve efficient and scalable solutions.

Our work opens a wide range of directions for future research. Most obviously, it would be interesting to investigate the scheduling complexity of other network properties. Moreover, it would be intriguing to gain a deeper understanding of the connections between the notion of *capacity*, *interference*, and the *scheduling complexity* of wireless networks. We are convinced that a thorough understanding of these notions could help in designing better network protocols.

VIII. ACKNOWLEDGEMENTS

REFERENCES

[1] S. Banerjee and A. Misra. Minimum Energy Paths for Reliable Communication in Multi-Hop Wireless Networks. In *Proceedings of the 3rd ACM International Symposium on Mobile Ad Hoc Networking & Computing (MOBIHOC)*, pages 146–156, 2002.

[2] A. Behzad and I. Rubin. On the Performance of Graph-based Scheduling Algorithms for Packet Radio Networks. In *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM)*, pages 3432–3436, 2003.

[3] A. Behzad and I. Rubin. Impact of Power Control on the Performance of Ad Hoc Wireless Networks. In *Proceedings of the 24th Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, 2005.

[4] A. Behzad, I. Rubin, and A. Mojibi-Yazdi. Distributed Power Controlled Medium Access Control for Ad-Hoc Wireless Networks. In *Proceedings of the 18th IEEE Annual Workshop on Computer Communications (CCW)*, pages 47–53, 2003.

[5] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang. MACAW: A Media Access Protocol for Wireless LANS. In *Proceedings on Communications architectures & protocols (SIGCOMM)*, pages 212–225, 1994.

[6] M. Burkhart, P. von Rickenbach, R. Wattenhofer, and A. Zollinger. Does Topology Control Reduce Interference? In *Proceedings of the 5th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, pages 9–19, 2004.

[7] R. L. Cruz and A. V. Santhanam. Optimal Routing, Link Scheduling and Power Control in Multi-hop Wireless Networks. In *Proceedings of the 22th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, 2003.

[8] O. Dousse, F. Baccelli, and P. Thiran. Impact of Interferences on Connectivity in Ad Hoc Networks. In *Proceedings of the 22th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, 2003.

[9] T. ElBatt and A. Ephremides. Joint Scheduling and Power Control for Wireless Ad-hoc Networks. In *Proceedings of the 21th Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, 2002.

[10] J. Grönkvist and A. Hansson. Comparison Between Graph-Based and Interference-Based STDMA Scheduling. In *Proceedings of the 2nd ACM International Symposium on Mobile Ad Hoc Networking & Computing (MOBIHOC)*, pages 255–258, 2001.

[11] P. Gupta and P. R. Kumar. Critical Power for Asymptotic Connectivity in Wireless Networks. *Stochastic Analysis, Control, Optimization and Applications: A Volume in Honor of W. H. Fleming (March 1998)*, pages 547–566, 1998.

[12] P. Gupta and P. R. Kumar. The Capacity of Wireless Networks. *IEEE Trans. Information Theory*, 46(2):388–404, 2000.

[13] R. Hekmat and P. van Mieghem. Interference in Wireless Multi-Hop Ad-hoc Networks and Its Effect on Network Capacity. *Wireless Networks*, 10:389–399, 2004.

[14] E.-S. Jung and N. H. Vaidya. Power Control MAC Protocol for Ad Hoc Networks. In *Proceedings of the 8th International Conference on Mobile Computing and Networking (MOBICOM)*, pages 36–47, 2002.

[15] V. Kumar, M. Marathe, S. Parthasarathy, and A. Srinivasan. End-to-End Packet-Scheduling in Wireless Ad-hoc Networks. In *Proceedings of the 15th Symposium on Discrete Algorithms (SODA)*, pages 1021–1030, 2004.

[16] V. Kumar, M. Marathe, S. Parthasarathy, and A. Srinivasan. Algorithmic Aspects of Capacity in Wireless Networks. In *Proceedings of the International Conference on Measurement & Modeling of Computer Systems (SIGMETRICS)*, 2005.

[17] N. Li and J. Hou. Topology Control in Heterogenous Wireless Networks: Problems and Solutinos. In *Proceedings of the 23rd Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, 2004.

[18] X.-Y. Li, W.-Z. Song, and W. Wang. A Unified Energy Efficient Topology for Unicast and Broadcast. In *Proceedings of the 11th International Conference on Mobile Computing and Networking (MOBICOM)*, 2005.

[19] F. Meyer auf der Heide, C. Schindelhauer, K. Volbert, and M. Grünewald. Energy, Congestion and Dilation in Radio Networks. In *Proceedings of the 14th ACM Symposium on Parallel Algorithms and Architectures (SPAA)*, pages 230–237, 2002.

[20] P. Monks, V. Bharghavan, and W. W. Hwu. A Power Controlled Multiple Access Protocol for Wireless Packet Networks. In *Proceedings of the 20th Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, pages 1567–1576, 2001.

[21] T. Moscibroda and R. Wattenhofer. Coloring Unstructured Radio Networks. In *Proceedings of the 17th Symposium on Parallel Algorithms and Architectures (SPAA)*, pages 39–48, 2005.

[22] S. Narayanaswami, V. Kawadia, R. S. Sreenivas, and P. R. Kumar. Power Control in Ad-Hoc Networks: Theory, Architecture, Algorithm and Implementation of the COMPOW Protocol. In *Proceedings of the European Wireless Conference*, 2002.

[23] J. Polastre, J. Hill, and D. Culler. Versatile Low Power Media Access for Wireless Sensor Networks. In *Proceedings of the 2nd Conference on Embedded Networked Sensor Systems (SENSYS)*, pages 95–107, 2004.

[24] R. Ramanathan and R. Rosales-Hain. Topology Control of Multihop Wireless Networks Using Transmit Power Adjustment. In *Proceedings of the 19th Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, 2000.

[25] S. Ramanathan and E. L. Lloyd. Scheduling Algorithms for Multi-Hop Radio Networks. In *Conference proceedings on Communications architectures & protocols (SIGCOMM)*, pages 211–222, 1992.

[26] S. Singh and C. S. Raghavendra. PAMAS - Power Aware Multi-Access Protocol with Signalling for Ad Hoc Networks. *SIGCOMM Comput. Commun. Rev.*, 28(3):5–26, 1998.

[27] W.-Z. Song, Y. Wang, and X.-Y. Li. Localized Algorithms for Energy Efficient Topology in Wireless Ad Hoc Networks. In *Proceedings of the 5th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, 2004.

[28] R. Wattenhofer, L. Li, P. Bahl, and Y.-M. Wang. Distributed Topology Control for Power Efficient Operation in Multihop Wireless Ad Hoc Networks. In *Proceedings of the 20th Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, 2001.

[29] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides. Energy-Efficient Broadcast and Multicast Trees in Wireless Networks. *Mobile Networks and Applications (MONET)*, 7(6):481–492, 2002.

[30] A. Woo and D.-E. Culler. A Transmission Control Scheme for Media Access in Sensor Networks. In *Proceedings of the 7th International Conference on Mobile Computing and Networking (MOBICOM)*, pages 221–235, 2001.

[31] F. Xue and P. R. Kumar. The Number of Neighbors Needed for Connectivity of Wireless Networks. *Wireless Networks*, 2003.

[32] W. Ye, J. Heidemann, and D. Estrin. An Energy-Efficient MAC Protocol for Wireless Sensor Networks. In *Proceedings of the 21th Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, pages 1567–1576, 2002.