EURASIP Journal on
Wireless Communications and Networking
a SpringerOpen Journal

RESEARCH Open Access

Efficient link scheduling for rechargeable wireless ad hoc and sensor networks

Guodong Sun[1], Guofu Qiao[2*] and Lin Zhao[1]

Abstract

We study the link scheduling for rechargeable wireless ad hoc and sensor networks to improve network throughput. Efficient link scheduling is highly critical for wireless networks, and a lot of efforts have been made to address this issue. The previous work which only considers how to reduce interference within close-by links, however, fails to improve the network throughput for rechargeable wireless network in which a node, powered by ambient energy (such like solar and vibration) has to be recharged before it can communicate with other nodes. Thus, link scheduling with rechargeable nodes is constrained not only by co-channel interference but also by energy replenishment of the node. In this paper, we develop efficient link scheduling algorithms to improve the network throughput of rechargeable wireless ad hoc and sensor networks. The proposed design, based on realistic network models and mathematical formulation, achieves constant ratio of the best solution. Besides the theoretical analysis, simulation experiments are conducted to evaluate our algorithms.

Introduction

Many wireless network applications, such as sensor networks and wireless ad hoc networks, are expected to operate a long time with limited energy source. Harvesting the environmental energy to power wireless nodes is a promising method to address the conflict between node's limited energy and application lifetime requirements. For example, the renewable energy transformed from solar, vibration, thermo, water flow, and corrosion can be employed to recharge wireless nodes [1-6]. However, two major characteristics of rechargeable wireless network make it more difficult to optimize communication performances, in comparison with the traditional wireless network without depending on renewable energy.

First, the operation of rechargeable node often has to be intermittent. In practice, the energy-harvesting rate is usually lower than the energy consumption rate. For example, the energy generated by mechanical vibration is about the level 300 μW/cm^2 [2], that is, a vibration energy harvester with a surface of 40 cm^2 can output 12 mW of energy. If such harvester powers Micaz mote working with 60 mW of energy, a recharging cycle of at least 5 s is needed before a 1-s task can be carried out. For the thermoelectric energy harvester fitted on a wristwatch of body health care sensor network [5], it outputs energy at the level of 22 μW, and then it needs a longer recharging duration. Hence, a node has to wait until it accumulates sufficient energy before it can perform tasks of computation and communication. For applications demanding some degree of real-time , therefore, the network throughput would deteriorate significantly without link scheduling.

Second, the ability of harvesting energy is usually heterogeneous among distinct nodes. This is because the energy-producing ability of environments and the energy-accumulating ability of harvesters are possibly different. For instance, in a vibration-powered scenario, a harvester with a small surface of sensing vibration signals will output power lower than that output by the harvester with a large sensing surface, and then it may need more time for recharging before communication can be carried out. Thus, a new challenge facing the link scheduling is that the heterogeneity of ambient power sources must be accounted for.

In this paper, we focus on the link scheduling problem for a rechargeable wireless network to improve the network throughput. Even though link scheduling has attracted a lot of attention in recent years [7-16], the previous works, which only consider how to avoid the co-channel interference in throughput improvement, often

*Correspondence: qgfhit@163.com
[2]School of Civil Engineering, Harbin Institute of Technology, 77 Huanghe Rd, Harbin 150006, China
Full list of author information is available at the end of the article

gives undesirable or even failed answers in the case with energy replenishment. These previous link scheduling policies have shown their efficiency in network throughput and energy consumption. However, we argue that these polices are all based on a precondition: the two end-nodes of each link is available in terms of energy whenever it is to be scheduled. But this precondition cannot always be satisfied in many energy-rechargeable wireless networks, especially scenarios with extremely low energy-harvesting rate (such as applications powered by vibration, corrosion, or thermo energy).

For a simple four-node network operating under successive, synchronized time slots, Figures 1 and 2 show two link schedules with and without considering energy replenishment, respectively. Suppose here that links (u_2, u_3) and (u_1, u_4) of Figures 1 and 2 interfere with each other. Figure 1 shows a schedule using three distinct time slots for all four links. Figure 2 considers the restriction posed by both interference and energy replenishment, and assumes that each node needs some time units to recharge its energy buffer. For simplicity, we assume that the energy buffer of each node contains only energy that transmits or receives a single packet. If a node recharges fully at time slot t, it can then communicate at $t + 1$. If the energy buffers of all four nodes of Figure 2 are empty at time 0, link (u_1, u_2) can just be scheduled after time 6, even though u_1 will have already recharged fully at time 2. Similarly, due to the restriction of energy replenishment, we complete scheduling all four links of Figure 2 until time 12. Therefore, we can see that the couple of interference and energy replenishment determines the link scheduling and its performance. Moreover, it is worth pointing out that the total schedule time in Figure 2 should not be measured by the number of time assignments for all links, as done in Figure 1, but by the span of time units for all links. This is an important difference from the previous wireless link scheduling without energy replenishment or from the frequency assignment for cellular network.

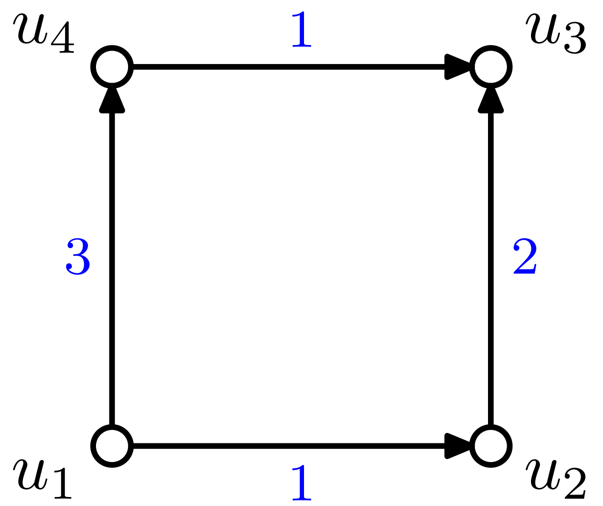

Figure 1 A link schedule with only interference restriction. In this case, links (u_2, u_3) and (u_1, u_4) interfere with each other, and the blue number represents the assigned time slot.

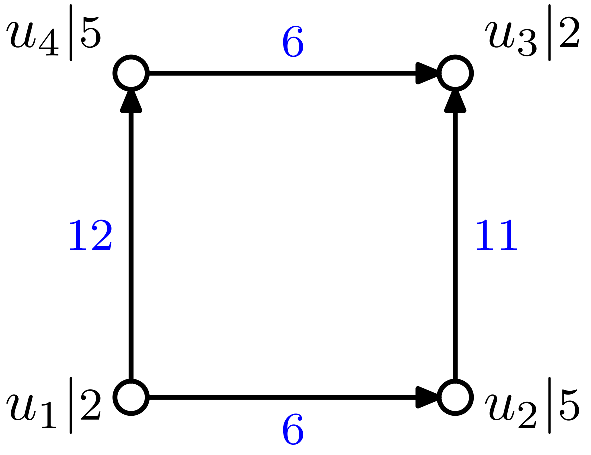

Figure 2 A link schedule with restrictions of interference and energy replenishment. In this case, links (u_2, u_3) and (u_1, u_4) interfere with each other, and the blue number represents the assigned time slot. Here, $u_i|x$ means that u_i needs x time slots to harvest the energy enough to transmit/receive a single packet.

The main objective of our work is to find a link scheduling for a rechargeable wireless network, using a period as small as possible, while guaranteeing that each node has enough energy when being scheduled. To efficiently address the link scheduling restricted by both energy harvesting and co-channel interference, we employ the following studies. First, we take the link scheduling problem under a more realistic network model, the request-to-send (RTS)/clear-to-send (CTS) model, under which the communication graph can be an arbitrary geometric graph. We also use a practical energy-harvesting model, which profiles a large class of ambient energy sources outputting very low energy, such as vibration, thermo, and corrosion. Based on these realistic models, we transform the link scheduling problem to a vertex-coloring problem, using the concept of conflict graph which profile interferences in a co-channel wireless network. Second, we design two effective, low-complexity algorithms for link scheduling in two cases - with and without considering traffic flows. Specially, we provide theoretical bounds for our algorithms in these two cases. Finally, we evaluate our algorithms via primary simulation experiments.

Related work

Rechargeable wireless networks, especially rechargeable sensor networks, have attracted more attention recently. The node uses an energy harvester to collect energy from its surroundings. More effort is put on the design of energy-recharging circuit which can be integrated to customer-off-the-shelf sensor mote, such like Crossbow Mica-family mote and Berkeley Telos mote. The early work is attributed to Kansal et al. [1] who design a solar-powered sensor node and an energy metric, EEHF, which assigns more loads at nodes with a higher energy-harvesting rate. Helimote [17] is an energy-harvesting system with a single storage for solar energy. Built on Mica2 mote, Helimote recharges two AA Ni-MH batteries. Helimote can learn its energy availability and usage via an energy monitoring component. Jiang et al. [18] design Prometheus, a hybrid energy storage system for solar energy. Built on Telos, Prometheus mote is powered by the supercapacitors, called the primary energy buffer, or by the rechargeable Li-ion battery, called the secondary energy buffer. If the primary buffer energy is less than some threshold, the mote falls back to the secondary buffer until the primary one recharges fully again. Similar to Prometheus, AmbiMax [19] uses hybrid energy storage, but AmbiMax tracks the maximum power point automatically, without needing the control of MCU. EverLast [20] is a supercapacitor-driven sensor mote, and it uses PFM controller and regulator to harvest the solar energy. To track the maximum power point, EverLast is built with a complex charging circuit. Qiao et al. [21] discuss the feasibility of powering wireless nodes with the energy from corrosion process.

Besides the platform designs, there are recently some studies focusing on the routing algorithms in rechargeable wireless network. Lin et al. [22] developed a routing scheme to optimize the network throughput by an energy model assuming heterogeneous energy-harvesting rates. CHESS [23] is a routing metric for energy-harvesting sensor networks. It uses hybrid energy storage and assigns different costs to the energy in supercapacitor and the rechargeable batteries and favors routes with more energy. Eu et al. [24] studied optimal routing in an energy-harvesting sensor network with optimal relay node placement and investigated the impact of routing and node placement on the network performance. In [25], they proposed a lexicographically maximal data rate assignment for each source node such that no node runs out of energy. In [26], two schemes, QuickFix and SnapIt, were proposed. QuickFix computes the static routes using the sub-gradient method, and based on QuickFix, SnapIt adaptively allocates sampling rate for a single node with a possibly dynamic recharging rate. In [27], the authors investigated the finite-horizon problem in rechargeable sensor networks and proposed an approach including three steps: first, optimize the throughput of a single node; second, design an online algorithm, assuming that the future energy-harvesting rate is predictable; and third, to improve complexity, propose a heuristic distributed algorithm which is optimal with homogeneous energy-harvesting rate of nodes.

As a kind of resource allocation to optimize network throughput, link scheduling has been studied extensively in the past few years. Essentially, link scheduling can be reduced to graph coloring problems. An early work [28] has a vertex color graph G with $\delta + 1$ colors, where δ is the minimum degree in G. A unified framework for TDMA-, FDMA-, and CDMA-based multi-hop wireless network is proposed in [29], in which the time slot assignment for all edges is at most $\mathcal{O}(\theta)$ times the best solution, where θ evaluates the degree that the graph can be decomposed. In [30], a $\mathcal{O}(\Delta + 1)$-coloring is designed with time $\mathcal{O}(\log n + \Delta)$, where Δ is the maximum degree of G. However, those algorithms do not consider the interference of practical wireless application and then cannot work directly in wireless networks.

Gandham et al. [13] designed a two-phase approach which first computes a valid edge coloring and then assigns directions for all edges to avoid interference. The performance is guaranteed only with tree topology. The coloring in [31] is a $\mathcal{O}(\Delta \log n)$ distributed algorithm using Δ colors, but in the unit disk graph (UDG) model, they may make interfering links still use the same color. Kodialam et al. [32] considered jointly routing and link scheduling with the protocol interference model and solved the scheduling problem as an edge-coloring problem. They also extended the algorithm to the multi-radio and multi-channel scenario [33]. Alicherry et al. [9] established necessary and sufficient conditions for interference-free link schedule and proposed a simple greedy algorithm based on the UDG model. Sharma et al. [34] defined a K-hop interference model under which no two links within K hops can be scheduled simultaneously and presented a centralized greedy matching algorithm with a constant factor. The same authors studied greedy maximal scheduling with the K-hop interference model [35]. Wan et al. [36] studied the link scheduling with a physical interference model.They also presented a link scheduling algorithm without requiring node's location information [15]. Koulali et al. [37] designed a centralized power control scheme for rechargeable wireless multimedia sensor networks.

Models and assumptions

In this section, we will present the wireless communication models and some assumptions used in our study and will describe the link scheduling problem formally.

Network and interference models

We assume that there is a set V of nodes deployed in a plane, each of which is location-fixed and equipped with the identical radio. The wireless network then forms a

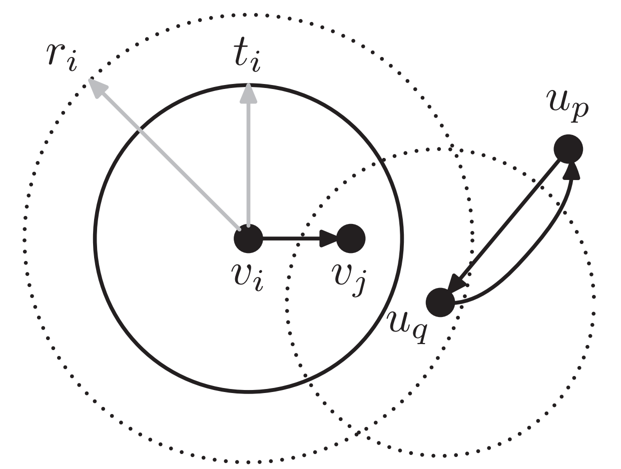

Figure 3 The primary and secondary interferences in link scheduling.

communication graph $G = (V, E)$, where E is the set of all radio links among nodes. Denote link (v_i, v_j) by $\ell_{i,j}$. The transmission range of each node v_i of V is r_i. It is necessary for node v_j to correctly receive data from v_i that the Euclidean distance between v_i and v_j, $||v_i - v_j||$, is not larger than r_i. Note here that $||v_i - v_j|| \leq r_i$ is not the sufficient condition for $(v_i, v_j) \in E$ because some links may be cut off by obstacles or forbidden by topology controls as well as routing policies. Each node v_i has an interference range, which is generally equal to or greater than r_i. For the ease of analysis, the interference range is hereafter assumed to be the same as the transmission range. But we use a more realistic interference radius setting in simulation experiments. We assume that the operation time of all nodes is equally slotted and synchronized and that the size of all packets are the same. In general, the link scheduling is to assign every link a set of time slots. Let τ denote the time span of a slot and assume that in a slot, a node can either transmit or receive only one packet, except for necessary control bits.

In a single-radio network, interference between adjacent or close-by links has to be avoided in order to reduce the delay or save energy. There are two major kinds of interferences: primary interference and secondary interference. When a node transmits and receives at the same time slot, primary interference happens; when a node receive two or more distinct transmissions, secondary interference occurs. In Figure 3, links $\ell_{p,q}$ and $\ell_{q,p}$ cannot be scheduled concurrently for a single-radio network. Additionally, since node u_q locates within the interference range of v_i, links $\ell_{i,j}$ and $\ell_{p,q}$ cannot be scheduled at the same time to avoid conflict at v_j. Besides, in real scenarios, for improving the network reliability or other network performance, there are other kinds of interference that pose restrictions on link scheduling, such as the RTS and CTS control schemes used in 802.11-like MAC protocols.

In this study, we use the RTS/CTS-based interference model [9,38-40], which is more realistic than the UDG model extensively used in previous literature. Under RTS/CTS model, all nodes within the interference range of either a transmitter or a receiver are not allowed to communicate. In detail, if two links $\ell_{i,j}$ and $\ell_{p,q}$ can be active simultaneously, it must be satisfied that v_i, v_j, v_p, and v_q are four distinct nodes, and v_i as well as v_j is not covered by the interference regions of v_p and v_q, and vice versa. In Figure 4a, the RTS message from v_j will be listened by u_p, so u_p must postpone its transmission to u_q. In Figure 4b, if v_j admits to receive data from v_i, v_j will send a CTS message, which will also be received by u_p, then $\ell_{p,q}$ should not be scheduled while $\ell_{i,j}$ is active. Clearly, in comparison with the UDG model, the RTS/CTS model neither requires that all nodes have the same radio range nor assumes that two nodes within each other's radio range must be able to communicate. Therefore, the RTS/CTS model does not preclude the

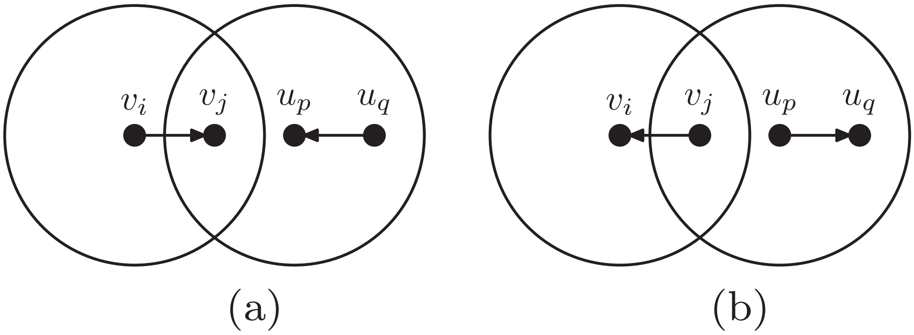

Figure 4 Constraints on link scheduling under RTS/CTS model. (a) The RTS message from v_j will be listened by u_p, so u_p must postpone its transmission to u_q. **(b)** If v_j admits to receive data from v_i, v_j will send a CTS message, which will also be received by u_p, then $\ell_{p,q}$ should not be scheduled while $\ell_{i,j}$ is active.

topology or transmit power control algorithms, thereby being more realistic and layer-independent for real scenarios.

Energy harvesting model

In practice, the rechargeable wireless node is equipped with a harvester unit which transforms ambient signals into trickles of electrical energy. The energy buffer of the rechargeable node is often very limited in capacity due to the constraints of package size and cost. In addition, for some scenarios where the environmental energy is extremely low in output power, nodes have to use longer periods of time to recharge their energy buffer before they can carry out tasks. Therefore, it is often impossible for a node to quickly buffer much energy for longtime, continual operation. Combined with the challenge of co-channel interference, the constrained and intermittent energy provision motivates us to investigate efficient link scheduling in the context of recharging wireless networks, such that the network throughput can be improved by carefully assigning time slots for links.

For simplicity, we only consider the energy consumption in communication because in low-rate, low-power wireless scenarios, communication consumes significant energy in comparison with computation. We also assume that transmitting and receiving data consumes identical power, e.g., for TI's CC2420 radio chip, the current consumption in transmitting and receiving are 17.4 and 19.4 mA, respectively.

Denote by ϵ the energy consumed by one-packet transmission or reception. In data communication with acknowledgement (ACK) scheme, ϵ represents the total energy used in transmitting a packet and receiving the ACK message. In the real world, the surrounding energy distributes unevenly because of different natural environments or deployments. Therefore, different nodes may have different energy-harvesting abilities - different recharging durations. Let γ be the duration (in time slot) for a node to harvest 1ϵ energy, and obviously, γ evaluates the energy-harvesting ability of that node. If $\gamma = 0$ for a node, then it means that this node has an always-available energy source. We assume that for a given node, its energy-harvesting duration is constant or varies in a very small range and is known *a priori*. This assumption profiles many typical environmental energy sources in practice, such as the vibration in factory [2], the thermo-heat of the human body in mobile health care [5], and the corrosion of steel-concrete structures [3], where all the energy generation rates vary little even in a long period, say, several weeks, or even months. In this paper, the rechargeable nodes work with two alternate states: active and dormant states [41-44]. In detail, when a node has enough energy, it will enter the active state to perform computation and communication; otherwise, it will enter the sleeping state while leaving only the harvesting module active.

Problem formulation

In a time-slotted wireless network, a link schedule is to find a time slot assignment for each link, and it is called *valid* if at any time slot, two active links are not conflicting. T is called the *schedule period* if it takes T time slots to schedule all the links once (without considering traffic load) or all the flows once (with considering traffic load). Link scheduling often aims at improving the network throughput, i.e., reducing T to be as small as possible while making the network conflict-free.

Besides the interference restriction, the link scheduling in rechargeable wireless network is restricted by the energy dynamics of node because one or two end nodes of a link may need more time to recharge until it (they) harvest sufficient energy to transmit or receive a packet. In this paper, therefore, the objective of link scheduling is to determine a time slot assignment for all links or flows with a smallest possible schedule period, such that not only interference can be avoided, but also each node has enough energy upon being scheduled.

For a given communication network, the link schedule is essentially a kind of edge-coloring problem in the communication graph. Different from the UDG model, however, our RTS/CTS model leads to a more realistic but more complex interference relationship among close-by links. To address this problem, we use the concept of the conflict graph [10,11,40] to transform link scheduling to a vertex-coloring problem as follows.

Let F_G be the conflict graph of a given communication graph G. Each edge (link) of G is converted to a vertex of F_G, and two vertices of F_G (both of them represent two links of G) are connected with an edge if they conflict with each other in G, according to the RTS/CTS model. Figure 5, for example, shows communication graph G and its conflict graph F_G. It is clear that we can obtain an interference-free link schedule for G by finding a proper vertex-coloring of F_G. To make the conflict graph able to evaluate the level and dynamics of the node's energy, we extend the above conflict graph model by adding a weight, denoted by $\varpi_{i,j}$, for each vertex of F_G, where $\varpi_{i,j} = \max\{w_i, w_j\}$ and w_i and w_j represent the earliest possible time for v_i and v_j to be scheduled, respectively. Obviously, w_i varies according to time and then is determined by γ_i, the harvesting duration of v_i, and by the current schedule of links nearby or adjacent to v_i. In Figure 5, $\gamma_1 = 2$ and $\gamma_2 = 3$. If the initial energy of both v_1 and v_2 is empty at time slot 0, we have $w_1 = 3$ and $w_2 = 4$, i.e., the earliest schedulable times for v_1 and v_2 are time slots 3 and 4, respectively, and consequently, $\varpi_{1,2}$ is equal to 4 according to the definition.

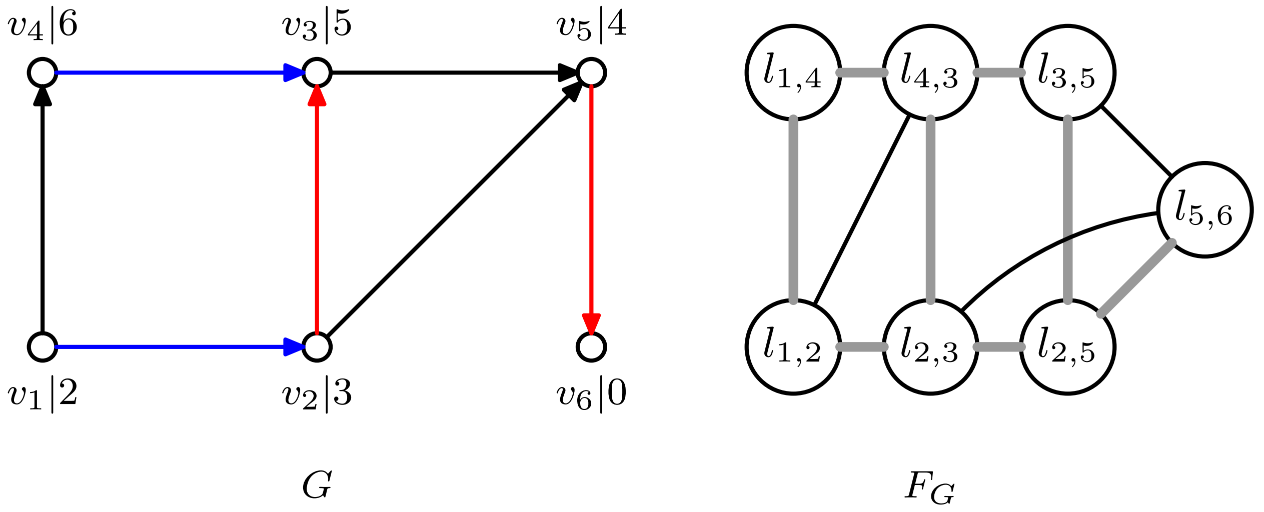

Figure 5 Converting a communication graph *G* to a conflict graph F_G. In G, $\ell_{1,2}$ interferes with $\ell_{4,3}$, and $\ell_{2,3}$ interferes with $\ell_{5,6}$. In F_G, the thick gray edge and the thin dark edge describe the primary interference and secondary interference, respectively.

Note that in the context of rechargeable wireless network, the concern on vertex-coloring F_G is not to achieve the smallest possible number of colors (time slots), much the way the previous link scheduling schemes assuming always-available energy source does, but to find a schedule for all links with the smallest possible span of time slots, which is the way to evaluate and improve the performance of link scheduling for energy-rechargeable wireless applications.

Algorithm design

In this section we will, in detail, present two link scheduling algorithms for rechargeable wireless network, with and without considering the traffic load, and theoretically analyze their complexity and prove their performance.

Scheduling without traffic load

We first introduce some necessary definitions and properties before presenting a centralized link scheduling algorithm and proving its performance. For each vertex u of G, besides its harvesting duration γ_u as well as its earliest schedulable time w_i, we also use two other variables to describe u's state at time t: e_u and s_u, where e_u represents the energy level of u and s_u the times that u has been already scheduled till time t. At a given time t, if we know γ_u and s_u, then we can calculate the current energy level of u by Equation 1. At time t, if $e_u > 1\epsilon$, u can be scheduled immediately, that is, $w_u = t$, and if $e_u = 1\epsilon$, we have $w_u = t+1$. If $e_u < \epsilon$ at time t, however, u needs more time to collect sufficient energy before it can be scheduled, and w_u can be calculated by Equation 2:

$$e_u = \frac{t}{\gamma_u} - s_u \times \epsilon \tag{1}$$

$$w_u = t + \gamma_u(\epsilon - e_u) + 1 \tag{2}$$

According to the calculation of w_u of each vertex u, we can easily determine the earliest schedulable time of a link. For two vertices (links) $\ell_{i,j}$ and $\ell_{p,q}$ of F_G, if $\varpi_{i,j}$ is less than $\varpi_{p,q}$ or if $\varpi_{i,j} = \varpi_{p,q}$ but the degree of $\ell_{i,j}$ is less than that of $\ell_{p,q}$, we say that link $\ell_{i,j}$ *proceeds* link $\ell_{p,q}$.

Among the centralized link scheduling algorithms previously presented, the common method is to use the greedy graph coloring to assign each vertex the minimum possible time slot. There is a critical difference among these works: they use different orders to greedily color vertices. Our centralized scheduling is also based on the greedy vertex coloring used in general graphs: the earliest available vertex of F_G will always be first colored with the smallest possible time slot (color), and of two links with identical earliest available time, the link with minimum degree will be scheduled preferentially. Algorithm 1 details the centralized link scheduling without considering traffic load. Construction of conflict graph needs time of $\mathcal{O}(m^2)$, where m is the number of links of G. It can be easily deduced that in Algorithm 1, every vertex of F_G is selected only once and that all unselected vertices adjacent to the currently selected vertex are scanned to update their earliest schedulable times. So the time complexity of Algorithm 1 is also $\mathcal{O}(m^2)$. Next, we will prove the performance of Algorithm 1.

Let R_x be the interference region of vertex x, a circle centering x and with radius of r_x. Let $N_x^{\geqslant}$ be the set of vertices whose interference radius is at least r_x and who interfere with some vertices covered by R_x. The interference radius of a link $\ell_{u,v}$ is defined as $r_{u,v}$ (or r_ℓ) which equals $\max\{r_u, r_v\}$. And denote by $I^{\geqslant}(\ell)$ the set of links who interfere with ℓ and whose interference radius is at least r_ℓ. Clearly $I^{\geqslant}(\ell)$ is an effective lower bound for the potential interferences to be experienced by vertex x.

Algorithm 1 Link scheduling without traffic load

```
Require: $G$, the communication graph
Ensure: a valid link schedule for all links of $G$
1: For each vertex $u$ of $G$, set $e_u = 0$, $s_u = 0$, and $w_u = \gamma_u + 1$. Construct $G$'s conflict graph, $F_G$, and let $F'_G = F_G$.
2: while $F'_G$ is not empty do
3:     Find vertex $\ell_{u,v}$ of $F'_G$ such that it proceeds all other vertices.
4:     If none of $\ell_{u,v}$'s neighbors in $F_G$ is already assigned to time slot $\varpi_{u,v}$, then assign time slot $\varpi_{u,v}$ to $\ell_{u,v}$; otherwise, assign $\ell_{u,v}$ the minimum possible time slot not yet used by any of its neighbors of $F_G$. Denote by $k$ the time slot for $\ell_{u,v}$.
5:     Increase $s_u$ and $s_v$ both by one. Update the energy level of $u$ by Equation 1. If $e_u \geq \epsilon$, then $w_u$ is set to be $k + 1$; otherwise, it is set to be the time slot calculated by Equation 2. Similarly, $e_v$ and $w_v$ can be updated also. Notice that updating $w_u$ or $w_v$ changes consequently the weights of other links incident with $u$ or $v$. Delete vertex $\ell_{u,v}$ from $F'_G$.
6: end while
```

Theorem 1. *Given a vertex v_x and any subset V_x of $N_x^{\geqslant}$, there exists a vertex subset V'_x of V_x with size at least $|V_x|/\varphi$ such that each vertex of V'_x interferes with or is interfered by at least one other vertices, where φ is upper-bounded by 30.*

Proof. See Appendix. □

Theorem 1 implicates that in $N_x^{\geqslant}$, a set of vertices that possibly interferes with the transmission from v_x or to v_x, there exists a subset of $N_x^{\geqslant}$, in which for any two vertices, u_i and u_j, u_i interferes with u_j, u_j interferes with u_i, or both. Even though Theorem 1 estimates only the degree of local interference at level of vertex, we can extend it to the level of link using Corollary 1.

Corollary 1. *For any given link ℓ, there exists a subset of $|I^{\geqslant}(\ell)|$ of size at least $|I^{\geqslant}(\ell)|/2\varphi$, in which any pair of links interferes with each other under the RTS/CTS model.*

Proof. See Appendix. □

In fact, Corollary 1 indicates that if a vertex (link) ℓ_0 of conflict graph F_G has k neighboring vertices (links), $\ell_1, \ell_2, \ldots \ell_k$, then ℓ_0 must have at least $k/2\varphi$ neighbors that are mutually interfering; in other words, the graph formed by these k vertices and the edges between them contains a clique of order $k/2\varphi$.

Theorem 2. *For a given communication graph G, if its conflict graph is a clique of order m, denoted by $F_G^{\mathcal{K}}$, then we need as least $(\gamma_{min} + m + 1)$ time slots for scheduling all links, where γ_{min} is the shortest harvesting duration of G's vertices and m is the number of links of G.*

Proof. This theorem can be proven by first considering an extreme case, in which for any two vertices (links) of $F_G^{\mathcal{K}}$, they are not incident on any vertex of G. As shown in Figure 6, for example, all five links in G are interfering and form clique $F_G^{\mathcal{K}}$ of order five, but they have no common vertices. In such a case, the energy levels at both end vertices of a given link put no effect on scheduling other links of G in terms of the energy restriction. The schedule period depends only on close-by interferences and the local harvesting rates. Suppose that the fastest possible schedule is T_K time slots in this case; obviously, T_K is at least $(\gamma_{\min} + m + 1)$.

If two links of G is incident in Figure 6, the conflict graph is still clique $F_G^{\mathcal{K}}$; however, coloring these two adjacent vertices (links) of $F_G^{\mathcal{K}}$ will have to be affected (restricted) by the energy replenishment of the incident vertex. Thus, the schedule period must be greater than or equal to $T_K = \gamma_{min}+m+1$, where m is the order of the conflict graph. □

Because what we are concerned about is the span of time slots for all links, it is reasonable to treat the maximum assigned time slot as the schedule period. Theorem 2 helps in determining an effective lower bound for an arbitrary wireless network with constraints of co-channel interference and energy replenishment. Based on Theorem 2, the following theorem will prove that Algorithm 1 can achieve a constant factor of the optimal link scheduling.

Theorem 3. *Algorithm 1 computes a valid time slot assignment using at most $2\varphi(\gamma_{max} + 1)$ times the time slots needed by the optimal schedule, where γ_{max} is the longest harvesting duration of vertex in G and φ is the upper bound deduced in Theorem 1.*

Proof. Note that in step four of this algorithm, we always assign a vertex which is the time slot that has not yet been used by any of its adjacent vertices of F_G. Thus, it is easy to know that Algorithm 1 is conflict-free. Next, we prove the algorithm's performance.

Let $\mathcal{T}_{\mathrm{opt}}$ and $\mathcal{T}_{\mathrm{Alg1}}$ be the scheduling period's output by the optimal algorithm and Algorithm 1, respectively. Suppose that v is the vertex with the maximum degree in F_G and its degree is denoted by δ_v. Corollary 1 implies that for v, there exists a clique (including v and its neighbors), whose size is at least $(\delta_v + 1)/2\varphi$. Then, by Theorem 2, we need at least $(\gamma_{\min} + (\delta_v + 1)/2\varphi + 1)$ time slots for scheduling all vertices (links) of the clique, without considering other vertices of F_G. So we have an effective lower bound for the optimal algorithm's performance, that is, $\mathcal{T}_{\mathrm{opt}} \geq (\gamma_{\min} + (\delta_v + 1)/2\varphi + 1)$.

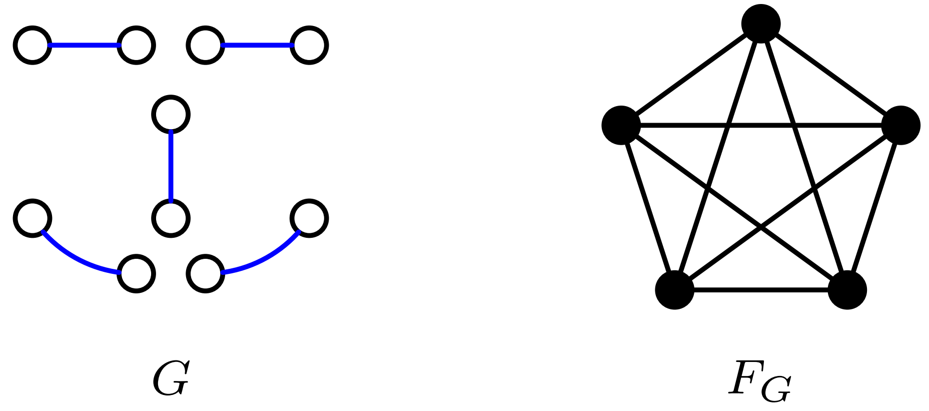

Figure 6 Illustration for the proof of Theorem 2. In G whose conflict graph F_G^K is of form clique, all links interfere mutually but are not incident with one another.

It is not straightforward to derive a tight upper bound of $\mathcal{T}_{\text{Alg1}}$. Here, we use some theories about T-coloring problem [45-47] for the proof. We construct a T-coloring-based schedule for G, whose period is longer than that of Algorithm 1 but is easy to be upper-bounded. Given a graph, a T-coloring is to assign each vertex an integer such that the absolute value of the difference between integers for adjacent vertices does not belong to T which is called T set. In our scenario, for example, if the T set is $\{0, 4\}$ and a vertex of F_G is the given color (time slot) c, then its neighbor (links) can be given neither color c nor color $c + 4$.

Now let T set be $\{0, 1, 2, \ldots, \gamma_{\max}\}$. In the schedule based on this T set, any two adjacent vertices (links) of F_G are sure to be conflict-free because $0 \in T$, meaning that two adjacent vertices will not be assigned the same time slot. In fact, for any two adjacent vertices of F_G, the difference of time slot assigned to them is at least $\gamma_{\max} + 1$, which is according to the definition of the T set, and therefore, such a T set results in a T-coloring schedule that is feasible for rechargeable wireless network, in terms of interference and energy. It is clear that the T-coloring-based link scheduling always waits at least $(\gamma_{\max} + 1)$ time slots between two consecutive schedules of two interfering links, without considering whether or not the second link is already available before being scheduled. In contrast, Algorithm 1, based on the maximal scheduling policy, always schedules a link the moment it is feasible in terms of both energy and interference and consequently uses less time than the above T-coloring counterpart does. Obviously, the performance of T-coloring-based scheduling provides a natural upper bound for Algorithm 1.

It is given in [45,46] that T-coloring graph G, the minimum span over all T-colors for G, denoted by $\text{sp}_T(G)$, can be expressed by $(r+1)(\chi(G)-1)$, where $\chi(G)$ is the chromatic number of G (without being restricted by the T set) and r is the maximum integer of the T set. Recall that we have assumed that the maximum degree of F_G is δ_v; we know that the chromatic number $\chi(F_G)$ must be less than or equal to $(\delta_v + 1)$, according to the principle of graph theory. Furthermore, we have $\text{sp}_T(G) \leq r\delta_v$. Thus, Algorithm 1 uses at most $(\gamma_{\max}+1)\delta_v$ time slots for scheduling all links once, that is, $\mathcal{T}_{\text{Alg1}} \leq (\gamma_{\max} + 1)\delta_v$:

$$\begin{aligned}\frac{\mathcal{T}_{\text{Alg1}}}{\mathcal{T}_{\text{opt}}} &\leq \frac{(\gamma_{\max} + 1)\delta_v}{\gamma_{\min} + 1 + (\delta_v + 1)/2\varphi} \\ &< \frac{(\gamma_{\max} + 1)(\delta_v + 1)}{(\delta_v + 1)/2\varphi} \\ &= 2\varphi(\gamma_{\max} + 1)\end{aligned} \tag{3}$$

According to the lower bound for $\mathcal{T}_{\text{opt}}$ and the upper bound for $\mathcal{T}_{\text{Alg1}}$, we have Equation 3, that is, the schedule period of Algorithm 1 is at most $2\varphi(\gamma_{\max} + 1)$ times that of the optimal solution. □

Note that in the proof of Theorem 3, we assume that the vertex with the maximum degree in F_G has the worst harvesting duration. Thus, the upper bound for the performance of Algorithm 1 can be improved with a tighter analysis if we have further knowledge about the distribution of harvesting rates among nodes.

Scheduling with traffic load

The algorithm proposed above is based on the assumption that all links are needed to deliver equal traffic load. But not all communication patterns support this assumption, especially the many-to-one wireless sensor networking that uses a few links to collect data from all source nodes. In this section, we will design a more general scheduling framework considering traffic load of link and prove its performance based on the previous analysis.

Now consider a communication graph G. For source vertex s_i and terminal vertex t_i, there are $k(k \geq 1)$

flows from s_i to t_i that are determined by some routing algorithm. For each link ℓ of G, its traffic load can be measured by $\sum f_i(\ell)$, where $f_i(\ell)$ means the ith flow traversing through link ℓ. Exactly, if the bandwidth of ℓ is b_ℓ, then ℓ needs to be assigned $\lceil \frac{1}{b_\ell} \sum f_i(\ell) \rceil$ time slots in a single schedule of all flows, which is the link traffic load of ℓ in terms of time slot. In practice, the traffic load through each link can be determined once the routing algorithm terminates correctly, and in general, a link is possibly scheduled multiple times - assigned with multiple time slots - for scheduling all flows.

```
Algorithm 2 Link scheduling with traffic load
Require: $G$, the communication graph, and the traffic
    load $\rho_\ell$ of each link $\ell$
Ensure: a valid link schedule for all flows of $G$
 1: Construct $G$'s conflict graph, $F_G$. As shown in
    Figure 7, add vertices and edges to $F'_G$, according to
    the following steps (from 2 to 4).
 2: Declare an empty graph $F'_G$
 3: For a given vertex (link) $\ell$ of $F_G$, if $\rho_\ell > 0$, then we add
    to $F'_G$ with $\rho_\ell$ identical vertices, $\ell^1, \ell^2 \ldots, \ell^\rho$, and with
    the edges connecting $\ell^i$ and $\ell^j$ of $F'_G$ $(1 \le i < j \le \rho)$.
 4: For any two adjacent vertices $\ell_1$ and $\ell_2$ of $F_G$, add to
    $F'_G$ the edges connecting $\ell_1^i$ and $\ell_2^j$, where $1 \le i \le \rho_1$
    and $1 \le j \le \rho_2$.
 5: Run steps 2 to 6 of Algorithm 1 directly on $F'_G$, which
    outputs a time slot for each $\ell_i^j$ of $F'_G$.
 6: Assign link $\ell$ of $G$ all the time slots used by $\ell^i (1 \le i \le
    \rho_\ell)$ of $F'_G$.
```

To schedule all flows, Algorithm 2 is proposed to assign each link one or more time slots according to the link load. The basic idea of Algorithm 2 is to extend the conflict graph used by Algorithm 1, such that the vertex needing multiple schedules are decomposed to equal-quantity vertices, each of which only needs to be scheduled once using Algorithm 1. Figure 7 shows a walk-through example. The communication graph G will deliver two flows: one is from v_1 to v_4 and the other is from v_2 to v_4. Both flows traverse past v_3, so link $\ell_{1,3}$ takes the load of one unit, $\ell_{2,3}$ takes the load of one unit, and $\ell_{3,4}$ takes the load of two units. The middle graph in Figure 7, F_G, is the conflict graph of G without considering the traffic load. To schedule two flows, however, we need to assign three links $\ell_{1,3}$, $\ell_{2,3}$, and $\ell_{3,4}$: one time slot, one time slot, and two time slots, respectively. We rename $\ell_{3,4}$ using $\ell_{3,4}^1$ and add into F_G a new vertex, denoted by $\ell_{3,4}^2$, such that $\ell_{3,4}^2$ relates with other vertices as $\ell_{3,4}$ does. In other words, we connect $\ell_{3,4}^2$ and ℓ if there is an edge between $\ell_{3,4}^1$ and ℓ in F_G. Of course we should also add an edge between $\ell_{3,4}^1$ and $\ell_{3,4}^2$. In doing so, the conflict graph F_G is extended to F'_G, shown on the right side of Figure 7. Clearly, F'_G not only demonstrates the restriction of interference in link scheduling, as F_G does, but also indicates the number of times for scheduling each link. Now we can complete scheduling all flows of G as soon as each vertex (link) of F'_G is scheduled once.

From the process of extending F_G into F'_G, we can see that the complexity of such extension depends on flows over communication network G, i.e., on the weight (traffic load) of each link. For a wireless network with n nodes, the traffic load through any node is of order $\mathcal{O}(n)$. Step 3 of Algorithm 2 examines each vertex to verify whether its weight is greater than one, consuming $\mathcal{O}(m)$ time where m is the number of vertices of F_G (or the number of links of G). Combining steps 3 and 4 of Algorithm 2, the extension from F_G to F'_G requires $\mathcal{O}(mn)$ time. Since the construction of F_G is of time complexity $\mathcal{O}(m^2)$, as analyzed in the previous section, Algorithm 2 terminates with time of order $\mathcal{O}(\max\{m^2, mn\})$. We next prove the validity and performance of Algorithm 2 via Theorem 4.

Theorem 4. *For a given communication graph, Algorithm 2 returns a valid link schedule for all flows, which uses at most k times that of the optimal solution considering the traffic load if Algorithm 1 uses at most k times the optimal solution without considering the traffic load.*

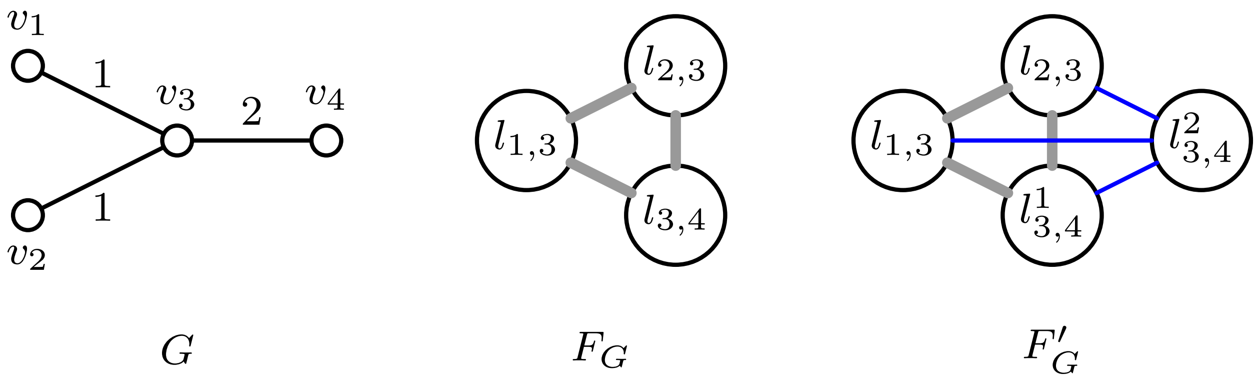

Figure 7 Illustration for Algorithm 2. The integer associated with each link of *G* is the traffic load of that link.

Proof. For a given communication graph G and a set of flows through G, suppose that the traffic load of each link of G is at least one. Let $\mathcal{T}^f_{\mathrm{OPT}}$ be the minimum span of time slots for scheduling all vertices of F_G, considering link traffic, and let $\mathcal{T}_{\mathrm{OPT}}$ be the minimum span of time slots for scheduling all vertices of F'_G, without considering link traffic. If $\mathcal{T}^f_{\mathrm{OPT}} > \mathcal{T}_{\mathrm{OPT}}$, then we can use the optimal schedule on F'_G to construct a new schedule for F_G with a smaller period. So $\mathcal{T}^f_{\mathrm{OPT}}$ is not the best solution; it contradicts. Similarly, we know that $\mathcal{T}^f_{\mathrm{OPT}} < \mathcal{T}_{\mathrm{OPT}}$ is impossible too. Thus, we have $\mathcal{T}^f_{\mathrm{OPT}} = \mathcal{T}_{\mathrm{OPT}}$ for a given G and any flow assignment through G.

Essentially, the operation of Algorithm 2 is equivalent to running Algorithm 1 on the extended conflict graph, F'_G. If the schedule period for F'_G output by Algorithm 1 is k times $\mathcal{T}_{\mathrm{OPT}}$, then it is also k times $\mathcal{T}^f_{\mathrm{OPT}}$. Note that it is possible for a routing to forbid some links, which makes them not to be scheduled at all; however, this theorem still holds in such a case. □

Performance evaluation

In this section, we conduct extensive simulations to evaluate our algorithms and compare their performance with the lower bound of the optimal solution. In the case without traffic flows, the lower bound on the optimum is set to be $\max\{\gamma_i \cdot \delta_i\} + 1$, where δ_i is the degree of vertex v_i in the communication G, and in the case with flows, the lower bound on the optimum is set to be $\max\{\gamma_i \sum f_i\}$, where f_i is the total flows past v_i. Clearly, the smaller the ratio to the lower bound for optimum, the better is the performance of the proposed algorithms.

In simulation, nodes are randomly deployed in a uniform way within a 40 × 40 square. The transmit range of node is randomly selected from 10 to 15 and fixed in simulation. For each node, its interference range is set to be twice its transmit range, much the way the IEEE 802.11 scheme works. The time span of a single time slot, τ, is set to be 500 ms which should be able to cover some control messages (RTS, CTS, and ACK), even though we neglect the control message exchange in our simulation. Every node randomly selects an energy-harvesting duration from the integer set of $\{0, 1, \ldots, \gamma_{\max}\}$. Each simulation result is reported with the average of 100 random deployments.

First, we examine the performance of Algorithm 1, which does not consider the link traffic and assumes that each link has the same data arrival rate. Figure 8 shows that given a $\gamma_{\max}$, when the number of nodes increases, more time is needed as expected. Why does the case with the smallest $\gamma_{\max}$ achieve a schedule period similar to the case with the largest $\gamma_{\max}$? Figure 9 gives an answer. From Figure 9, we can see that for a network size, the performance of Algorithm 1 becomes better as $\gamma_{\max}$ increases. This is because the larger the $\gamma_{\max}$ is, the larger the deviation of node's recharging cycles, and then the earliest schedulable times of distinct nodes will not overlap too much, as the case with what the smaller $\gamma_{\max}$ will experience. Additionally, given a $\gamma_{\max}$, increasing the network size deteriorates the performance due to more interference. Thus, as expected, Algorithm 1 may behave better with less interference, that is, it is more suitable for scenarios with sparse distribution or short radio range.

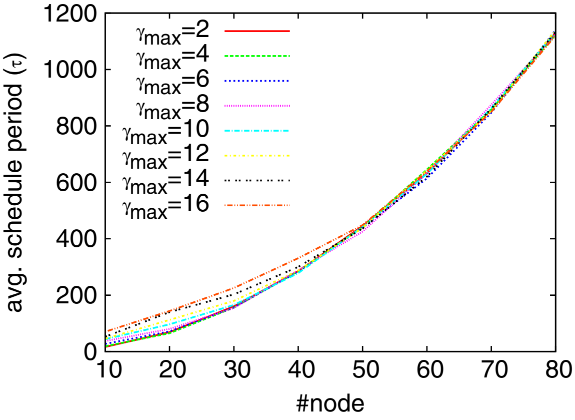

Figure 8 The schedule period under the link schedule without traffic flows.

Second, we evaluate Algorithm 2 which runs with the shortest path routing. Here, the shortest path routing algorithm, using the harvesting ability of nodes as the forwarding metric, always returns a path for each source such that the total harvesting durations of nodes along the path reaches the minimum. Note that Algorithm 2 depends neither on any particular routings nor on the communication patterns (one-to-one, many-to-one, or others). In

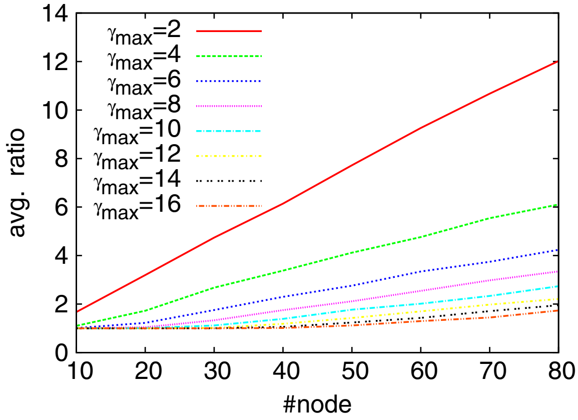

Figure 9 The performance ratio under the link schedule without traffic flows.

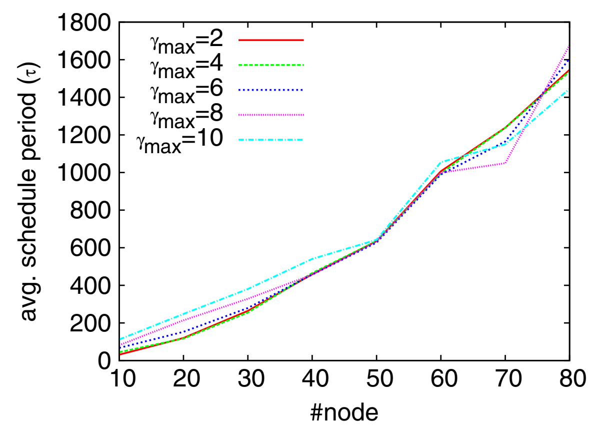

Figure 10 The schedule period under the link schedule with traffic flows.

every simulation, we always randomly select 20% nodes as sources, each of which uses a constant rate of one packet/s to report data to the sink node centering the square. Figures 10 and 11 show results that are similar to the case without considering traffic flows, as proven by Theorem 4.

Conclusions

In wireless ad hoc and sensor networks with renewable energy, not only the co-channel interference but also the energy replenishment weighs heavily against increasing the network throughput. So an efficient link scheduling is highly demanded in rechargeable wireless networks, especially in scenarios with very low ability in harvesting ambient energy. To the best of our knowledge, we are the first to investigate how to efficiently schedule links restricted by both interference and energy replenishment. Based on realistic network models, we first proposed a centralized scheduling for all links without considering the link load, and the achieved period of feasible schedule is within constant ratio of the best solution. We then proposed a more general link scheduling algorithm for network flows, with the proven performance of a constant factor. The simulation results also demonstrate the performance of our designs. In this paper, we did not analyze the stability of Algorithm 2 with an arbitrary pattern of network flows. Additionally, it is very difficult and challenging to design a localized link scheduling with a constant factor for a rechargeable wireless network because local decision may dramatically influence future performance in a hard predictable way. In the future, we will focus on these two aspects.

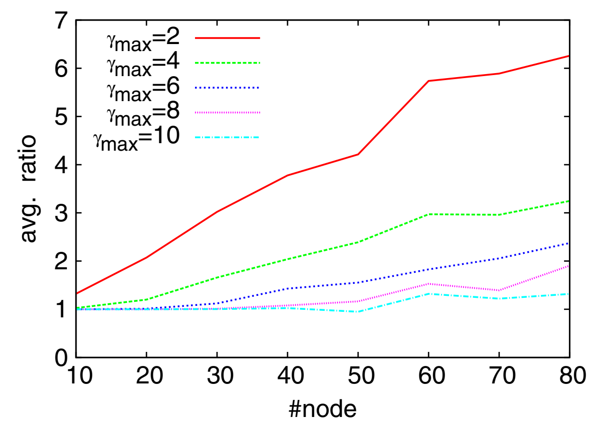

Figure 11 The performance ratio under the link schedule with traffic flows.

Appendix

Proof of Theorem 1. Given a vertex v_x and a subset V_x of $N_x^{\geqslant}$, suppose that v_x's interference radius and region are r_x and R_x, respectively. We divide V_x into three parts, according to the locations of its vertices: V_x^1 whose vertices are within R_x, V_x^2 whose vertices are out of R_x but covered by the circle centering v_x and with the radius of $\sqrt{3}r_x$, and V_x^3 (the other vertices of V_x). In this proof, we only need to prove that for any two vertices u and v of V_x', the distance $d_{u,v}$ is at most $\max\{r_u, r_v\}$. The basic idea of this proof is similar to the work of [40], but we improve it with a tighter upper bound for φ.

We first consider vertices of V_x^1 in region R_1, by slicing R_1 into six equally shape pies, like pie I in Figure 12. It is clear that for any two vertices in pie I, the distance between them is at most r_x. Recall that the interference radius of any vertex of $N_x^{\geqslant}$ is greater than or equal to r_x, and so the distance is not greater than the interference radius of each vertices. By the pigeonhole principle, there must be a pie in R_1, which covers at least $|V_x^1|/6$ vertices of V_x^1.

We can use 12 rays from v_x to slice ring R_2 and R_3 into 12 subregions of shape region II, and 12 subregions of shape region III, respectively. For any two points in region II, the distance is also at most r_x, and then it is less than the maximal one of the two vertices' interference radii.

We next consider the vertices of V_x^3 in R_3, the whole plane outside $R_1 \cup R_2$. For any two vertices v_p and v_q in R_3, suppose $d_{p,x} \geq d_{q,x}$, without losing generality; suppose v_p and v_q locate at points c and b of Figure 12, respectively. It is easy to prove that $||c-b|| < ||c-a||$ according to plane geometry principles (here we omit the proof details). Moreover, we know $||c-a||$ is always $\leq r_p$ (r_p is the interference radius of v_p) because v_p interferes some vertices in R_1 (i.e., $v_p \in N_x^{\geqslant}$). Hence, we know that $||c-b||$, the maximal possible distance between any pair of vertices in region III, is $< r_p$.

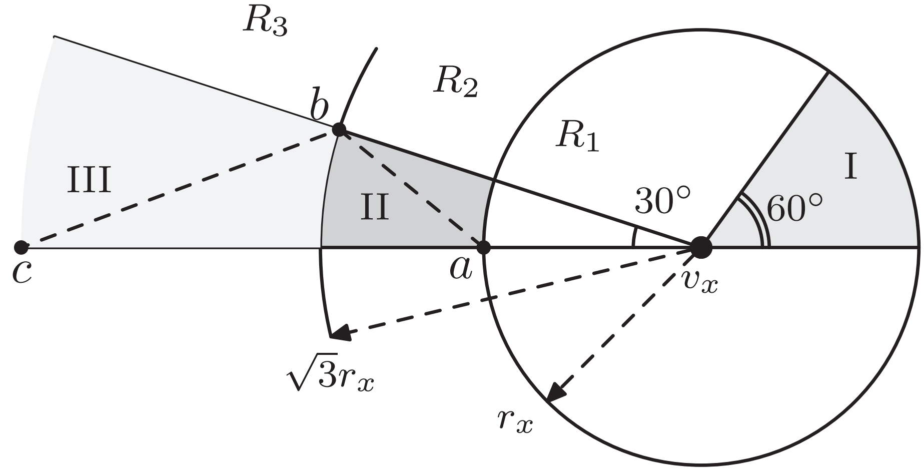

Figure 12 Illustration for the proof of Theorem 1.

We can now conclude that there exists a vertex subset V'_x of size at least $\max\{|V^1_x|/6, |V^2_x|/12, |V^3_x|/12\} \geq (|V^1_x| + |V^2_x| + |V^3_x|)/(6 + 12 + 12) = |V_x|/30$, such that each vertex of V'_x interferes with or is interfered by another vertex. □

Proof of Corollary 1. For two links $\ell_{i,j}$ and $\ell_{p,q}$, suppose $\ell_{p,q} \in I^{\geqslant}(\ell_{i,j})$. Similar to [40], we first consider the case in which $\ell_{i,j}$ and $\ell_{p,q}$ are incident on a vertex, say, $v_j = v_q$. It is clear that v_p interferes with v_i, and then it belongs to $N^{\geqslant}_i$. We next consider the case in which both links are not incident. As shown in Figure 13, there are three cases for the relationship between $\ell_{i,j}$ and $\ell_{p,q}$.

Case 1. Two end vertices of $\ell_{p,q}$ are covered by $R_i \cup R_j$, e.g., v_p and v_q.
Case 2. Only one end vertex is covered by $R_i \cup R_j$, e.g., $v_{p'}$.
Case 3. Both end vertices are outside $R_i \cup R_j$, e.g., $v_{p''}$ and $v_{q''}$.

In Case 1, at least one of v_p and v_q interferes with either v_i or v_j because $r_{p,q} \geq r_{i,j}$, according to the definition of $I^{\geqslant}(\ell_{i,j})$.

In Case 2, $v_{q'}$ interferes $v_{p'}$ that is covered by $R_i \cup R_j$, so $v_{q'}$ must belong to $N^{\geqslant}_i$, $N^{\geqslant}_j$, or both.

In Case 3, it is sure that either $R_{p''}$ or $R_{q''}$ covers at least one of v_i and v_j because $\ell_{p'',q''}$ has already been assumed to belong to $I^{\geqslant}(\ell_{i,j})$.

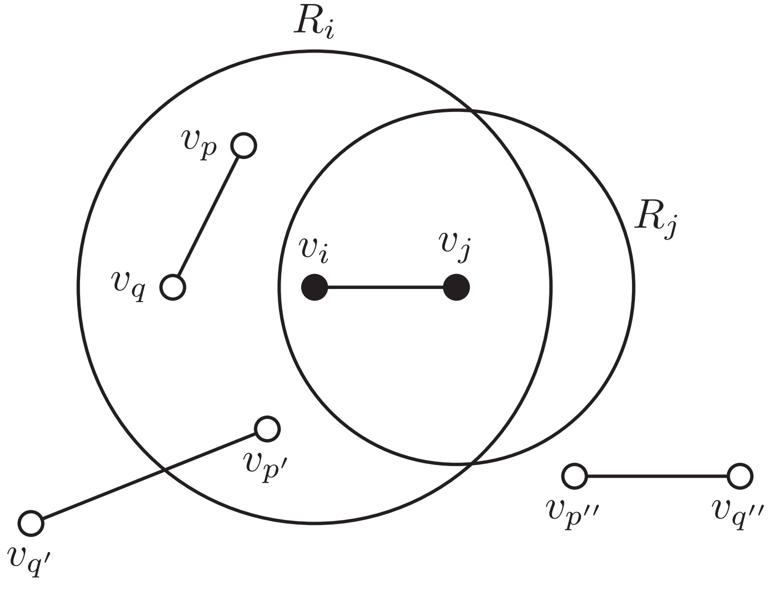

Figure 13 Illustration for the proof of Corollary 1.

By analyzing the above three cases, we can conclude that for any link $\ell_{p,q}$ of $I^{\geqslant}(\ell_{i,j})$, at least one of v_p and v_q covers some vertex (vertices) of $R_i \cup R_j$, regardless of whether $\ell_{p,q}$ and $\ell_{i,j}$ are incident or not. In other words, at least $|I^{\geqslant}(\ell_{i,j})|/2$ vertices belong to $N_i^{\geqslant}$ or $N_j^{\geqslant}$. By Theorem 1, we know that for any link $\ell_{i,j}$, there are at least $|I^{\geqslant}(\ell_{i,j})|/(2\varphi)$ vertices such that they are interfering mutually; thus, according to the restriction posed by RTS/CTS model, the links incident to these vertices must interfere with each other.

Competing interests

The authors declare that they have no competing interests.

Acknowledgements

Guodong Sun and Lin Zhao were supported by the Fundamental Research Funds for the Central Universities of China (grant no. BLX2012048), and Guofo Qiao was supported, in part, by NSF of China (grant no. 51008098).

Author details

[1]School of Information Science and Technology, Beijing Forestry University, 35 Tsinghua East Rd., Beijing 100083, China. [2]School of Civil Engineering, Harbin Institute of Technology, 77 Huanghe Rd, Harbin 150006, China.

Received: 5 May 2013 Accepted: 24 August 2013
Published: 5 September 2013

References

1. A Kansal, M Srivastava. An environment energy harvesting framework for sensor networks, in *ISLPED, Proceedings of the 2003 International Symposium on Low Power Electronics and Design, JW Marriott Hotel, Seoul, 25–27 Aug 2003* (ACM New York, 2003), pp. 481–486
2. H Kim, Y Tadesse, S Priya, in *Peizoelectric Energy Harvesting,* ed. by S Priya, DJ Inman. (Springer, New York, 2009), pp. 3–40
3. G Qiao, J Ou, Corrosion monitoring of reinforcing steel in cement morta by EIS and ENA. Electrochimica Acta. **52**, 8008–8019 (2007)
4. L Tang, C Guy. Radio frequency energy harvesting in wireless sensor networks, in *IWCMC. Proceedings of the 2009 International Conference on Wireless Communications and Mobile Computing: Connecting the World Wirelessly, Leipzig, 21–24 June 2009* (ACM New York, 2009), pp. 644–649
5. G Snyder, ed. by S Priya, DJ Inman. Thermoelectric energy harvesting, in *Energy Harvesting Technologies* (Springer New York, 2009), pp. 325–336
6. Z Eu, H Tan, W Seah. Wireless sensor networks powered by ambient energy harvesting: an empirical characterization, in *ICC* (Cape Town, 23-27 May 2010)
7. H Tamura, K Watanabe, M Sengoku, S Shinoda, On a new edge coloring related to multi-hop wireless networks. APCCAS. **2**, 357–360 (2002)
8. R Cruz, A Santhanam, Optimal routing, link scheduling and power control in multi-hop wireless networks. INFOCOM. **1**, 702–711 (2003)
9. M Alicherry, R Bhatia, L Li. Joint channel assignment and routing for throughput optimization in multi-radio wireless mesh networks, in *MobiCom. Proceedings of the 11th Annual International Conference on Mobile Computing and Networking, Cologne, 28 Aug–2 Sept 2005* (ACM New York, 2005), pp. 58–72
10. K Jain, J Padhye, V Padmanabhan, L Qiu, Impact of interference on multi-hop wireless network performance. Wireless Netw. **11**, 471–487 (2005)
11. G Brar, D Blough, P Santi. Computationally efficient scheduling with the physical interference model for throughput improvement in wireless mesh networks, in *MobiCom* (Los Angeles, 24–29 Sept 2006)
12. K Ramachandran, E Belding, K Almeroth, M Buddhikot. Interference-aware channel assignment in multi-radio wireless mesh networks, in *INFOCOM 2006. Proceedings of 25th IEEE International Conference on Computer Communications, Barcelona, April 2006* (IEEE Piscataway, 2006), pp. 1–12
13. S Gandham, M Dawande, R Prakash, Link scheduling in wireless sensor networks: distributed edge-coloring revisited. J. Parallel Distributed Comput. **68**, 1122–1134 (2008)
14. C Joo. A local greedy scheduling scheme with provable performance guarantee, in *MobiHoc. Proceedings of the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing, Hong Kong, 27–30 May 2008* (ACM New York, 2008), pp. 111–120
15. P Wan, Y Cheng, Z Wang, F Yao. Multiflows in multi-channel multi-radio multihop wireless networks, in *IEEE INFOCOM. The 30th IEEE International Conference on Computer Communications, Shanghai, 10–15 April 2011* (IEEE Piscataway, 2011), pp. 846–854
16. P Wan, C Ma, Z Wang, B Xu, M Li, X Jia. Weighted wireless link scheduling without information of positions and interference/communication radii, in *IEEE INFOCOM. The 30th IEEE International Conference on Computer Communications, Shanghai, 10–15 April 2011* (IEEE Piscataway, 2011), pp. 2327–2335
17. V Raghunathan, A Kansal, J Hsu, J Friedman, M Srivastava. Design considerations for solar energy harvesting wireless embedded systems, in *IPSN. Proceedings of the 4th International Symposium on Information Processing in Sensor Networks, Los Angeles, 25–27 April 2005* (IEEE Piscataway, 2005), pp. 457–463
18. XF Jiang, J Polastre, D Culler. Perpetual environmentally powered sensor networks, in *IPSN. Proceedings of the 4th International Symposium on Information Processing in Sensor Networks, Los Angeles, 25–27 April 2005* (IEEE Piscataway, 2005)
19. C Park, P Chou. AmbiMax: autonomous energy harvesting platform for multi-supply wireless sensor node, in *SECON. The 3rd Annual IEEE Communications Society Sensor and Ad Hoc Communications and Networks, Reston, 28–28 Sept 2006* (IEEE Piscataway, 2006), pp. 168–177
20. F Simjee, P Hou. EverLast: long-life, supercapacitor-operated wireless sensor node, in *ISLPED. Proceedings of the 2006 International Symposium on Low Power Electronics and Design, Tegernsee, 4–6 Oct 2006* (IEEE Piscataway, 2006), pp. 179–185
21. G Qiao, Y Hong, G Sun, O Yang, Corrosion energy: a novel source to power wireless sensor. IEEE Sensors J. **13**(4), 1141–1142 (2013)
22. L Li, N Shroff, R Srikant, Asymptotically optimal energy-aware routing for multihop wireless networks with renewable energy sources. IEEE/ACM Trans. Netw. **15**(5), 1021–1034 (2007)
23. A Kailas, M Ingram, Y Zhang. A novel routing metric for environmentally-powered sensors with hybrid energy storage systems, in *Wireless VITAE, Aalborg, 17–20 May* (IEEE Piscataway, 2009), pp. 42–47
24. Z Eu, H Tan, W Seah. Routing and relay node placement in wireless sensor networks powered by ambient energy harvesting, in *WCNC. IEEE Wireless Communications and Networking Conference, Budapest, 5–8 April 2009* (IEEE Piscataway, 2009), pp. 1–6
25. RS Liu, KW Fan, Z Zheng, P Sinha, Perpetual and fair data collection for environmental energy-harvesting sensor networks. IEEE/ACM Trans. Netw. **9**, 947–960 (2010)
26. RS Liu, P Sinha, C Koksal. Joint energy management and resource allocation in rechargeable sensor networks, in *INFOCOM. Proceedings of IEEE INFOCOM , San Diego, 14–19 March 2010* (IEEE Piscataway, 2010), pp. 1–9
27. S Chen, P Sinha, N Shroff, C Joo. Finite-horizon energy allocation and routing scheme in rechargeable sensor networks, in *INFOCOM. Proceedings of IEEE INFOCOM , Shanghai, 10–15 April 2011* (IEEE Piscataway, 2011), pp. 2273–2281
28. D Hochbaum, Efficient bounds for the stable set, vertex cover and set packing problems. Discrete Appl. Math. **6**, 243–254 (1983)
29. S Ramanathan, A unified framework and algorithm for channel assignment in wireless networks. Wireless Netw. **5**(2), 81–95 (1999)
30. A Panconesi, R Rizzi, Some simple distributed algorithms for sparse networks. Distributed Comput. **14**, 97–100 (2001)
31. T Moscibroda, R Wttenhofer. Coloring unstructured radio networks, in *SPAA. Proceedings of the Seventeenth Annual ACM Symposium on Parallelism in Algorithms and Architectures, Las Vegas, 18–20 July 2005* (ACM New York, 2005), pp. 39–48
32. M Kodialam, T Nandagopal, Characterizing achievable rates in multi-hop wireless network. MobiCom. **13**, 868–880 (2003)
33. M Kodialam, T Nandagopal. Characterizing the capacity region in multi-radio multi-channel wireless mesh networks, in *MobiCom. Proceedings of the 11th Annual International Conference on Mobile Computing and Networking, Cologne, 28 Aug–2 Sept 2005* (ACM New York, 2005), pp. 73–87
34. G Sharma, R Mazumdar, N Shroff. On the complexity of scheduling in wireless networks, in *MobiCom. Proceedings of the 12th Annual*

International Conference on Mobile Computing and Networking, Los Angeles, 24–27 Sept 2006 (ACM New York, 2006), pp. 227–238
35. C Joo, X Lin, N Shroff, Understanding the capacity region of the greedy maximal scheduling algorithm in multihop wireless networks. IEEE/ACM Transactions on Networking. **17**, 1132–1145 (2009)
36. P Wan, O Frieder, X Jia, F Yao, X Xu, S Tang. Wireless link scheduling under physical interference model, in *INFOCOM. Proceedings of IEEE INFOCOM , Shanghai, 10–15 April 2011* (IEEE Piscataway, 2011), pp. 838–835
37. MA Koulali, A Kobbane, ME Koutbi, H Tembine, H Ben-Othman, Dynamic power control for energy harvesting wireless multimedia sensor networks. EURASIP J. Wireless Commun. Netw. **158**, 1–8 (2012)
38. A Kumar, M Marathe, S Parthasarathy, A Srinivasan. End-to-end packet-scheduling in wireless ad hoc networks, in *SODA '04. Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms, New Orleans, 11–14 Jan 2004* (ACM New York, 2004), pp. 1021–1030
39. W Wang, Y Wang, XL Li, XL Song, O Frieder. Efficient interference-aware TDMA link scheduling for static wireless networks, in *MobiCom. Proceedings of the 12th Annual International Conference on Mobile Computing and Networking, Los Angeles, 24–27 Sept 2006* (ACM New York, 2006), pp. 262–273
40. Y Wang, W Wang, X Li, W Song, Interference-aware joint routing and TDMA link scheduling for static wireless networks. IEEE Trans. Parallel Distributed Syst. **19**(12), 1709–1725 (2008)
41. Y Gu, T He. Data forwarding in extremely low-duty-cycle sensor networks with unreliable communication links, in *SenSys'07. Proceedings of the 5th International Conference on Embedded Networked Sensor Systems, Sydney, 6–9 Nov 2007* (ACM New York, 2007), pp. 321–334
42. B Nazir, H Hasbullah, SA Madani, Sleep/wake scheduling scheme for minimizing end-to-end delay in multi-hop wireless sensor networks. EUROS. **92**, 1–14 (2011)
43. G Sun, G Qiao, B Xu, Corrosion monitoring sensor networks with energy harvesting. IEEE Sensor J. **11**(6), 1476–1477 (2011)
44. G Nan, G Shi, Z Mao, M Li, CDSWS: coverage-guaranteed distributed sleep/wake scheduling for wireless sensor networks. EURASIP J. Wireless Commun. Netw. **44**, 1–14 (2012)
45. M Cozzens, F Roberts, T-colorings of graphs and the channel assignment problem. Congressus Numerantium. **35**, 191–209 (1982)
46. A Raychaudhuri, Further results on T-coloring and frequency assignment problems. SIAM J. Discrete Math. **7**(4), 605–613 (1994)
47. D Liu, T-graphs and the channel assignment problem. Discrete Math. **161**, 197–205 (1996)

doi:10.1186/1687-1499-2013-223
Cite this article as: Sun *et al.*: Efficient link scheduling for rechargeable wireless ad hoc and sensor networks. *EURASIP Journal on Wireless Communications and Networking* 2013 **2013**:223.