

MAC Scheduling with Low Overheads by Learning Neighborhood Contention Patterns

Yung Yi, *Member, IEEE*, Gustavo de Veciana, *Senior Member, IEEE* and Sanjay Shakkottai, *Member, IEEE*

Abstract—Aggregate traffic loads and topology in multi-hop wireless networks may vary slowly, permitting MAC protocols to ‘learn’ how to spatially coordinate and adapt contention patterns. Such an approach could reduce contention, leading to better throughput. To that end we propose a family of MAC scheduling algorithms and its general conditions, if satisfied, ensure lattice-throughput-optimality (i.e., achieving any rate-point on a uniform discrete-lattice within the throughput-region). This general framework for lattice-throughput-optimality allows us to design MAC protocols which meets various objectives and conditions. In this paper, as instances of such a lattice-throughput-optimal family, we propose distributed, synchronous contention-based scheduling algorithms under graph and physical interference model that (i) is lattice-throughput-optimal, (ii) does not require node location information, and (iii) has a signaling complexity that does not depend on network size. Thus, it is amenable to simple implementation, and is robust to network dynamics such as topology and load changes. Further, we propose a heuristic, which belongs to the proposed throughput-optimal family, for achieving faster convergence, leading to a better transient throughput.

I. INTRODUCTION

A. Motivation and Overview

Since the seminal work [2] on throughput maximization, there has been growing interest in distributed MAC scheduling algorithms with *provable* throughput-guarantees over wireless multi-hop networks. The problem is to find a “throughput-optimal” algorithm, i.e., a link scheduling algorithm that stabilizes the system whenever possible, subject to the constraints on the sets of simultaneously schedulable links. Research on distributed algorithms is crucial in wireless multi-hop networks due to difficulty in having a centralized coordinator. Recent advances on this area propose various types of algorithms: maximal/greedy scheduling (e.g., [3]–[7]), pick-and-compare scheduling (e.g., [8], [9]), and random access scheduling (e.g., [10], [11]).

In this paper, our objective is to develop distributed, synchronous contention-based scheduling algorithms for both graph-based and physical interference model (i.e., path-loss based interference with a minimum SIR requirement for successful packet decoding at receivers) that: (i) is lattice-throughput-optimal, (ii) does not require node location/explicit path-loss information, and (iii) has a signaling complexity that does not depend on network size. Thus, it is amenable to simple implementation, and is robust to network dynamics

such as topology and load changes. Our work is motivated by the following:

(a) *Achieving low control overhead.* In a distributed implementation of link scheduling with throughput-guarantees, control overheads are unavoidable, since nodes need to exchange control messages to choose a “good” schedule over time-slots and achieve provable throughput-guarantees. Clearly, low control overhead is of great importance for both ease of implementation and higher throughput, particularly in wireless multi-hop networks with limited bandwidth resources.

(b) *‘Universality’ for interference models.* Existing algorithms to date have mostly targeted a particular set of graph-based interference models. In a more realistic model, one would identify interference relationships based on SINR (Signal-to-Interference-Noise-Ratio), considering “aggregate” interference. There is little work on *throughput-optimal* algorithms under this physical interference model. The graph interference model enables us to leverage well-developed graph theoretic results. The physical interference model reflects more realism, but the complex interference coupling and aggregate interference becomes an impediment to development of a distributed scheduling algorithm with a provable throughput-guarantee. Since both models (graph-based and physical interference) have their proponents, it would be useful to develop a unified scheduling framework, which allows the development of throughput-optimal distributed scheduling algorithms for both models.

In this paper, we first propose a MAC scheduling framework and associated conditions, which, if satisfied, ensure *lattice-throughput-optimality*, i.e., achieve any rate-point on a discrete-lattice (which can be made arbitrarily fine by a suitable choice of parameters) within the throughput-region, irrespective of underlying interference model. Next, as an instance of the proposed family of lattice-throughput-optimal algorithms, we develop a synchronized contention based algorithm, RCAMA (Randomized Contention Aware Multiple Access), which only requires very simple contention signaling on each time-slot. Further, RCAMA operates in a “dynamic” manner, i.e., schedules (determined in a distributed manner) are initially not necessarily conflict-free, but each node progressively adapts its schedule and converges to an optimal conflict-free schedule (see Figure 1). Our approach to contention signaling enables each node to ‘learn’ its neighborhood’s contention patterns in an autonomous manner and adapt to changes in traffic load and network topology.

Note that under the physical interference model, a *centralized scheduling* algorithm requires exact topology knowledge (i.e., node locations, path-loss coefficients and network con-

This research was supported by NSF Grants CNS-043507, CNS-0347400, CCF-0634898, and the DARPA ITMANET program. The work of G. de Veciana was supported in part by NSF Award CNS-0721532. A shorter version of this paper appeared in the Proceedings of IEEE Infocom 2007 [1].

nectivity) to achieve throughput-optimality – this information is needed to compute the amount of interference generated by simultaneously activated nodes, which in-turn is needed for computing the optimal schedule. However, somewhat surprisingly, the proposed distributed algorithm achieves lattice-throughput-optimality *without* centralized geographical information. To the best of our knowledge, this paper is the first to propose a distributed, throughput-optimal scheduling algorithm under the physical interference model.

We focus on a random access approach that uses memory of past (local) schedules in order to provide a higher throughput-guarantee than the algorithms described in current literature (we refer to Section I-B for additional discussion). “Random access” has been popularly implemented in a practical network due to its ease of deployment. In this paper, we assume that a node has explicit knowledge of its *local* (long-term) offered load (i.e., offered load on each of its outgoing links). We are able to prove that this extra local-information at nodes leads to a distributed, lattice-throughput-optimal algorithm that requires only three and two stages of simple contention signaling under the physical and graph interference models, respectively, irrespective of network size.

In practice, depending on the types of services supported by the network, information on the offered load can either be explicitly given to the nodes or be measured by the nodes. If we have a guaranteed-service network based on a resource reservation signaling (e.g., RSVP [12]), the amount of load could be known a priori by nodes in the path of a reserved flow. However, in a typical best-effort service network, the amount of load is not explicitly provided to the nodes, but the nodes could measure/estimate offered load over a suitable time-period. The main motivation for RCAMA is that although individual (end-to-end) traffic loads may change quickly, the aggregates on some congested links may, in many relevant applications, change more slowly and locally. Similarly, node mobility (that leads to changes in topology and load) might be slow enough to permit a MAC scheduler to learn and exploit the offered traffic characteristics so as to quickly realize “good” schedules. Because the loads may exhibit some variation, or measurements may be noisy, a node may use an upper estimate for it.

B. Related Work

Scheduling achieving maximum throughput dates back to the work [2] by Tassioulas and Ephremides, where the authors show that *max-weight* scheduling proves to maximize throughput in constrained queueing systems. Max-weight scheduling essentially corresponds to a maximum weight independent set problem, which is known to be NP-hard, and thus requiring significant overhead. *Maximal/greedy scheduling* (e.g., [3]–[5], [7]) has been proposed with the objective of polynomial-complexity overhead, however at the cost of a reduced throughput-region. The work in [10], [11] employs random access approach, where transmission attempt probability is decided based on the queue lengths of interfering links and just requires constant overheads with throughput-regions close to that of maximal/greedy scheduling. In this spirit (i.e., queue-length based access probability selection), related work also

includes [13]. Research efforts, called a *pick-and-compare approach* have been developed to achieve the maximum throughput with polynomial complexity motivated by the work in [14]. The pick-and-compare based algorithms [8], [9], [15] are essentially an “infrequent” computation of the max-weight schedule, such that complexity on each time-slot is divided into multiple time-slots [16], [17]. Thus pick-and-compare algorithms reduce (average) complexity without loss of throughput, but incurs exponentially large delay.

Note that all related work that we have surveyed so far does not make explicit use of the offered load (i.e., statistics of the load over any link in the topology). This indicates that scheduling algorithms, which are unaware of arrival rate, pay significantly large cost in terms of control overhead or delay. As mentioned earlier, in this paper we propose an algorithm with a small knowledge of local offered load can significantly reduce control overhead as well as achieve large throughput-region. Most importantly, unlike much of the earlier work (which are predominantly for graph-based interference models), the current approach works for physical interference models as well.

We conclude this subsection with a brief survey of related work to scheduling under the physical interference model (these algorithms however have a different objective from that in this paper). The work of [18]–[20] develops a mathematical programming formulation for minimizing the frame size over a TDMA wireless multi-hop networks, and proposes a distributed heuristic [18], [19] that considers only closest interferers and a centralized heuristic which is used as a benchmark [20]. The authors in [21] define the “scheduling complexity,” i.e., minimum amount of time required until every link is scheduled at least once, which is studied in an asymptotic manner. In [22], [23], the authors have focused only on computing maximum throughput under the physical interference model by jointly considering routing, MAC scheduling, and power control in an optimization framework, but no practical, throughput-optimal, distributed algorithm is presented.

C. Main Contributions and Organization

The main contributions of this paper are as follows:

- (i) We first propose a scheduling framework (DRS: Dynamic Randomized Scheduling), that achieves any rate-point on a uniform discrete-lattice within the throughput region (i.e., lattice-throughput-optimal). To that end, we give two general conditions, which, if satisfied, ensure that an algorithm in the DRS family is lattice-throughput-optimal, and we further study their rate of convergence.
- (ii) Next, as an instance of the DRS family, we propose a synchronous contention-based algorithm, RCAMA-MAX (Randomized Contention-Aware Multiple Access-MAX), where multi-stage contention signaling in conjunction with randomized time-slot selection is used. We prove lattice-throughput-optimality of RCAMA, by showing that RCAMA satisfies the two conditions in (i). The key ideas are summarized as a priority-based slot access with preemption and persistence properties. In spite of lattice-

throughput-optimality of RCAMA-MAX for general network topologies, it may not be hard to be practically implemented due to its large power consumption in signaling. We propose a practical alternative, RCAMA-VIR, whose signaling power consumption is significantly smaller than that of RCAMA-MAX, but achieves lattice-throughput-optimality for many practical network topologies in wireless multi-hop networks.

- (iii) We propose an adaptive variation of RCAMA, AR-CAMA (Adaptive RCAMA), which again satisfies the two conditions in (i) and adaptively biases slot selection probabilities based on the past contention histories. We show via simulation that only a short duration of memory is required to increase performance, resulting in good adaptation to load/topology changes.

The rest of this paper is organized as follows: We begin with a description of the system model, notations, and definitions in Section II. Next, in Section III, we define the DRS algorithm family, and present two general conditions for a DRS algorithm to be lattice-throughput-optimal. In Sections IV and V, we propose RCAMA as an instance of such a lattice-throughput-optimal family. Finally, in Section VIII, we validate our results using simulations.

II. SYSTEM MODEL, NOTATIONS, AND DEFINITIONS

A. System Model

We assume that time is slotted. A time-slot duration is suitably chosen to accommodate the transmission of one fixed-size packet. We model the wireless multi-hop network by a graph $\mathcal{G}(\mathcal{L}, \mathcal{V})$, where \mathcal{L} and \mathcal{V} denote a set of directed links and nodes, respectively. We assume that for any link between two nodes there is a counter-part in the opposite direction. We denote a directed link from node i to node j by $i \rightarrow j$. For concreteness, the wireless system under study has a *single* frequency/code and each node is time-synchronized and has a half-duplex radio.

We assume a fixed power model, where a transmitter uses the power P for data transmission, and SINR (Signal-to-Interference-Noise Ratio) is considered to determine success or failure of a transmission.

A message from i to j is *decodable*, if

$$\frac{G_{ij}P}{\eta_j + \sum_{k \in \mathcal{V}_I(i)} G_{kj}P} \geq \gamma, \quad (1)$$

where $\mathcal{V}_I(i)$ is the set of nodes transmitting simultaneously with i on a given time-slot, G_{ij} is the propagation loss from i to j , and η_j is the thermal noise power at j . The SINR threshold γ depends on the desired bit rate, bit error rate, and design parameters such as modulation, coding, and so on.

In practice, in addition to interference, wireless links are prone to errors due to many other factors (e.g., fading). This leads to high packet loss rate detrimental to upper-layer performance. Thus, in many MAC protocols, reliability is provided by acknowledging transmissions and possibly retransmitting. Thus, we say that a transmission over $i \rightarrow j$ is *successful*, if both the *data* message from i to j and the corresponding *ack* message from j to i are decodable at j and i , respectively,

where the ack message from j will be sent only when the data message is decodable at i .

B. Lattice-Throughput-Optimality: Notation and Definitions

Definition II.1. A link schedule $\vec{A} = (A_l \in \{0, 1\} : l = 1, \dots, |\mathcal{L}|)$, A_l , where $A_l = 1$ if the link l is scheduled for attempted transmission, and 0 otherwise.

A link schedule \vec{A} is said to be *successful*, if the transmissions scheduled by \vec{A} are successful when they occur simultaneously. We denote the collection of all *successful* link schedules by \mathcal{A} .

Definition II.2. We define the throughput region Λ by:

$$\Lambda = \left\{ \vec{\alpha} \mid \vec{\alpha} = \sum_{\vec{A}_i \in \mathcal{A}} \beta_i \vec{A}_i, \quad 0 \leq \beta_i \leq 1, \quad \sum_{i=1}^{|\mathcal{A}|} \beta_i = 1 \right\}.$$

Definition II.3. For any fixed positive integer F , we define the F -lattice-throughput region Λ_F by:

$$\Lambda_F = \left\{ \vec{\alpha} \mid \vec{\alpha} = \sum_{\vec{A}_i \in \mathcal{A}} \beta_i \vec{A}_i, \quad \beta_i = \frac{k_i}{F}, \quad \sum_{i=1}^{|\mathcal{A}|} k_i = F, \quad k_i \in \{0, \dots, F\} \right\}.$$

Intuitively, Λ_F is the lattice-sampling of Λ with adjacent points having a distance of $1/F$. Note that $\Lambda = \text{CL}(\cup_{F=1, \dots, \infty} \Lambda_F)$, where $\text{CL}(Z)$ is the closure of a set Z .

A *scheduling algorithm* Π chooses a sequence of link schedules (which are not necessarily successful), $(\vec{A}[s] : s = 0, 1, \dots)$, where $\vec{A}[s]$ is the link schedule on time-slot s .

Definition II.4. For a fixed F , the offered load $\vec{\rho}$ is said to be F -lattice-feasible if $\vec{\rho} \in \Lambda_F$. A scheduling algorithm Π is said to be F -lattice-throughput-optimal, if Π stabilizes the system for any F -lattice-feasible load.

For a F -lattice-feasible load $\vec{\rho}$, by multiplying the offered load by F , we henceforth deal with positive integer-valued load, $\vec{\theta} \in \mathcal{Z}_+^{|\mathcal{L}|}$, i.e., θ_l corresponds to the number of requested time-slots over link l out of F time-slots. We call a group of F time-slots a *frame* throughout this paper.

In our framework, the lattice-parameter F is a system-wide parameter that is known to every node in the network a-priori. Thus, throughout this paper, we implicitly assume that the lattice-parameter, denoted by F , is fixed. Further, for simplicity, we use the terms ‘‘throughput-optimal’’ and ‘‘feasible’’ to refer to ‘‘ F -lattice-throughput-optimal’’ and ‘‘ F -lattice-feasible,’’ respectively, unless explicitly needed.

III. DYNAMIC RANDOMIZED SCHEDULING: CONDITIONS FOR THROUGHPUT-OPTIMALITY

In this paper, we consider ‘‘frame-based’’ scheduling algorithms, where scheduling patterns are determined on a frame-by-frame basis (i.e., F time-slots)¹, and we will see that it is sufficient to consider such class of algorithms.

Definition III.1. We define a frame schedule (FS) to be a consecutive sequence of F link schedules, i.e., an $|\mathcal{L}| \times F$

¹Thus, we henceforth use a term ‘time-slot s ’ to refer to the s -th time-slot inside a frame. We typically use ‘ s ’ and ‘ t ’ to refer to the indexes of a time-slot and a frame, respectively.

matrix, $C(F, \vec{\theta}) = (c_{ls} : l = 1, \dots, |\mathcal{L}|, s = 1, \dots, F)$, where $c_{ls} = 1$ if a transmission is scheduled over link l on time slot s , and 0 otherwise. Further, the l -th row vector of $C(F, \vec{\theta})$, is said to be a slot schedule over l . A FS $C(F, \vec{\theta})$ is said to be feasible, if all of F link schedules (column vectors) in $C(F, \vec{\theta})$ are successful.

As mentioned in Section I, we assume that a node has knowledge only of the local offered load (i.e., arrival rate) on each of its outgoing links. Thus, for all $l \in \mathcal{L}$, $\theta_l = \sum_{s=1}^F c_{ls}$, i.e., the number of scheduled time-slots on each link is equal to the load offered on that link.

Definition III.2. We additionally define a transmission priority, $R = (r_{ls} : l = 1, \dots, |\mathcal{L}|, s = 1, \dots, F)$ where $r_{ls} = 1$ ($r_{ls} = 0$) if $c_{ls} = 1$ and its priority is high (low), and NULL otherwise ($c_{ls} = 0$).

In this paper, we consider the following class of frame-scheduling algorithms:

Definition III.3. A dynamic randomized scheduling (DRS) algorithm randomly chooses a sequence of $(C[t], R[t] : t = 0, 1, \dots)$ over frames, where $C[t]$ and $R[t]$ are the FS and the transmission priority at frame t , respectively. A randomly chosen $(C[t], R[t])$ at frame t may depend on FSs of the previous, say m , frames. In this case we say that a DRS algorithm has history m . Note that in a DRS algorithm without priority, $R[t]$ is not in use.

Remark III.1. It is clear that $\vec{\theta}$ is F -lattice-feasible, if and only if there exists a feasible frame schedule $C(F, \vec{\theta})$, by Definition II.3. Our objective in this paper is to develop a DRS scheduling algorithm which finds a feasible frame schedule within a finite number of frames, and sustains the schedule thereafter, for any given feasible load. It can be easily seen that a DRS algorithm satisfying such properties achieves lattice-throughput-optimality. Thus, it suffices to consider the family of DRS algorithms.

Now, we derive two conditions, which, if met, ensure that a DRS algorithm is throughput-optimal. We first define a “distance” between two FSs (under the same topology and load), $C = (c_{ls})$ and $C' = (c'_{ls})$, to be:

$$D(C, C') = \sum_{l=1}^{|\mathcal{L}|} \theta_l - \sum_{l=1}^{|\mathcal{L}|} \sum_{s=1}^F c_{ls} \times c'_{ls}. \quad (2)$$

Note that $D(C, C') = 0$ implies $C = C'$.

Definition III.4. For a given fixed load and topology, let the current frame to be t_i .

- 1) **Finite Sustenance Condition (FSC).** If $C[t_i]$ is feasible, $C[t] = C[t_i]$, w.p. 1 $\forall t > t_i$.
- 2) **Finite Improvement Condition (FIC).** If $C[t_i]$ is not feasible, for any feasible FS C^* , there is a $t < \infty$ (not dependent on C^*), such that $D(C[t_i], C^*) > D(C[t_i + t], C^*)$ with positive probability.

The finite sustenance condition means that if a FS converges to a feasible one, it has to be sustained thereafter. The finite improvement condition is such that before converging to a

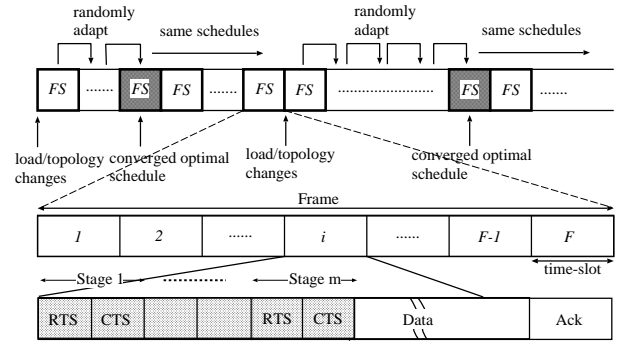


Fig. 1. Frame and slot structure of RCAMA

feasible FS, a sequence of FSs over frames tend to be progressively “closer” to a feasible FS with positive probability.

Subject to these two conditions, we have the following theorem:

Theorem III.1. For any fixed feasible offered load and topology, consider a DRS algorithm Π which satisfies the finite improvement and sustenance conditions. We have that

- 1) Π converges to a feasible FS, and thus Π is throughput-optimal.
- 2) Let $\tau_{\Pi}(C)$ be the convergence time of Π to a feasible FS for a given initial frame schedule C . Then, $\forall t \in \mathbb{Z}_+$, there exist constants $0 < K_{\Pi} < \infty$ and $0 < p_{\Pi} < 1$, such that $\Pr\{\tau_{\Pi}(C) > tK_{\Pi}\} \leq p_{\Pi}^t$.

The sketch of proof is as follows: First, it is easily seen that a sequence of FSs over frames forms a Markov chain. Then, the finite improvement condition implies that we can construct a converging path to a feasible FS (say, C^*) within a finite time, since $D(C, C^*)$ is upper-bounded. The complete proof is presented in the Appendix.

The finite sustenance and improvement conditions described above enable us to verify throughput-optimality of an instance of the DRS family. In addition, it allows customization or enhancement of an algorithm with its throughput-optimality maintained, as long as the modified version satisfies those conditions. In this paper, we develop a “base-line” DRS algorithm with history 1. We later discuss how such base-line algorithms with history 1 can be extended to adaptive versions with multiple frame histories for better adaptation to load/topology changes and faster convergence to the optimal schedule in Section VII.

IV. RCAMA: OVERVIEW AND PER-FRAME OPERATION

A. Overview

The general frame and time-slot structure of RCAMA are shown in Figure 1. A time-slot is divided into two parts: time for contention signaling and time for data and ack transmission (TX)². We will describe RCAMA by dividing its behavior into two different time-scales: (i) per-frame operation, where each node randomly determines the slot-schedules for the TXs

²For notational simplicity, we use the term ‘TX’ to refer to the word ‘transmission’ throughout this paper.

over its adjacent outgoing links, and (ii) per-slot operation, where a node initiates a RTS/CTS-like contention signaling to resolve contentions and implicitly *learn* contention patterns in the neighborhood. In this section, we describe only per-frame operation, and per-slot operations will be discussed in Sections V.

The RCAMA is designed to ensure the following two properties:

- 1) **Persistence:** A successful TX at a given time-slot at the current frame persists on the same slot at the next frame.
- 2) **Preemption:** An unsuccessful TX can preempt a time-slot (with positive probability) used by a persistent successful TX.

As discussed earlier, it suffices to show that the system *converges* to a feasible FS to achieve throughput-optimality. By the persistence property, once the system reaches a “good” (i.e., feasible) FS, it stays in that FS. Preemption property ensures that there is no deterministic “winner-loser” relationships among TXs, and enables the system to avoid deadlocks, i.e., being stuck in a “bad” FS. These two properties ensure that the system will visit arbitrary FSs, and finally reach a feasible FS, which is sustained thereafter. We satisfy these two properties by assigning priority to scheduled TXs. More specifically, by assigning high priority to unsuccessful TXs and low priority to persistent successful TXs, respectively, we allow a newly scheduled unsuccessful TX on a time-slot to beat existing successful ones.

In addition to provable throughput-optimality, by using a low-cost contention signaling (i.e., message complexity does not depend on network size), the algorithm can adapt to load and topology changes by “learning” local contention patterns. In other words, RCAMA does not need any explicit mechanism to inform the nodes of such network changes, and it automatically avoids the situation where multiple time-slots are commonly accessed by interfering links. Further, application of non-uniform time-slot access probability for unsuccessful TXs enable the system to learn local contention levels, and to distribute scheduled TXs at different time-slots in a more efficient manner (see Section VII).

We note that a similar idea of using multiple priorities was introduced in Z-MAC [24]. However, Z-MAC considers only the graph-based interference model, and its major objective of multiple priorities is to solve the hidden terminal problem with no provable throughput-guarantee, whereas we use two-level priority to get both provable convergence and throughput-guarantee.

B. Per-Frame Operation: Randomized Slot-Selection

When each frame starts, each node (say, $v \in \mathcal{V}$) determines the slot-schedules and contention priorities for the TXs over its adjacent outgoing links. To do this, the following simple rules are used:

Rule IV.1 (Slot and Priority Selection Rule).

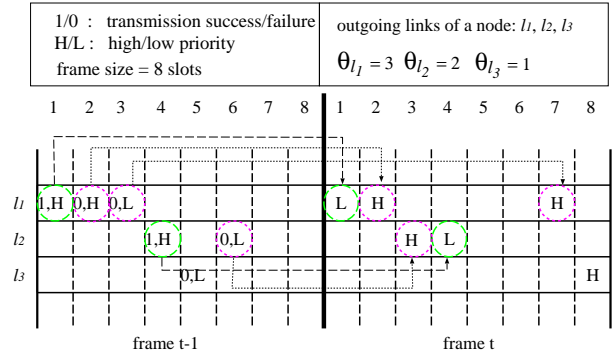


Fig. 2. Example of Rule IV.1: Since at frame $t-1$, the TX over l_1 on time-slot ‘1’ and over l_2 on time-slot ‘4’ were successful, these TXs are scheduled once again with *low* contention priority at the same time-slot positions at frame t . For the unsuccessful TXs over l_1 on time-slots ‘2’ and ‘3’, we randomly choose two time-slots of the remaining time-slots, which were not taken by previously successful TXs (i.e., the node does not consider time-slots ‘1’ and ‘4’ in this random selection). In the example, time-slot ‘2’ and ‘7’ are selected, and they are scheduled with *high* contention priority.

- (i) A successful TX on time-slot s at frame $t-1$ persists on the same time-slot s at frame t , with priority set to be low.
- (ii) If a TX was unsuccessful at frame $t-1$, a time-slot is randomly selected from the time-slots not already taken in (i), and its priority is set to be high.

Rule IV.1(i) corresponds to the **persistence** property. **Preemption** property is satisfied by Rule IV.1(ii) in conjunction with the proposed multi-stage signaling in Section V. An example of Rule IV.1 is given in Figure 2.

V. PER-SLOT OPERATION: PHYSICAL INTERFERENCE MODEL

A. Basic Algorithm

Following the slot-schedules as explained in Section IV-B, on each time-slot, nodes use the three-stage (synchronized) RTS/CTS contention signaling mechanism to resolve contentions, and data/ack TXs follow (see Figure 3 for a representative, pictorial algorithm description).

Definition V.1. A scheduled TX over $i \rightarrow j$ is said to be valid, if j decodes the RTS from i , and i decodes the CTS from j .

Note that in our three-stage signaling, the validity of a TX does not imply success of the TX, i.e., even if RTS/CTS are decoded, the data TX or ack reception can fail. For this reason, we differentiate between *validity* and *success* of a TX.

We denote by \mathcal{H} (resp. \mathcal{M}) a set of links scheduled (on a given time-slot and frame) with high (resp. low) priority, $\mathcal{H}, \mathcal{M} \subset \mathcal{L}$. At each stage, contention signaling is conducted for high and/or low priority TXs. We use the notations \mathcal{H}_V^i and \mathcal{H}_I^i to refer to valid and invalid high priority TXs at stage i , respectively. Similarly, \mathcal{M}_V^i and \mathcal{M}_I^i are used for low priority TXs.

(i) **Stage 1:** Contention signaling is performed for only the TXs in \mathcal{H} , based on which \mathcal{H}_V^1 and \mathcal{H}_I^1 are determined (note that $\mathcal{H}_V^1 \cup \mathcal{H}_I^1 = \mathcal{H}$). The three-stage contention

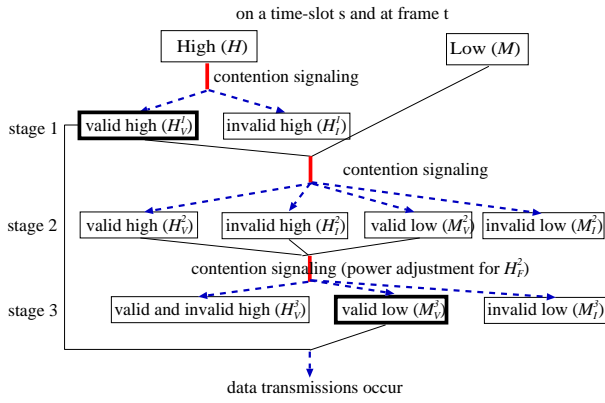


Fig. 3. Three-stage contention signaling in RCAMA under the physical interference model

signaling is constructed to ensure that data TXs *occur* over the links in \mathcal{H}_V^1 , irrespective of the results of the subsequent stages 2 and 3. However, their success is not guaranteed, because TXs in \mathcal{H}_V^1 could fail if their actual data/ack TXs occur together with TXs in \mathcal{M} .

We will later show that it suffices to guarantee the success of all TXs in \mathcal{H}_V^1 on each time-slot for throughput-optimality (see Theorem V.1). Thus, the objective of subsequent stages 2 and 3 is to ensure the success of TXs in \mathcal{H}_V^1 .

(ii) **Stage 2:** Contention signaling is performed for the TXs in \mathcal{H}_V^1 and the TXs in \mathcal{M} , based on which \mathcal{H}_V^2 , \mathcal{H}_I^2 , \mathcal{M}_V^2 , and \mathcal{M}_I^2 are determined. Note that $\mathcal{H}_V^2 \cup \mathcal{H}_I^2 = \mathcal{H}_V^1$, and $\mathcal{M}_V^2 \cup \mathcal{M}_I^2 = \mathcal{M}$. The role of this stage is to identify high priority TXs in \mathcal{H}_V^1 , which fail due to interference from low priority TXs, i.e., identify \mathcal{H}_I^2 .

(iii) **Stage 3:** Contention signaling is performed again for the TXs in \mathcal{H}_V^1 and only for the TXs in \mathcal{M}_V^2 . Recall that our **preemption** property for throughput-optimality is intended to ensure the success of high priority TXs in \mathcal{H}_V^1 . The objective of Stage 3 is to invalidate low priority TXs, which can cause the TXs in \mathcal{H}_I^2 to fail (note that TXs in \mathcal{H}_V^2 will be successful even with interference by low priority TXs). To that end, we employ *signaling power adjustment* in RTS/CTS signaling for TXs of \mathcal{H}_I^2 , i.e., the transmitters and the receivers in \mathcal{H}_I^2 adjust their signaling powers appropriately, such that interfering low priority TXs are invalidated.

(iv) **Data/ack TXs:** Data TXs occur for TXs in \mathcal{H}_V^1 and TXs in \mathcal{M}_V^3 . ACK messages are sent back to the transmitters by the receivers which can decode data.

An example of the three-stage contention signaling in RCAMA is exemplified in Figure 4. It shows how it operates and converges to a feasible FS.

We note that transmission power control for signaling, which is similar to signaling power adjustment in this paper, has been proposed with the main objective of throughput improvement (see [25] and references therein). The approaches in [25], however, do not consider the physical interference model and they do not provide a study of provable performance guarantees (i.e., no throughput-optimal properties).

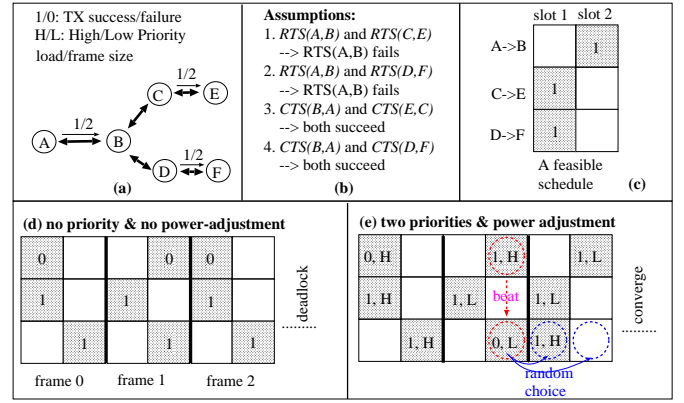


Fig. 4. Example of RCAMA: In absence of contention priority and signaling power adjustment, TX over $A \rightarrow B$ keeps failing with either choice of time-slot ‘1’ or ‘2’, since RTS from A is not decodable at B due to interference from either C or D , over frames. However, in RCAMA, from Rule IV.1, the unsuccessful TX over $A \rightarrow B$ at frame ‘0’ is assigned high priority at frame ‘1,’ and due to stages 2 and 3, B adjusts the power for its CTS (destined to A and broadcast to D), such that CTS from F is not decodable at D (see the frame 1 in (b)). The same procedure can be applied when TXs over $A \rightarrow B$ and $C \rightarrow E$ are assigned high and low on a same time-slot, respectively. By this procedure, the system ultimately converges to a feasible FS.

B. Signaling Power Adjustment and Throughput-Optimality

The remaining question is how to compute the adjusted powers for TXs in \mathcal{H}_I^2 in an efficient, distributed manner, which we will discuss in this section.

We will use the notation $\mathcal{H}_I^2(s)[t]$ to explicitly refer to \mathcal{H}_I^2 on the time-slot s at frame t . We first let $\vec{P}_s^A[t] = (\vec{P}^r, \vec{P}^c)_s[t]$, $\vec{P}^r = (P_i^r)$, $\vec{P}^c = (P_l^c)$, $l \in \mathcal{H}_I^2(s)[t]$ be the adjusted signaling power vector on time-slot s and frame t at stage 3, where \vec{P}^r and \vec{P}^c corresponds to the powers for sending RTS and CTS messages, respectively.

Definition V.2. For any fixed topology and load, consider a sequence of adjusted signaling power vectors, $(\vec{P}_s^A[t] : s = 1, \dots, F, t = 0, 1, \dots)$. RCAMA is said to satisfy High Priority Condition (HPC) with $(\vec{P}_s^A[t])$, if with $\vec{P}_s^A[t]$, all the TXs in $\mathcal{H}_V^1(s)[t]$ are successful, over any time-slot and frame.

As described in Section IV-A, Definition V.2 corresponds to a condition ensuring that “good” high priority TXs (i.e., valid TXs at stage 1) are successful. Now, Theorem V.1 implies that it suffices to guarantee the success of TXs in \mathcal{H}_V^1 by using sufficiently large adjusted power in stage 3 for throughput-optimality of RCAMA. Recall that $\mathcal{H}_V^1 = \mathcal{H}_V^2 \cup \mathcal{H}_I^2$, and TXs in \mathcal{H}_V^2 are guaranteed to be successful even with interference by low priority TXs.

Theorem V.1. For any given fixed topology and load, suppose that RCAMA satisfies HPC with $(\vec{P}_s^A[t])$, then

- 1) RCAMA satisfies the finite sustenance and improvement conditions. Thus, by Theorem III.1, it is throughput-optimal.
- 2) RCAMA satisfies HPC with any $(\vec{Q}_s^A[t])$, where $\vec{Q}_s^A[t] \geq \vec{P}_s^A[t]$, $s = 1, \dots, F, t = 0, 1, \dots$, in element-wise.

This result enables us to develop the following simple, distributed throughput-optimal algorithm:

RCAMA-MAX: All the signaling power adjustment (i.e., $\bar{P}_s^A[t]$) are set to be P_{max} , where P_{max} is the amount of signaling power, such that signaling with P_{max} in a TX invalidates *all other simultaneously scheduled TXs*.

The assumption that P_{max} exists is reasonable for wireless multi-hop networks deployed of a finite size. The following immediate corollary follows:

Corollary V.1 (RCAMA-MAX). *For any fixed topology and feasible load, RCAMA-MAX satisfies HPC, and thus is throughput-optimal.*

Remark V.1. *Note that under the physical interference model, a centralized algorithm needs information on node locations and network connectivity to achieve throughput-optimality. Surprisingly, however, Corollary V.1 implies that there exists a distributed throughput-optimal scheduling algorithm that does not need such centralized topology information.*

In spite of the provable throughput-optimality and the fully distributed nature of RCAMA-MAX, it may not be a practical algorithm, since for a large-scale multi-hop network, P_{max} should be very large. This is not a desirable feature leading to low efficiency of energy utilization and poor transient throughput. In other words, with RCAMA-MAX, every low priority TXs will fail, and only high priority TXs surviving stage 1 will succeed. The main observation behind this limitation of RCAMA-MAX is that we need to consider the “worst-case,” i.e., the case when a large number of far field low priority TXs interfere with a high priority TX (which was valid at stage 1). However, it is known that interference is dominated by a small number of nearby transmissions mainly due to non-linear signal power loss. Using this observation, in the next section, we propose a new distributed algorithm, RCAMA-VIR, which uses far lower powers than P_{max} , but still guarantees throughput-optimality under reasonable assumptions.

C. RCAMA-VIR

The main idea in RCAMA-VIR is to use a sufficiently high power (but not as large as P_{max} in Stage 3 signaling), such that low priority interferers of \mathcal{H}_I^2 can be suppressed. This is done by estimating (and developing bounds) on the interference power.

In this section, we assume the following: (i) A receiver can only measure the total received signal power (the desired signal power plus interference) and know a boolean result about the target SINR (i.e., the target SINR is larger than the threshold γ or not)³; (ii) the propagation loss is modeled by $G_{ij} = 1/d(i, j)^{\alpha(i, j)}$, where $d(i, j)$ is the distance between nodes i and j , and $\alpha(i, j)$ is an “effective” path loss exponent (which may depend on the node-pair), for which each node knows (lower and upper) bounds (i.e., $\underline{\alpha} \leq \alpha(i, j) \leq \bar{\alpha}$); (iii)

³Note that we do not assume that the receiver is able to know the exact SINR value as well as individual or even aggregate pure interference generated by other transmissions.

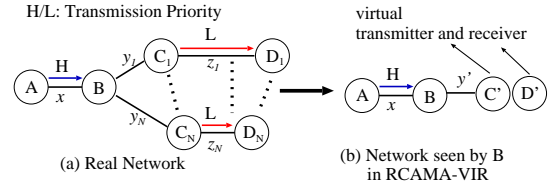


Fig. 5. Example of RCAMA-VIR: we have one high priority and N low priority TXs scheduled on a same time-slot. The high priority TX is clearly valid at stage 1. At stage 2, suppose that at stage 2 an RTS over $A \rightarrow B$ is not decodable due to the aggregate interference of RTSs from C_i to D_i , $i = 1, \dots, N$. Now, B assumes that its RTS decoding failure is due to a single virtual low priority TX. By estimating such aggregate interference, B computes the distance from itself to C' (the virtual transmitter). In the CTS-slot of stage 3, B sets the sufficiently large CTS power to invalidate a CTS from D' (the virtual receiver of C'), based on the “worst-case” assumption that there does not exist a signal power path-loss between C' and D' .

the system is interference-limited⁴.

The transmitter $s(l)$ and the receiver $d(l)$ of link $l \in \mathcal{H}_I^2$ perform the following procedures:

RCAMA-VIR:

- 1) $d(l)$ ($s(l)$) estimates the aggregate interference generated by low priority TXs during RTS (CTS) slot, and assumes that such interference is caused by the transmitter (receiver) of a single *virtual* low priority TX. (see Section V-D for discussion on estimation of the aggregate interference).
- 2) $d(l)$ ($s(l)$) computes an upper-bound on the distance to the transmitter (the receiver) of the virtual TX. This upper-bound is computed based on the bounds on the path loss exponent (i.e., $\underline{\alpha} \leq \alpha \leq \bar{\alpha}$), and the interference estimation in (i).
- 3) By assuming that there is no power path-loss between the virtual transmitter and receiver, $d(l)$ ($s(l)$) computes the adjusted CTS (RTS) power, required to invalidate the virtual TX.

An example of RCAMA-VIR is shown in Figure 5. Note that RCAMA-VIR may not be throughput-optimal, when many far field low priority transmissions are interfering a high priority transmission. However, we will show that RCAMA-VIR achieves throughput-optimality under reasonable assumptions (see Theorem V.2).

D. Estimation of Interference

Note that the major difference between Stages 1 and 2 is the existence of low priority TXs. Thus, it is intuitive to use measurement of the total received signal powers at Stages 1 and 2 and using their differences to estimate the interference by low priority TXs.

Consider a TX $l \in \mathcal{H}_I^2$. We denote by $\hat{R}_{d(l)}^1$ (resp. $\hat{C}_{s(l)}^1$), the total received signal power on RTS (resp. CTS) slots at Stage 1 by $d(l)$ (resp. $s(l)$). Similarly, we use the notations

⁴In this system, the link operates at a sufficiently high γ (SINR threshold), so that the effect of thermal noise is negligible as compared to the interference. However, this can be readily extended to the more general assumption that $0 \leq \eta_j \leq \epsilon \times (\text{interference})$, where ϵ is the ratio of thermal noise to the total interference.

$\hat{R}_{d(l)}^2$ and $\hat{C}_{s(l)}^2$, at Stage 2. We also let $I_{d(l)}^r$ and $I_{s(l)}^c$ be the *exact aggregate low priority interference* to $d(l)$ and $s(l)$. To estimate the interference by low priority transmitters and receivers, we use the values defined in the following: $\hat{I}_{d(l)}^r \triangleq \hat{R}_{d(l)}^2 - \hat{R}_{d(l)}^1$ and similarly, $\hat{I}_{s(l)}^c \triangleq \hat{C}_{s(l)}^2 - \hat{C}_{s(l)}^1$.

Using the above method for estimation, we have

$$\hat{I}_{d(l)}^r \leq I_{d(l)}^r, \quad \hat{I}_{s(l)}^c \leq I_{s(l)}^c, \quad (3)$$

since we have

$$\begin{aligned} \hat{I}_{d(l)}^r &= \hat{R}_l^2 - \hat{R}_l^1 \\ &= \hat{R}_l^2(\mathcal{H}_V^1) + \hat{R}_l^2(\mathcal{M}) - (\hat{R}_l^1(\mathcal{H}_V^1) + \hat{R}_l^1(\mathcal{H}_I^1)) \\ &= \hat{R}_l^2(\mathcal{M}) - \hat{R}_l^1(\mathcal{H}_I^1) \leq I_{d(l)}^r \quad (\because \hat{R}_l^2(\mathcal{M}) = I_{d(l)}^r), \end{aligned} \quad (4)$$

where $\hat{R}_l^i(\mathcal{A})$ corresponds to the total received power by a receiver of link l from RTS TXs in \mathcal{A} during stage i . Similarly, we also have that $\hat{I}_{s(l)}^c \leq I_{s(l)}^c$.

In other words, our estimation is a lower-bound on the exact interference by low priority TXs. This lower-bound in the interference estimation and the bounds on the path loss exponent lead to an upper-bound on the distance to the transmitter/receiver of the virtual TX, which is used in the proof of throughput-optimality of RCAMA-VIR.

Theorem V.2 (RCAMA-VIR). *Suppose that there exists a maximum distance of interference between nodes and a maximum number of interferers, denoted by d_{int} and N_{int} , respectively. If $\sqrt[2\alpha]{N_{int}}(d_{int})^{\alpha/(2\alpha)} \leq d_{min}$, where d_{min} is the minimum distance between two nodes, then RCAMA-VIR satisfies HPC. Thus, it is throughput-optimal from Theorem V.1.*

Theorem V.2 implies that if the inter-node distance is sufficiently large, i.e., node density in a plane is not too high and nodes are distributed in a sufficiently uniform manner, throughput-optimality is provably guaranteed in RCAMA-VIR. The proof is presented in Appendix.

Numerical Example V.1. *As a numerical example, consider the case when $d_{int} = 2 \times d_{min}$ (a typical setting in the IEEE 802.11 DCF by assuming that transmission range is set to be d_{min}) for different values of bounds on path-loss exponents and N_{int} , given by:*

$$\begin{aligned} d_{min} &\geq 2.5 m && \text{if } \bar{\alpha} = \underline{\alpha} = 3, N_{int} = 2, \\ d_{min} &\geq 4 m && \text{if } \bar{\alpha} = \underline{\alpha} = 4, N_{int} = 16, \\ d_{min} &\geq 8 m && \text{if } \bar{\alpha} = 4, \underline{\alpha} = 3, N_{int} = 4. \end{aligned}$$

As discussed earlier, due to non-linear path-loss exponents, the number of interferers affecting other simultaneously scheduled TXs seems to be quite limited, i.e., N_{int} is small, where we have more relaxed condition on d_{min} , which still gives a provable guarantee on performance.

VI. PER-SLOT OPERATION: GRAPH-BASED INTERFERENCE MODEL

A. Interference Model

In the graph-based interference model, for a given network graph, the success of transmission over a link is decided purely by the graphical structure of transmissions over other links (e.g., the number of hops and transmission direction). We

consider the following three models, which are often used in literature:

(i) *Node-exclusive model.* This model mainly considers “primary constraints,” where each node can communicate with at most a single neighbor at a time, i.e., a node cannot transmit and receive simultaneously. A transmission over link l is successful if no transmission occurs over the *one-hop* neighbors of l . This model is known to be appropriate for Bluetooth or FH-CDMA networks (e.g., [4], [26]).

(ii) *Unidirectional model.* In this model, in addition to the “primary constraints” in (i), a “secondary constraint” is placed: a node receiving transmission should not be within the transmission range of other transmissions not intended for it. This model is often used in the single channel TDMA (S-TDMA) system (e.g., [27] and the references therein). Note that this model is *asymmetric*, i.e., there could exist two links l, m , such that l interferes with m , but m does not interfere with l .

(iii) *Bidirectional model.* This model is applied to the system, where both the transmitter and the receiver transmit data and acknowledgements *sequentially*, as in IEEE 802.11 [28]. A transmission over link l is successful if no transmission occurs within the *two-hop* neighbors of l .

B. Algorithm Description

In this section, we describe a per-slot operation for the GIM by focusing on the unidirectional model, which is asymmetric. The algorithm can be readily extended to other symmetric graph interference models.

Similar to the per-slot operation in Section V, nodes use multi-stage RTS/CTS signaling mechanism to resolve contentions for data TN. However, the algorithm under GIM presented here requires only *two-stage* RTS/CTS signaling. Even with this two-stage algorithm, the algorithm is provably throughput-optimal as it satisfies persistence and preemption properties. This is due to the facts that (i) we can utilize the graph structure of interference relations, and (ii) there is no aggregate interference. The algorithm is described below:

(i) Stage 1: The transmitters and the receivers of high priority TNs perform RTS/CTS signaling, which is monitored by the transmitters and the receivers of low priority TNs.

(ii) Stage 2: The transmitters and the receivers of low-priority TNs perform RTS/CTS signaling, if they decide not to release this time-slot under the time-slot release rule (Rule VI.1).

Rule VI.1 (Time-slot release rule). *A low priority TN on a given slot s over link l releases the slot s , if **at Stage 1** of slot s , (i) $s(l)$ senses CTS, (ii) $d(l)$ senses RTS, (iii) $s(l)$ transmits CTS, or (iv) $d(l)$ transmits RTS.*

We say that a (low priority) TN over link l *releases* a time-slot s if $s(l)$ or $d(l)$ does not perform RTS/CTS signaling, but the TN is scheduled on slot s . We say that a node *senses* a control message if it decodes a control message or receives non-decodable packet collision. Here, we assume that a node can distinguish between the absence of any TN and packet

collisions. This is possible due to the graph-based modeling of “binary” interference. A low priority TN decides on its time-slot release (for conflicting high priority TNs) by conforming to the following simple rule:

Note that $d(l)$ is unaware that it is a receiver (of a low-priority TN over link l) before it receives and decodes the corresponding RTS message (at Stage 2) intended for itself. Thus, Rule VI.1(ii) seems to be non-sense. However, if the corresponding RTS message is not correctly received (due to packet collisions among low priority transmissions) at Stage 2, $d(l)$ will not send CTS message, leading to automatic slot-release.

Further, we reiterate that one of the major advantages of *synchronous contention* in Rules VI.1(i) and (ii) is that a node is able to identify the kind of control message, irrespective of its decodability, which helps low priority TN decide on its time-slot release.

Theorem VI.1. *For any fixed topology and load, RCAMA satisfies FSC and FIC under the unidirectional interference model. Thus, it is throughput-optimal from Theorem III.1.*

The proof is presented in the Appendix, and similar to that of Theorem V.1. The major difference is that the HPC condition in Theorem V.1 is not guaranteed here. Instead, however, the graph based formulation of interference relationship in conjunction with multi-stage control messages enables us to prove this result without the HPC condition.

VII. ARCAMA (ADAPTIVE RCAMA)

Note that RCAMA chooses new time-slots for unsuccessful TXs with equal probability in the subsequent frames. In fact, one can potentially increase the rate of convergence or adapt to load changes more effectively by intelligently guessing which time-slot is likely to be successful and by biasing the time-slot access probability. As an example, a time-slot with consecutive success is highly likely to be “safe”, so that it would be beneficial to sustain the corresponding time-slot with higher probability at the next frame than other time-slots. In this section, we propose a general family of variations of DCAMA, ARCAMA (Adaptive RCAMA) family (a subset of the DRS family), which adaptively assigns different time-slot access probabilities, depending on the past contention history. This provides ARCAMA with a more efficient learning of local contention patterns, leading to more robustness to network changes. As shown in Proposition VII.1 below, such variations of RCAMA inherit all throughput-optimal properties.

To that end, each link is assigned its own *slot weight vector*, and the individual nodes maintain slot weight vectors for its adjacent outgoing links. This slot weight vector is updated every frame, mainly based on the TX results (success or failure) at the past frames. To increase/decrease the slot weight vector, we define the *time-slot status*, which corresponds to the result of past TXs on the corresponding time-slots. Then, the slot access probability is set to be *inversely proportional* to the current weight. This biased probability is used for selecting time-slots for unsuccessful TXs. Also, by setting a minimum and maximum for each weight, we can avoid pathological

TABLE I
WEIGHT INCREASE/DECREASE, $S_s^l[t]$: SLOT STATUS OVER LINK l ON SLOT s AT FRAME t

$S_s^l[t-3]$	$S_s^l[t-2]$	$S_s^l[t-1]$	Weight Inc/Dec
SUCC	SUCC	SUCC	$-D_1$
FAIL/IDLE	SUCC	SUCC	$-D_2$
FAIL	FAIL	FAIL	$+I_1$
SUCC/IDLE	FAIL	FAIL	$+I_2$

cases (e.g., the time-slot access probability could be arbitrarily small or close to ‘1’), i.e., there exist \bar{w} and \underline{w} , such that $1 \leq \underline{w} < \bar{w} < \infty$ and $\forall s \in \{1, 2, \dots, F\}, \forall l \in \mathcal{L}$, and $\forall t > 0$, $\underline{w} \leq w_s^l[t] \leq \bar{w}$, where we denote the slot weight vector of link l at frame t by $\vec{w}^l[t] = (w_s^l[t] : s = 1, \dots, F)$.

Proposition VII.1. *For any fixed topology and feasible load and any positive integer $m < \infty$, in ARCAMA with history m , Theorems V.1, V.2, and VI.1 still hold.*

We skip the proof for brevity, since it is analogous to those for RCAMA.

VIII. SIMULATIONS

In this section, we evaluate the performance of the RCAMA and ARCAMA by comparing them to the base-line RANDOM algorithm. The RANDOM algorithm determines slot-schedules (based on the requested loads) in a purely random manner at each frame, and uses a single-level RTS/CTS signaling to gain access to the channel. We choose the RANDOM algorithm as a base-line, since it is similar to Aloha-like strategy (a “standard” algorithm for link scheduling), and behaves like a slotted version of a CSMA-like contention-based scheme.

Weight maintenance algorithm. We use a simple weight maintenance algorithm based on three frame contention history in ARCAMA, where we increase (decrease) a weight more aggressively for back-to-back failures (successes) on a slot over the past three frames. We expect to see even better performance increase when more sophisticated maintenance algorithms are used. The intuition for these choices is that more back-to-back successes at a time slot indicate that the offered loads around the corresponding node at that time-slot are relatively low (i.e., less “congested”), and transmissions in that time-slot are likely to be successful in the future. Similar intuition is applied for back-to-back failures. We have three kinds of time-slot status: SUCC (FAIL), where a transmission occurs and are successful (unsuccessful), and IDLE otherwise. Table I shows the (additive) increase/decrease parameters to adapt slot weights based on the past three transmission result histories, respectively. The parameters are chosen such that $D_1 > D_2 > 0$, and $I_2 > I_1 > 0$. We have used $D_1 = I_1 = 3$, $D_2 = I_2 = 1$, in all simulation results, where the maximum and minimum weights (i.e., \bar{w} and \underline{w}) are set to 30 and 1.

A. Physical Interference Model

Simulation environment. We simulate wireless multi-hop networks with nodes which are randomly distributed in a 1000×1000 meter-square area. Thermal noise power at each

receiver (i.e., η_j), the minimum required SINR level (i.e., γ), and the transmit power level (i.e., P) are set to be -90 dBm, 18 dB, and 15 dBm, respectively. The frame size is 10 time-slots.

Different signaling power adjustment schemes. First, we investigate the effect of different signaling power adjustment schemes on the throughput performance and energy consumption at the “steady” state (i.e., no load or topology changes for some time).

Figure 6(a) shows the network topology and link connectivity generated at random using the parameters above. Figure 6(b) shows a trace of the used powers for different RCAMA versions. Figure 6(c) shows the performance of RCAMA and ARCAMA algorithms for a normalized load by a randomly chosen maximally feasible load⁵, which varies from 50% to 100%. We measure the aggregate normalized throughput for every varying load over 3000 frames. Each point in the graph is the mean value of 50 simulation experiments with different random seed values. In the simulation results, (A)RCAMA-NOR represents the (A)RCAMA without signaling power adjustment at stage 3. Similarly, Figure 6(d) shows the aggregate average power used in contention signaling per one successful transmission for different values of normalized load. With both simulation results, we observe that the algorithms without power adjustment.

From these simulation results, we observe the following: (i) ARCAMA has better transient throughput than RCAMA, (ii) With both ARCAMA and RCAMA, the algorithm without power adjustment has greater transient throughput than other throughput-optimal versions with power adjustment (i.e., (A)RCAMA-VIR and (A)RCAMA-MAX), as well as better energy saving.

Note that, in practice, we may need lower powers than those used by RCAMA-VIR, and the condition on d_{min} in Theorem V.2 can be relaxed. This is because RCAMA-VIR is conservatively designed again by considering the point-of-view from *one* single high priority TX and other low priority TXs for the probable throughput-optimality. In other words, we have not considered the fact that other high priority TXs, which were valid at stage 1, also generate interference to interfering low priority TXs, and interference among low-priority TXs still exists. In fact, as seen above from the simulation results, RCAMA with no signaling power adjustment has a better (transient) performance than RCAMA-MAX and RCAMA-VIR even if it is not provably throughput-optimal. Essentially, overall higher performance than RANDOM is due to accessing the channel with *two-level* priority, which significantly reduces contentions.

Adaptation to load changes. In this simulation, we investigate the effect of network changes in load on the performance of RCAMA-NOR and ARCAMA-NOR algorithm, again for the network topology in Figure 6(a). We generate time-varying loads by a random walk model, where we first determine a normalized offered load of 60% by a randomly chosen maximally feasible load. Then, at the beginning of each frame we randomly choose L_{ch} links and increase their link loads

⁵A load is said to be *maximally feasible* if the resulting system load becomes infeasible with *any* load increase anywhere in the network.

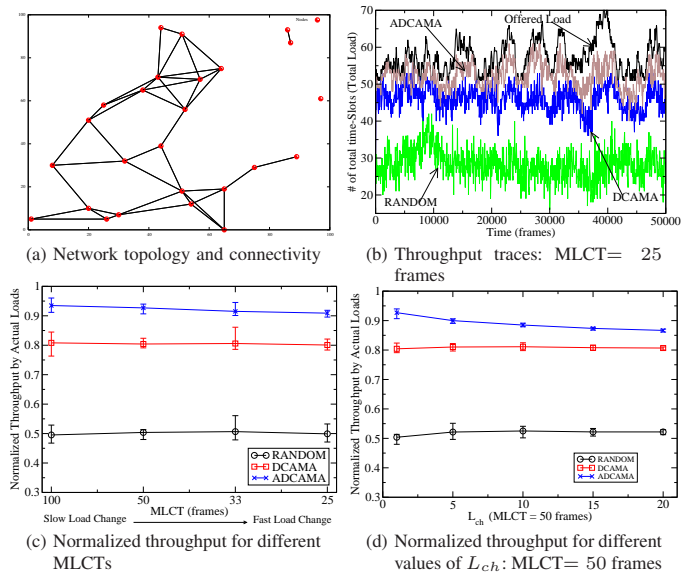


Fig. 8. GIM: Adaptation to load changes

by one slot with probability P_I , decrease their link loads with probability P_D , or stay at the current load (i.e., no change) with probability $1 - P_I - P_D$. For simplicity, in the simulation, we set $\hat{P} \triangleq P_I = P_D$. Thus, higher values of \hat{P} corresponds to a faster load change with time. Then, the mean load change time (MLCT) over L_{ch} links is $1/(2\hat{P})$ frames.

Figure 7(a) shows an example trace of throughput (i.e., number of successful transmission slots) for MLCT = 25 frames and $L_{ch} = 5$, where we observe that ARCAMA algorithm tracks the actual load very well, resulting in nice adaptation to time-varying load changes. Figure 7(b) shows that the throughput (over 50000 frames) normalized by the actual (time-varying) offered load for different values of MLCTs ($L_{ch} = 1$) varying from 25 to 100 frames, where the error bars represent the maximum and minimum values of 10 simulations with different random seed values (i.e., different load changing patterns). For a network with a link capacity of 10 Mbps, and a frame-size of 10 (which corresponds to a 10 msec frame duration), this corresponds to a load change ranging from once every 250 msec to once every 1 seconds. We observe that with ARCAMA algorithm, the normalized throughput is above 90%, whereas the RANDOM achieves about 60%.

B. Graph Interference Model

Simulation environments. Every parameter (e.g., node distribution and frame size) is same as that under PIM. The difference is that under GIM, node connectivity is not determined by the physical parameters, but by the transmission range, which is set to be 200 m .

Adaptation to load changes. Figures 8(a) and (b) shows the network topology and an example throughput trace of the throughput for MLCT = 25 frames, where we again observe that ADCAMA algorithm tracks the actual load very well under GIM. Time-varying loads is generated similarly to the simulation under PIM. Figure 8(c) shows that the throughput

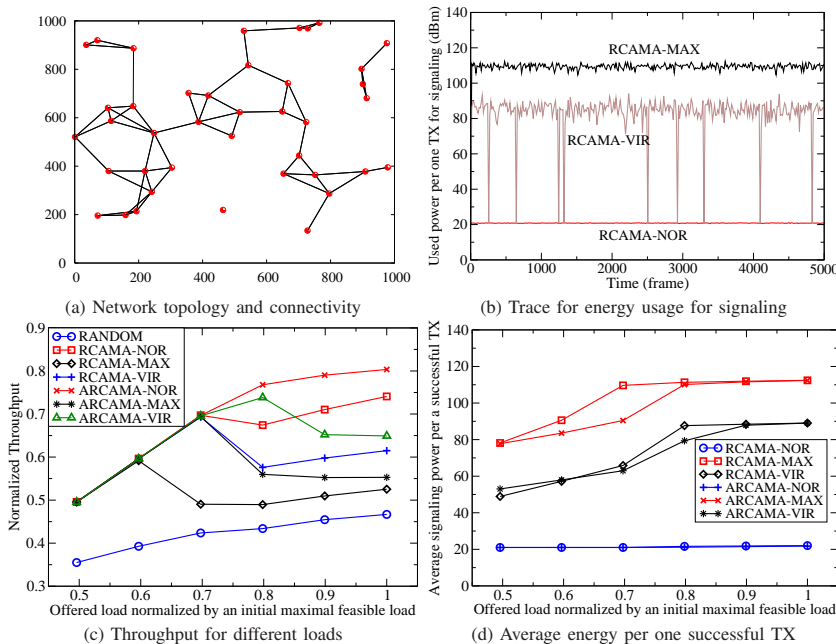


Fig. 6. Steady-state throughput and energy usage

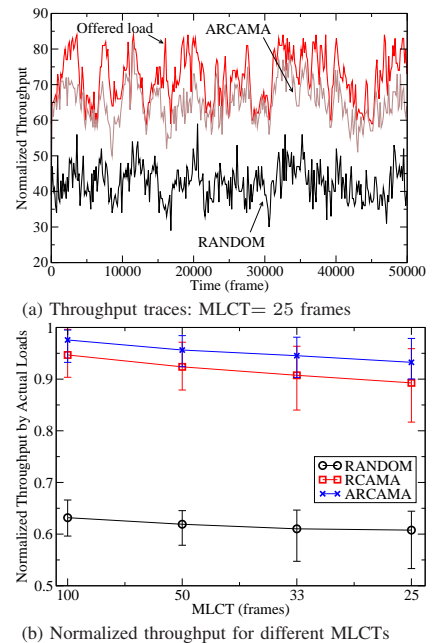


Fig. 7. Adaptation to load changes

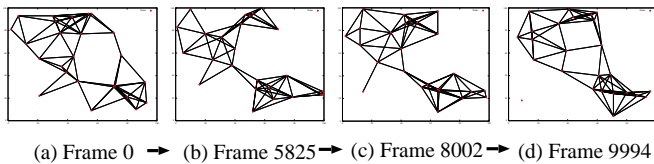


Fig. 9. An instance of topology changes at four time instances.

(over 50000 frames) normalized by the actual (time-varying) offered load for different values of MLCTs ($L_{ch} = 1$) varying from 25 to 100 frames. We observe that with ADCAMA algorithm, the normalized throughput is above 90%, whereas the RANDOM achieves about 50%. Figure 8(d) shows the normalized throughput by actual offered loads in faster load changing scenario, where with MLCT= 50 frames, L_{ch} varies from 1 to 20. As L_{ch} becomes larger, the throughput difference between DCAMA and ADCAMA becomes slightly smaller. This is because with faster changing loads, ADCAMA algorithm does not have sufficient time to completely adapt to changes. However, even in this fast changing regime, ADCAMA shows a 5% throughput improvement over DCAMA.

Adaptation to topology changes. Now, we investigate the effect of topology changes on the performance of DCAMA and ADCAMA algorithm. With time-varying topology changes, new links and existing links are dynamically added and deleted in the network, which possibly changes the offered loads in the links. To simulate practical scenarios of such load/topology changes, we use end-to-end flows to generate offered loads in the network. For a fixed frame size, we first randomly generate a network topology. To initialize the end-to-end flows, we employ the following procedure: we randomly choose two nodes (source and destination of an end-to-end connection), and increase the load in the path of the chosen source-

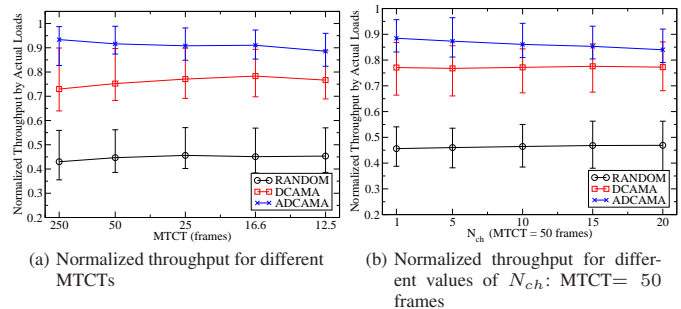


Fig. 10. GIM: Adaptation to topology changes

destination end-to-end connection. We assume that a shortest path routing is used to determine the path. We carry out this random selection of end-to-end connections $2 * |\mathcal{V}|$ times (note that $|\mathcal{V}|$ is the number of nodes in the network), and skip the chosen end-to-end connection if the load increase of its path generates infeasible offered load. Thus, the initially generated loads are very close to a maximally feasible load.

For the evolving time-varying topology, we randomly choose N_{ch} nodes and move each of the nodes independently in one of four directions (north, south, east, or west) by one meter with probabilities of P_T^N , P_T^S , P_T^E , or P_T^W . In the simulations, we set them to be all equal, denoted by P_T . Then, the mean topology change time (MTCT) is $1/(4 * P_T)$ frames. If the topology changes due to node movement, we establish the new end-to-end routes based on the shortest path routing, for initially found end-to-end connections.

Figure 10(a) shows that the throughput (over 10000 frames) normalized by the actual (time-varying) offered load (due to topology changes) for different values of MTCTs varying from 12.5 frames to 250 frames. Again, for a network with

a link capacity of 10 Mbps with the frame size of 10, this corresponds to a change ranging from once every 125 msec to once every 2.5 seconds, which corresponds to a mobile terminal moving with an average velocity of 8 m/sec and 0.4 m/sec. This seems reasonable for a typical slowly varying environment. Further, Figure 10(b) shows the normalized throughput when with MTCT= 25 frames, N_{ch} varies from 1 to 20 nodes. We again observe that with ADCAMA algorithm, the normalized throughput is above 90%, whereas RANDOM achieves at most 60%. Thus, ADCAMA algorithm exhibits good robustness to topology changes.

APPENDIX

We first observe that Rule IV.1 satisfies the Property VIII.1. It basically says that any TX scheduled at some slot s could be re-scheduled with positive probability. In particular, for a successful TX, probability that the same time-slot is chosen is ‘1.’ We will use this property in the proof of throughput-optimality of RCAMA algorithm in Section V-B.

Property VIII.1. *For any time-slot s , and link l , there exists a positive probability that $c_{ls}[t-1] = c_{ls}[t]$, irrespective of $c_{l's'}[t-1], l' \neq l, s' \neq s$.*

C. Proof of Theorem III.1

Proof of Theorem III.1(i): From Remark III.1, for the proof of throughput-optimality, we will show that the system “converges” to a feasible frame schedule. For a DRS algorithm with history m , it is easily seen that a sequence of $(C[t], R[t])$ over frames forms a Markov chain, by defining a system state as

$$\mathbf{X}_m[t] \triangleq \left((C[t-m+1], R[t-m+1]), \dots, (C[t], R[t]) \right),$$

and we let $C[n] = R[n] = 0$, when $n < 0$, for the initial conditions.

Then, from the standard Markov chain theory and the finite sustenance condition, it suffices to show that from *any* state $\mathbf{X}_m[0]$, where $C[0]$ is infeasible, we can construct a finite sequence of times $0 = t_i < t_2 < \dots < t_n < \infty$, such that $C[t_n]$ in $\mathbf{X}_m[t_n]$ is feasible.

Suppose that $C[t_i]$ in $\mathbf{X}_m[t_i]$ is infeasible (otherwise, the result immediately follows). Choose any feasible frame schedule C^* . Then, at frame $t_{i+1} < \infty$, we have the following two-cases with positive probability: (a) from the finite improvement condition, $D(C[t_{i+1}], C^*) < D(C[t_i], C^*)$, or (b) $C[t_{i+1}]$ is feasible (i.e., $D(C[t_{i+1}]) = C^{**}$, $C^{**} \neq C^*$). First, if (b) is true, the result immediately follows. Second, if (a) is true, we are also done, since $D(C[0], C^*)$ is upper-bounded by $\sum_{l=1}^{|\mathcal{L}|} \theta_l$. This completes the proof. ■

Proof of Theorem III.1(ii): Let $n_\theta = \sum_{l=1}^{|\mathcal{L}|} \theta_l$ be the total number of loads. We first note that the total number of frame schedules and n_θ are fixed, the number of frames to decrease the distance from a certain infeasible schedule to a feasible schedule (i.e., t in Definition III.4) is uniformly bounded by a finite number of frames, and depends only on the scheduling algorithm, which we denote by T_{II} . For simplicity, we use T to refer to T_{II} here in this proof.

Recall that the proof of Theorem III.1(i) implies that for the current frame i , we have

$$\Pr\left\{D(C[i+T], C^*) = D(C[i], C^*) - 1\right\} \geq (1/F)^{Tn_\theta},$$

i.e., over an interval of two frames, with at least probability $q = (1/F)^{Tn_\theta}$, $D(C[i], C^*)$ decreases by 1, since in the worst cast, each transmission was unsuccessful, and a time-slot is randomly chosen for each transmission with probability $1/F$, and we have n_θ number of scheduled transmissions, and this happens over two frames.

This also implies that with at least probability of $q^{D(C[0], C^*)}$, the system will converge to C^* within $T \times D(C[0], C^*)$ frames. Note that convergence to a feasible schedule (possibly different from C^*) could occur much earlier, since there could be multiple feasible frame schedules.

Further, note that for *any* frame schedule C , we have

$$D(C, C^*) \geq n_\theta.$$

Thus, we have for any frame schedule C , we have $\forall C \in \mathcal{C}, \forall C^* \in \mathcal{C}^*$ (where \mathcal{C} and \mathcal{C}^* is the set of all possible and feasible frame schedules, respectively),

$$\Pr\left\{\tau(C) \leq Tn_\theta\right\} \geq \Pr\left\{\tau(C) \leq TD(C, C^*)\right\} \geq q^{D(C, C^*)} \geq q^{n_\theta}.$$

Now, let us study the evolution of the frame schedule at frames $\{0, Tn_\theta, 2Tn_\theta, \dots\}$. Observe that at any frame $Tt \cdot n_\theta$ it will converge to a feasible frame schedule within up to frame $T(t+1) \cdot Tn_\theta$ with probability at least $q^{n_\theta^2}$. This immediately provides a exponential upper bound on the convergence rate, i.e., $\forall C \in \mathcal{C}$,

$$\Pr\left\{\tau(C) > tTn_\theta\right\} \leq (1 - q^{n_\theta^2})^t.$$

By letting $p = 1 - q^{n_\theta^2}$, and $K = Tn_\theta$, the result follows. ■

D. Proof of Theorem V.1

Proof of Theorem V.1(i): First, we will prove the finite sustenance condition. Note that from Rule IV.1, the time-slots for the scheduled TXs that were successful are sustained in the same position at the next frame. Clearly, if all the scheduled TXs were successful (i.e., $C[t]$ reaches a feasible FS), then the frame schedule at the current frame would be same as that at the previous frame. Thus, the finite sustenance condition is satisfied. In what follows, we focus on the proof of the finite improvement condition.

Let the current frame be t , and choose an arbitrary feasible FS C^* . Since $C[t]$ is not feasible (otherwise the result immediately follows), there exists some link l , such that the TX over l on some time-slot s is not successful at this frame t . Let us denote the set of such “unsatisfied links” by $\mathcal{L}_u[t]$. Also, denote by $\mathcal{L}_g[t]$ the set of links with “good” position w.r.t. C^* , i.e., the set of links whose slot schedules (i.e., row vector of a FS) are equal to those in C^* ⁶. We first choose such an $l \in \mathcal{L}_u \setminus \mathcal{L}_g$ if $\mathcal{L}_u \setminus \mathcal{L}_g \neq \emptyset$, and choose $l \in \mathcal{L}_g$ otherwise.

Case 1: $l \in \mathcal{L}_u \setminus \mathcal{L}_g$. In this case, we again consider two sub-cases based on whether $c_{ls}^* = 0$ or 1 (i.e., whether the

⁶For notational simplicity, we henceforth omit the frame index, $[t]$, in $\mathcal{L}_g[t]$ and $\mathcal{L}_u[t]$, unless explicitly needed.

unsuccessful TX over l on time-slot s is scheduled C^* or not).

- (i) $c_{ls}^* = 0$. Since $l \notin \mathcal{L}_g$, there is a slot $s' \neq s$, such that $c_{ls'}^* = 1$ and $c_{ls'}[t] = 0$. Note that the slot schedules of l in $C[t]$ and C^* must have the same number of 1's. Now from Property VIII.1, there is a positive probability that at frame $t+1$ we have $C[t+1]$, such that the scheduled transmission over l on slot s at frame t is moved to slot s' , and all other scheduled transmissions at frame $t+1$ are scheduled at the same slots as at frame t . Then, we have $D(C[t+1], C^*) = D(C[t], C^*) - 1$.
- (ii) $c_{ls}^* = 1$. We first let \mathcal{L}'_s denote the set of scheduled links on s by $C[t]$, but not by C^* , i.e., $\mathcal{L}'_s = \{i \in \mathcal{L} \mid c_{is} = 1, c_{is}^* = 0\}$.

Then, again there are two sub-cases: (a) there exists a unsuccessful link $l' \in \mathcal{L}'_s$, $l' \neq l$, or (b) all the scheduled links in \mathcal{L}'_s are successful on s .

(a): Similar to (i), we can move the unsuccessful TX over l' on s to a time-slot $s' \neq s$, on which a TX is scheduled by C^* , since $\mathcal{L}'_s \cap \mathcal{L}_g = \emptyset$. Then, we have $D(C[t+1], C^*) = D(C[t], C^*) - 1$.

(b): In this sub-case, we first have the following claim (the proof will be presented later):

Claim VIII.1. *Suppose that we choose a FS at frame t , such that $C[t+1] = C[t]$ (which is possible from Property VIII.1), i.e., $D(C[t+1], C^*) = D(C[t], C^*)$. Then, there exists a link $l' \in \mathcal{L}'_s[t+1]$, such that the scheduled TX over l' on s becomes unsuccessful at frame $t+1$.*

If Claim VIII.1 is true, then, at the frame $t+1$, l' corresponds to **Case 1(i)**. Thus, after two frames from t , we have $D(C[t+2], C^*) = D(C[t], C^*) - 1$.

Case 2: $l \in \mathcal{L}_g \subset \mathcal{L}_u$. Note that the fact we are in this case implies that all the links in $\mathcal{L}_u \setminus \mathcal{L}_g$ are satisfied, since we always choose first an unsatisfied link in $\mathcal{L}_u \setminus \mathcal{L}_g$ by construction.

Then, using the same definition of \mathcal{L}'_s as that in **Case 1(ii)**, this case corresponds to **Case 1(ii)(b)**, i.e., when all the links in \mathcal{L}'_s are successful on slot s . Thus, again based on Claim VIII.1, with a positive probability we have $D(C[t+2], C^*) = D(C[t], C^*) - 1$. This completes the proof. Now, it remains to prove Claim VIII.1.

Proof of Claim VIII.1: By hypothesis (i.e., **Case 1(ii)(b)**), all the scheduled links on s are successful except for l . In other words, all the links in $\mathcal{L}_g \cup \mathcal{L}'_s[t]$ are successful except for l . Note that $c_{ls}^* = 1$ and $l \in \mathcal{L}_u$. This implies that the aggregate interference by TXs over the links in \mathcal{L}_g that are scheduled on s is not enough to make the TX over l on s unsuccessful, and the TXs in $\mathcal{L}'_s[t] \setminus \{l\}$ necessarily contribute to the TX failure over l .

Also, since $C[t+1] = C[t]$ by assumption, all the scheduled links on s except for l should have low priority at frame $t+1$. Then, HPC condition implies that the high priority valid TX at stage 1 (in fact, the TX over l should be valid at stage 1) is guaranteed to be successful, there must an unsuccessful TX over a link in $\mathcal{L}'_s[t]$ at frame $t+1$. ■

Proof of Theorem V.1(ii): It is clear that with any $(\vec{Q}_s^A[t])$, where $\vec{Q}_s^A[t] \geq \vec{P}_s^A[t]$, $s = 1, \dots, F, t = 0, 1, \dots$, in element-wise, more low priority TXs will be invalidated at stage 3. Note that irrespective of the signaling power adjustment scheme, the set of high priority TXs which have data transmission is fixed. Thus, RCAMA satisfies HPC with $(\vec{Q}_s^A[t])$. This completes the proof. ■

E. Proof of Theorem V.2

To prove that RCAMA-VIR satisfies HPC subject to the condition on d_{min} , it suffices to show that for arbitrary high priority TX in \mathcal{H}_F^2 , its success is guaranteed. In this proof, we consider the case of RTS-decoding failure of a high priority TX in \mathcal{H}_F^2 . Similar proof can be applied to the case of CTS-decoding failure.

We consider a high priority TX over the link from A to B , and a set of most dominant N_{int} low priority TXs over $C_i \rightarrow D_i$, $i = 1, \dots, N_{int}$. Note that we have more high and low priority TXs in the network. Suppose that RTS from A to B fails due to the aggregate interference by RTSs from C_i , $i = 1, \dots, N_{int}$. (see Figure 5 for a similar scenario with only differences that more high and low priority TXs are in the network and N is replaced by N_{int}).

First, from (4) we have that

$$\hat{I}_{d(l)}^r \leq I_{d(l)}^r. \quad (5)$$

Note that the exact aggregate interference by RTS messages from $C_1, \dots, C_{N_{int}}$ is given by:

$$I_r^l = P \times \sum_{i=1}^{N_{int}} \frac{1}{(y_i)^{\alpha_i}}, \quad (6)$$

where α_i is the path loss exponent from C_i to B . Then, from (5) and (6), we have

$$\hat{I}_{d(l)}^r \leq I_{d(l)}^r \leq P \times \sum_{i=1}^{N_{int}} \frac{1}{(y_i)^{\alpha}}. \quad (7)$$

Now, let $y' = d(B, C')$, which is computed by B as follows:

$$(y')^{\alpha_v} = \frac{P}{\hat{I}_{d(l)}^r} \geq 1 / \sum_{i=1}^{N_{int}} \frac{1}{(y_i)^{\alpha}}, \quad (8)$$

where α_v is the path loss exponent from B to C' .

As described in the algorithm description, B assumes that there is no signal power loss between C' and D' . Based on such assumption and y' , B will adjust its CTS message power (denoted by P_c^v) enough to invalidate the CTS from D' to C' . From (8), this is given by:

$$\frac{P}{P_c^v / (y')^{\alpha_v}} \leq \gamma \Rightarrow P_c^v \geq \frac{P (y')^{\alpha_v}}{\gamma} \geq \frac{P}{\gamma \sum_{i=1}^{N_{int}} 1 / (y_i)^{\alpha}} \quad (9)$$

Let the path loss exponent from D_i to C_i be β_i . Then, the SINR value at C' for its CTS message from D' will be:

$$\frac{P / (z_i)^{\beta_i}}{Z_H + P_c^v / (y_i)^{\alpha_i}},$$

where Z_H is the total received power at C' by other high and low priority TXs except for TXs over $A \rightarrow B$ and $C_i \rightarrow D_i$, $i = 1, \dots, N_{int}$.

Then, it suffices to show that with P_c^v , all of N_{int} low priority TXs over $C_i \rightarrow D_i$, $i = 1, \dots, N_{int}$ are invalidated, i.e.,

$$\frac{P/(z_i)^{\beta_i}}{Z_H + P_c^v/(y_i)^{\alpha_i}} \leq \frac{P/(z_i)^{\beta_i}}{P_c^v/(y_i)^{\alpha_i}} = \frac{P(y_i)^{\alpha_i}}{P_c^v(z_i)^{\beta_i}} \leq \gamma. \quad (10)$$

Now, we have

$$\begin{aligned} \frac{P/(z_i)^{\beta_i}}{Z_H + P_c^v/(y_i)^{\alpha_i}} &\leq \frac{P(y_i)^{\alpha_i}}{P_c^v(z_i)^{\beta_i}} \leq \frac{P(d_{int})^{\alpha_i}}{P_c^v(d_{min})^{\beta_i}} \leq \frac{P(d_{int})^{\bar{\alpha}}}{P_c^v(d_{min})^{\alpha}} \\ &\leq \frac{\gamma(d_{int})^{\bar{\alpha}} \sum_{i=1}^{N_{int}} 1/(y_i)^{\alpha}}{(d_{min})^{\alpha}} \quad (\text{from (9)}) \\ &\leq \gamma \frac{N_{int}(d_{int})^{\bar{\alpha}}}{(d_{min})^{2\alpha}}. \end{aligned} \quad (11)$$

Thus, if $\sqrt[2\alpha]{N_{int}(d_{int})^{\bar{\alpha}}/(2\alpha)} \leq d_{min}$, we have

$$\frac{N_{int}(d_{int})^{\bar{\alpha}}}{(d_{min})^{2\alpha}} \leq 1.$$

Thus, (10) is proved, which completes the proof. ■

F. Proof of Theorem VI.1

The proof is very similar to the Proof of Theorem V.1 except for **Case 1(ii)(b)**. This is because in **Case 1(i)** and **Case 1(ii)(a)**, we rely on the idea that the scheduled TN in a bad position w.r.t. a feasible FS can be moved into a good position. Further **Case 2** entirely relies on the result in **Case 1(ii)(b)**.

Thus, it suffices to prove FIC in **Case 1(ii)(b)**, where the proof of Claim VIII.1 is the key to the proof, i.e., we need to prove Claim VIII.1 under GIM. The big difference from the proof under PIM is that HPC condition is not guaranteed under GIM.

First, we introduce a *link contention graph*, where the vertices correspond to the links in the original graph $\mathcal{G}(\mathcal{L}, \mathcal{V})$, and two vertices (say i, j) are connected if i interferes j or j interferes i . Then, we additionally define a *contention group* as a maximal clique⁷ in the link contention graph.

Recall $\mathcal{L}'_s = \{i \in \mathcal{L} \mid c_{is} = 1, c_{is}^* = 0\}$. Additionally, we let:

$$\begin{aligned} \tilde{\mathcal{L}}_s &= \{i \in \mathcal{L} \mid c_{is}[t] = 1, c_{is}^* = 1\} \\ \bar{\mathcal{L}}_s &= \{i \in \mathcal{L} \mid c_{is}[t] = 1, c_{is}^* = 1, \\ &\quad \text{TN over } l \text{ on } s \text{ are unsuccessful}\}. \end{aligned}$$

Note that the set of all links having scheduled transmissions at time-slot s at frame t is $\mathcal{L}'_s[t] \cup \tilde{\mathcal{L}}_s[t]$. Also, it is clear that $\tilde{\mathcal{L}}_s \subset \bar{\mathcal{L}}_s$, $\tilde{\mathcal{L}}_s \cap \mathcal{L}'_s = \emptyset$, and $l \in \tilde{\mathcal{L}}_s$ (recall that we are in **Case 1(ii)(b)**).

Then, all the links in $\tilde{\mathcal{L}}_s$ (thus, also $\bar{\mathcal{L}}_s$) are in the *different contention groups*, since C^* is feasible, at most only one link in the same contention group has to be scheduled at the same time-slot (in this case, s) in a feasible FS. Further, since the links inside $\tilde{\mathcal{L}}_s$ are interference-free (due to feasibility of C^*),

⁷A maximal clique is a complete subgraph such that if one further node were included anywhere the completeness condition would be violated.

by hypothesis that the TN over l on slot s is unsuccessful, *there exists a link $l' \in \mathcal{L}'_s$, $l' \neq l$ such that l and l' share the same contention group* (i.e., there must be an interfering link in \mathcal{L}'_s which makes the TN over l unsuccessful).

For the $l' \in \mathcal{L}'_s$, let

$$\tilde{\mathcal{L}}'_s = \{i \in \mathcal{L}'_s \cup \tilde{\mathcal{L}}_s \mid i \text{ shares a contention group with } l'\},$$

i.e., $\tilde{\mathcal{L}}'_s$ is the set of links which have a scheduled TN on slot s and share a contention group with l' . With this definition, we must have that $\tilde{\mathcal{L}}'_s \subset \bar{\mathcal{L}}_s$, because (i) all the links in \mathcal{L}'_s have successful TNs, and (ii) all the links in $\tilde{\mathcal{L}}'_s$ have unsuccessful TNs, and at most one TN can be successful within a contention group. We also note that *all the links in $\tilde{\mathcal{L}}'_s$ are from different contention groups*, since all the links in $\tilde{\mathcal{L}}_s$ ($\supset \bar{\mathcal{L}}_s \supset \tilde{\mathcal{L}}'_s$) are from different contention groups. Further, we note that $l \in \bar{\mathcal{L}}_s$, i.e., $\tilde{\mathcal{L}}'_s \neq \emptyset$.

If $C[t+1] = C[t]$ by assumption, then by Rule IV.1,

$$r_{i's}[t+1] = \begin{cases} 1 & \text{if } i \in \tilde{\mathcal{L}}_s, \text{ i.e., high priority} \\ 0 & \text{if } i \in \mathcal{L}'_s \cup (\tilde{\mathcal{L}}_s \setminus \tilde{\mathcal{L}}_s), \text{ i.e., low priority} \end{cases}$$

since all TNs in $\tilde{\mathcal{L}}_s$ were not successful, and all the other links scheduled on slot s (e.g., $\mathcal{L}'_s \cup (\tilde{\mathcal{L}}_s \setminus \tilde{\mathcal{L}}_s)$) were successful. In particular, $r_{l's}[t+1] = 0$, since $l' \in \mathcal{L}'_s$.

Note that since C^* is feasible, with high priority), the TNs over $\tilde{\mathcal{L}}_s$ will have successful RTS/CTS signaling at Stage 1. In particular, high priority RTS/CTS signaling over $\tilde{\mathcal{L}}_s \setminus \tilde{\mathcal{L}}'_s$ does not affect link l' , since none of the links in $\tilde{\mathcal{L}}_s \setminus \tilde{\mathcal{L}}'_s$ shares a contention group with l' . Thus, it is enough to see the TN over $\tilde{\mathcal{L}}'_s$ to investigate success or failure of the TN over l' at frame $t+1$. Then, from Rule VI.1, l' will be unsuccessful. This completes the proof. ■

REFERENCES

- [1] Y. Yi, G. de Veciana, and S. Shakkottai, "On optimal MAC scheduling with physical interference," in *Proceedings of IEEE Infocom*, Anchorage, AK, 2007.
- [2] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling for maximum throughput in multihop radio networks," *IEEE Transactions on Automatic Control*, vol. 37, no. 12, pp. 1936–1949, December 1992.
- [3] P. Chaporkar, K. Kar, and S. Sarkar, "Throughput guarantees through maximal scheduling in wireless networks," in *Proceedings of the 43rd Annual Allerton Conference on Communication, Control and Computing*, 2005.
- [4] X. Lin and N. B. Shroff, "The impact of imperfect scheduling on cross-layer rate control in wireless networks," in *Proceedings of INFOCOM*, 2005.
- [5] X. Wu and R. Srikant, "Bounds on the capacity region of multi-hop wireless networks under distributed greedy scheduling," in *Proceedings of INFOCOM*, 2006.
- [6] G. Sharma, N. B. Shroff, and R. R. Mazumdar, "Joint congestion control and distributed scheduling for throughput guarantees in wireless networks," in *Proceedings of IEEE Infocom*, 2007.
- [7] G. Sharma, R. R. Mazumdar, and N. B. Shroff, "On the complexity of scheduling in wireless networks," in *Proceedings of MOBICOM*, 2006.
- [8] E. Modiano, D. Shah, and G. Zussman, "Maximizing throughput in wireless networks via gossiping," in *Proceedings of ACM Sigmetrics*, New York, NY, USA, 2006.
- [9] A. Eryilmaz, A. Ozdaglar, and E. Modiano, "Polynomial complexity algorithms for full utilization of multi-hop wireless networks," in *Proceedings of IEEE Infocom*, 2007.
- [10] X. Lin and S. Rasool, "Constant-time distributed scheduling policies for ad hoc wireless networks," in *Proceedings of IEEE Conference on Decision and Control*, 2006.
- [11] C. Joo and N. B. Shroff, "Performance of random access scheduling schemes in multi-hop wireless networks," in *Proceedings of IEEE Infocom*, 2007.

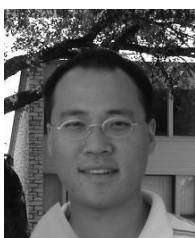
- [12] P. White, "RSVP and integrated services in the internet: A tutorial," *IEEE Communications Magazine*, May 1997.
- [13] Y. S. P. Gupta and A. Stolyar, "Random-access scheduling with service differentiation in wireless networks," in *Proceeding of INFOCOM*, March 2005.
- [14] L. Tassiulas, "Linear complexity algorithms for maximum throughput in radionetworks and input queued switches," in *Proceedings of IEEE INFOCOM*, 1998.
- [15] S. Sanghavi, L. Bui, and R. Srikant, "Distributed link scheduling with constant overhead," in *Proceedings of ACM Sigmetrics*, 2007.
- [16] Y. Yi, A. Proutiere, and M. Chiang, "Complexity-stability-delay tradeoff in scheduling over wireless network," Preprint. Available at <http://www.princeton.edu/~yyi/tradeoff.pdf>, 2007.
- [17] K. Jung and D. Shah, "Low delay scheduling in wireless network," in *Proceeding of ISIT*, 2007.
- [18] A. Behzad and I. Rubin, "Optimum integrated link scheduling and power control for ad hoc wireless networks," *IEEE Transactions on Vehicular Technology*, 2006, to appear.
- [19] R. Negi and A. Rajeswaran, "Physical layer effect on mac performance in ad-hoc wireless networks," in *In Proceedings of Communications, Internet, and Information Technology*, 2003.
- [20] P. Björklund, P. Värbrand, and D. Yuan, "Resource optimization of spatial TDMA in ad hoc radio networks: A column generation approach," in *Proceedings of IEEE INFOCOM*, 2003.
- [21] T. Moscibroda and R. Wattenhofer, "The complexity of connectivity in wireless networks," in *Proceedings of INFOCOM*, 2006.
- [22] R. Cruz and A. Santhanam, "Optimal routing, link scheduling and power control in multi-hop wireless networks," in *Proceeding of INFOCOM*, 2003.
- [23] K. Jain, J. Padhye, V. N. Padmanabhan, and L. Qiu, "Impact of interference on multi-hop wireless network performance," in *Proceedings of ACM MOBICOM*, 2003.
- [24] I. Rhee, A. Warriar, M. Aia, and J. Min, "Z-mac: a hybrid mac for wireless sensor networks," in *Proceedings of the ACM conference on Embedded networked sensor systems (SenSys)*, 2005.
- [25] M. Krunz, A. Muqattash, and S. Lee, "Transmission power control in wireless ad hoc networks: Challenges, solutions, and open issues," *IEEE Network*, vol. 18, no. 5, pp. 8–14, 2004.
- [26] S. Sarkar and L. Tassiulas, "End-to-end bandwidth guarantees through fair local spectrum shar in wireless ad-hoc networks," in *Proceeding of IEEE CDC*, 2003.
- [27] S. Ramanathan, "A unified framework and algorithm for channel assignment in wireless networks," *Wireless Networks*, vol. 5, no. 2, pp. 81–94, 1999.
- [28] IEEE Standard 802.11, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," 1997.



Gustavo de Veciana Gustavo de Veciana (S'88-M'94-SM 2001) received his B.S., M.S., and Ph.D. in electrical engineering from the University of California at Berkeley in 1987, 1990, and 1993 respectively. He is currently a Professor at the Department of Electrical and Computer Engineering at the University of Texas at Austin and the Director of the Wireless Networking and Communications Group (WNCG). His research focuses on the design, analysis and control of telecommunication networks. Current interests include: measurement, modeling and performance evaluation; wireless and sensor networks; architectures and algorithms to design reliable computing and network systems. Dr. de Veciana has been an editor for the IEEE/ACM Transactions on Networking. He is the recipient of General Motors Foundation Centennial Fellowship in Electrical Engineering and a 1996 National Science Foundation CAREER Award, co-recipient of the IEEE William McCalla Best ICCAD Paper Award for 2000, and co-recipient of the Best Paper in ACM Transactions on Design Automation of Electronic Systems, Jan 2002-2004.



Sanjay Shakkottai Sanjay Shakkottai (M'02) received his Ph.D. from the University of Illinois at Urbana-Champaign in 2002. He is currently with The University of Texas at Austin, where he is an Associate Professor in the Department of Electrical and Computer Engineering. He was the finance chair of the 2002 IEEE Computer Communications Workshop in Santa Fe, NM. He received the NSF CAREER award in 2004. His research interests include wireless and sensor networks, stochastic processes and queuing theory. His email address is shakkott@ece.utexas.edu.



Yung Yi Yung Yi received his B.S. and M.S. in the School of Computer Science and Engineering from Seoul National University, South Korea in 1997 and 1999, respectively, and his Ph.D. in the Department of Electrical and Computer Engineering at the University of Texas at Austin. Now he is a post-doctoral research associate in the Department of Electrical Engineering at Princeton University. His current research interests include scheduling and QoS for wireless networks, congestion control in the Internet, and design and performance analysis of computer networks. His email address is yyi@princeton.edu.