On Optimal MAC Scheduling With Physical Interference

Yung Yi, Gustavo de Veciana, and Sanjay Shakkottai

***Abstract*— We propose a general family of MAC scheduling algorithms that achieve any rate-point on a uniform discrete-lattice within the throughput-region (i.e., lattice-throughput-optimal) under a physical interference model. Under the physical interference model, a centralized algorithm requires information on node locations (and distance among nodes) to determine a schedule that is provably throughput-optimal. In this paper, we propose a distributed, synchronous contention-based scheduling algorithm 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.**

I. Introduction

We consider a MAC link scheduling algorithm for a time-slotted wireless ad-hoc network under a physical interference model. The MAC scheduling problem for such networks has been an active research topic during the past decades, with much of the research focusing on a graph-based interference model, where primary (i.e., one-hop) and secondary (i.e., two-hop) conflict models are considered. However, the graph-based interference model is a tremendous simplification of wireless communications mainly because it is oblivious to a realistic aggregate interference [1], [2], i.e., the interference caused by various transmitters may accumulate to eventually impede reception.

MAC scheduling algorithms under graph-based interference models can be broadly classified into: (i) distributed sub-optimal algorithms with partial throughput-guarantees (e.g., [3]–[6]), where distributed, fast algorithms achieving provable lower bounds on the throughput region have been proposed based on sub-optimal choices for a schedule on each time-slot (e.g., maximal scheduling used in switch scheduling [7]), and (ii) graph-coloring based algorithms (see [8] and references therein), where the coloring problem (which is NP-complete) is formulated by transforming the original graph into a *link contention graph*, and suboptimal polynomial centralized or distributed heuristics algorithms have been proposed.

In this paper, under the physical interference model, we propose a family of dynamic, randomized, distributed MAC scheduling algorithms and associated generalized conditions, which, if satisfied, ensure lattice-throughput-optimality, i.e., *achieving any rate-point on a uniform discrete-lattice within the throughput-region.* Note that under the physical interference model, *even a centralized scheduling algorithm requires exact topology knowledge (i.e., node locations and network connectivity) to achieve throughput-optimality*, which is needed to compute the amount of interference generated by simultaneously activated nodes. However, somewhat surprisingly, we propose a distributed algorithm that achieves lattice-throughput-optimality without centralized geographical information. To the best of our knowledge, this paper is the first attempt to propose a distributed, throughput-optimal scheduling algorithm under the physical interference model.

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 requires only a simple contention signaling on each time-slot, leading to significant simplification of implementation. In addition to being provably lattice-throughput-optimal, RCAMA operates in a “dynamic” manner, i.e., determined schedules are not necessarily conflict-free, but they are progressively improved to approach an optimal schedule (see Figure 1). Further, simple contention signaling enables RCAMA to adapt to changes in traffic load and network topology by learning a neighborhood’s contention patterns in an autonomous way.

The main motivation of 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.

There has been much recent work in the context of algorithms with provable (partial) throughput guarantees under the graph-based interference model. Examples include distributed suboptimal schemes [3]–[6], which provide lower-bounds on the throughput-region with no explicit knowledge of the offered load (i.e., statistics/knowledge of the load over any link in the network). These algorithms require either control message overheads growing with the network size [3]–[5] or constant overhead but can only realize limited throughput guarantees [6]. Clearly, their message complexities will significantly increase under the physical interference model due to the more complex interference relationships among links. Recently, the authors in [9], [10] have proposed scheduling algorithms with a 100% throughput guarantee under graph-based interference models, but they still require high control message complexity.

In contrast with the above sub-optimal, distributed ap-

Y. Yi is with the Department of Electrical Engineering, Princeton University (e-mail: yyi@princeton.edu). G. de Veciana and S. Shakkottai are with the Department of Electrical and Computer Engineering, The University of Texas at Austin (email: {gustavo, shakkott}@ece.utexas.edu). This research was supported by NSF Grants ACI-0305644, CNS-0325788 and CNS-0347400.

proaches, we assume that a node only has explicit knowledge of the *local* (long-term) offered load (i.e., offered load on each of its outgoing links). However, we are able to prove that *this extra local-information at nodes leads to a distributed, lattice-throughput-optimal algorithm that requires only three-stages of simple contention signaling (which leads to six signaling messages over one-hop) on each time-slot, irrespective of network size.* In our preliminary work [11], a family of lattice-throughput-optimal scheduling algorithms was proposed based on a graph based interference model. However, as discussed earlier, under the physical interference model, finding a throughput-optimal algorithm seems to be a much more difficult task, since the maximum number of hops that need to be considered is at most two for the graph-based interference model, whereas under the physical interference model, interference is aggregate, leading to the possibility that an arbitrary number of hops must be considered in deciding a "good" schedule.

In practice, depending on the service 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. Because the loads might exhibit some variation, or measurements might be noisy, a node might use an upper estimate for it (i.e., overbook).

Recently, there have been several efforts towards analysis and design of wireless multi-hop networks under more general interference models than the graph-based interference model.

The work of [13] develops a mathematical programming formulation for minimizing the frame size over a TDMA wireless multi-hop networks, based on the optimal joint MAC scheduling and power control under the physical interference model. This work differs from ours in that the proposed distributed, heuristic algorithm in [13] is sub-optimal, and it operates under a "relaxed physical interference model," where the interference generated by only a single interfering transmitter *closest* to the intended receiver is considered. The authors in [14], [15] also adopt this relaxed physical interference model to design and study the performance of a MAC protocol. The authors in [15] consider a mathematical optimization formulation for MAC scheduling under the physical interference model, and propose a heuristic centralized scheme with the goal of using the centralized heuristic as a benchmark of other distributed on-line algorithms.

In [16], the authors introduce a new class of interference models characterized by the parameter K, where a successful reception of a message at a receiver requires no transmission from nodes that are K-hops away from the receiver. The work of [17] defines the "scheduling complexity," which corresponds to the minimum amount of time required until a connected graph structure can be scheduled. However, they do not address the scheduling problem directly, and focus only on asymptotic analysis of scheduling complexity. In [18], [19], 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. However, in [18], [19], the MAC scheduling part is captured using an abstract fluid model, and no practical, throughput-optimal, distributed algorithm is presented.

A. Main Contributions and Organization

The main contributions of this paper are as follows:

(i) Under the physical interference model, we first propose a family of scheduling algorithms (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 (Randomized Contention-Aware Multiple Access), 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). Further, we propose an adaptive variation of RCAMA, ARCAMA (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 enough to increase performance, resulting in good adaptation to load/topology changes.

The 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 Section IV, as an instance of such a lattice-throughput-optimal family, we propose RCAMA, and discuss its variations for better adaptation to load/topology changes. Finally, in Section VI, 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 and includes a guard time corresponding to the maximum differential propagation delay between pairs of nodes in the network. 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 directional links, and a set of 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 $i{\rightarrow}j$.

The wireless system has a *single* frequency/code, which is available for both data and control message transmission, and there is no separate physical channel for control messages (i.e., in-band signaling). Each node in the system is equipped with an omni-directional antenna, and synchronized. We assume that each transmission is intended for only one receiver[1], and each node has only a single transceiver (i.e., half-duplex radio).

We denote the (fixed) power level which a transmitter uses for data transmission by P, and the *physical interference model* based on SINR (Signal-to-Interference-Noise Radio) is considered. In this interference model, a link $i \to j,\ i, j \in \mathcal{V}$ is *connected,* if $G_{ij}P/\eta_j \geq \gamma$, where 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.

A message (e.g., data, ack, or control messages) from i to j is *decodable,* if

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

where $\mathcal{V}_I$ is the set of nodes, which transmit simultaneously with i on a given time-slot.

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 define the following:

Definition 2.1: We say that a transmission over $i \to j$ 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 ack message from j will be launched only when the data message is decodable at i.

In this paper, we focus only on link-level flows, and we do not consider routing and transport-layer end-to-end flows.

B. Lattice-Throughput-Optimality: Notation and Definitions

Definition 2.2: A *link schedule* $\vec{A} = (A_l : l = 1, \ldots, |\mathcal{L}|)$, $A_l \in \{0, 1\}$ is a binary vector representing the set of links scheduled for transmission attempt on a time-slot, where $A_l = 1$ if the link l is scheduled, and 0 otherwise. We define a *reverse* of a link schedule $\vec{A}$, $\mathsf{R}(\vec{A}) = (B_l)$ to be: $B_{(j,i)} = 1$, if $A_{(i,j)} = 1$, and 0 otherwise.

Definition 2.3: A link schedule $\vec{A}$ is said to be *successful*, if the transmissions scheduled by $\vec{A}$ and $\mathsf{R}(\vec{A})$ are successful, respectively. We denote the collection of all *successful* link schedules by $\mathcal{A}$.

Definition 2.4: We define the *throughput region* Λ by:

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

Definition 2.5: For any fixed positive integer F, we define the *F-lattice-throughput region* Λ_F by:

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

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

Definition 2.6: A *scheduling algorithm* Π chooses a sequence of link schedules (which are not necessarily successful), $(\vec{A}[s] : s = 0, 1, \ldots)$, where $\vec{A}[s]$ is the link schedule on time-slot s.

Definition 2.7: For a fixed F, the offered load $\vec{\rho}$ is said to be *F-lattice-feasible* if $\vec{\rho} \in \Lambda_F$.

Definition 2.8: A scheduling algorithm Π is said to be *F-lattice-throughput-optimal*, if Π achieves 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)[2], and we will see that it is sufficient to consider such class of algorithms.

Definition 3.1: We define a *frame schedule* (FS) to be a consecutive sequence of F link schedules, i.e., a $|\mathcal{L}| \times F$ matrix, $C(F, \vec{\theta}) = (c_{ls} : l = 1, \ldots, |\mathcal{L}|, s = 1, \ldots, F)$, 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 3.2: We additionally define a *transmission priority,* $R = (r_{ls} : l = 1, \ldots, |\mathcal{L}|, s = 1, \ldots, F)$ 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 3.3: A dynamic randomized scheduling (DRS) algorithm randomly chooses a sequence of $(C[t], R[t] : t = 0, 1, \ldots)$ 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 3.1: 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

[1] In other words, we consider a “link” scheduling, not “node” scheduling.

[2] Thus, 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.

Definition 2.5. 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: (i) *FSC (Feasibility Sustenance Condition),* where if a FS converges to a feasible one, it has to be sustained thereafter, and (ii) *FIC (Finite Improvement Condition),* where before converging to a feasible FS, a sequence of FSs over frames tend to be progressively "closer" to a feasible FS with positive probability.

We first define a "distance" between two FSs (under the same topology and load), $C = (c_{ls})$ and $C' = (c'_{ls})$, to be:

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

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

Definition 3.4: For a given fixed load and topology, let the current frame to be t_i.
(1) FSC: If $C[t_i]$ is feasible, $C[t] = C[t_i]$, w.p. 1 $\forall t > t_i$.
(2) FIC: If $C[t_i]$ is not feasible, for any feasible FS $C^\star$, there is a $t < \infty$ (not dependent on $C^\star$), such that $D(C[t_i], C^\star) > D(C[t_i + t], C^\star)$ with positive probability.

Subject these two conditions, we have the following theorem:

Theorem 3.1: For any fixed feasible offered load and topology, consider a DRS algorithm Π which satisfies FSC and FIC. We have that

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

Due to space limitations, we skip the proof, which is available in [20]. The sketch of proof is as follows: First, it is easily seen that a sequence of FSs over frames forms a Markov chain. Then, FIC implies that we can construct a converging path to a feasible FS (say, $C^\star$) within a finite time, since $D(C, C^\star)$ is upper-bounded.

FIC and FSC enable us to verify throughput-optimality of a DRS algorithm. In addition, it can be customized/enhanced, and still be throughput-optimal, as long as the extended version satisfies FIC and FSC. In this paper, we develop a "base-line" DRS algorithm (RCAMA) with history 1 in Section IV, and then extend RCAMA to ARCAMA (Adaptive RCAMA) with multiple frame histories for better adaptation to load/topology changes in Section V, with both RCAMA and ARCAMA satisfying FSC and FIC.

IV. RCAMA

A. Overview

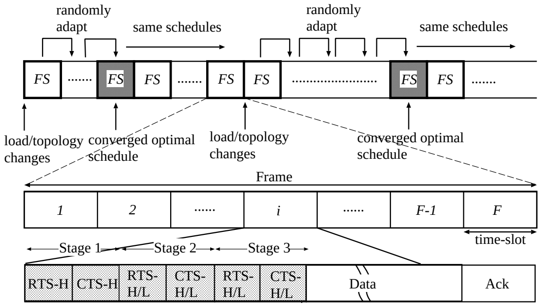


Fig. 1. Frame and slot structure of RCAMA: RTS-H/CTS-H and RTS-L/CTS-L refer to signaling messages sent by high and low priority transmitters/receivers, respectively.

The 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[3]. 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 transmissions over its adjacent outgoing links, and *(ii)* per-slot operation, where a node initiates a RTS/CTS-like contention signaling to resolve contentions and learn contention patterns in the neighborhood.

The RCAMA is designed to ensure the following two properties:

(i) **Persistence:** A successful transmission (TX) on a time-slot at the current frame persists on the same slot at the next frame.
(ii) **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 persistence property, once the system reaches a "good" (i.e., feasible) FS, it stays in that FS. Preemption property does not make a deterministic "winner-loser" relationship among TXs, and enables the system to avoid a deadlock, i.e., being stuck in a "bad" FS. These two properties ensure that the system will visit arbitrary FSs, finally reach a feasible FS, and sustain 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. Later, we will show that *it is sufficient to always ensure the success of newly incoming TXs (which was unsuccessful at the previous frame), for throughput-optimality* (see Theorem 4.1).

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

[3] For notational simplicity, we use the term 'TX' to refer to the word 'transmission' throughout this paper.

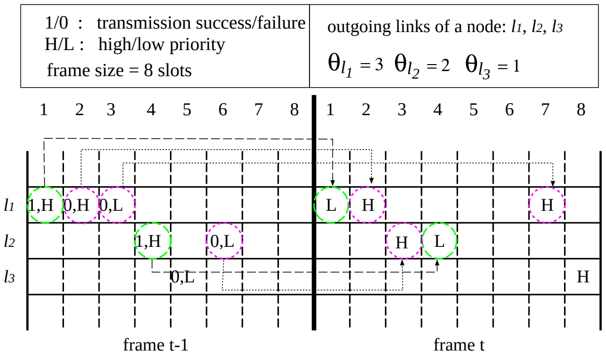


Fig. 2. Example of Rule 4.1: Since at frame $t-1$, the TX over l_1 on time-slot '1' and over l_2 on time-slot '4' were successful, these TXs are scheduled once again with *low* contention priority at the same time-slot positions at frame t. For the unsuccessful TXs over l_1 on time-slots '2' and '3', 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.

unsuccessful TXs enables the system to learn local contention levels, and to distribute scheduled TXs at different time-slots in a more efficient manner (see Section V).

We note that a similar idea of using multiple priorities has been introduced in the TDMA scheduling used in Z-MAC [21]. However, Z-MAC considers only the graph-based interference model, and its major objective of multiple priorities is to solve the hidden terminal problem without 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 4.1 (Slot and Priority Selection Rule):*
> **(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 4.1(i) corresponds to the **persistence** property. **Preemption** property is satisfied by Rule 4.1(ii) in conjunction with three-stage signaling in Section IV-C. An example of Rule 4.1 is given in Figure 2.

C. Per-Slot Operation: Three-Stage Contention Signaling

Following the determined slot-schedules 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 TX follows (see Figure 3 for the pictorial algorithm description).

Definition 4.1: A scheduled TX over $i \to j$ is said to be *valid*, if j decodes RTS from i, and i decodes 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, its data TX or ack reception could fail. For this reason, we differentiate between validity and success of a TX.

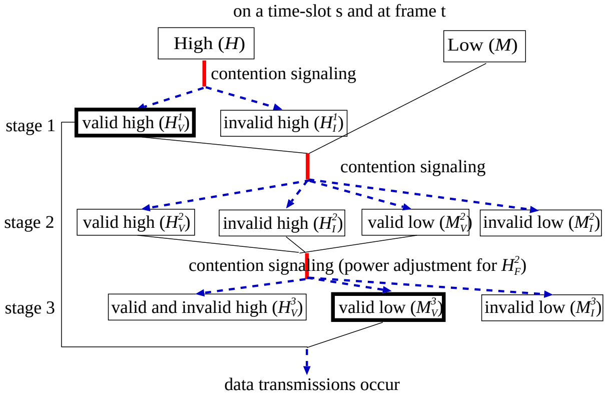


Fig. 3. Three-stage contention signaling in RCAMA

We first denote a set of links where TXs are scheduled (on this time-slot and at this frame) with high and low priority by $\mathcal{H}$ and $\mathcal{M}$, respectively, where $\mathcal{H}, \mathcal{M} \subset \mathcal{L}$, and recall that $\mathcal{L}$ is the set of all links in the network. 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 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 4.1). 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 fails due to interference by 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 **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: 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 presented in Figure 4, to show how it operates

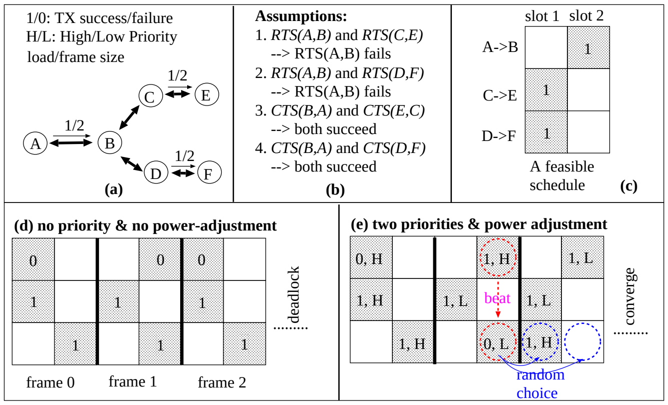


Fig. 4. Example of RCAMA: In absence of contention priority and signaling power adjustment, TX over $A \rightarrow B$ keeps failing with either choice of time-slot '1' or '2', since RTS from A is not decodable at B due to interference from either C or D, over frames. However, in RCAMA, from Rule 4.1, 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.

for convergence 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 [22] and the references therein). The approaches in [22], 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).

D. 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 $\vec{P}_s^A[t] = (\vec{P^r}, \vec{P^c})_s[t]$, $\vec{P^r} = (P_l^r), \vec{P^c} = (P_l^c), l \in \mathcal{H}_I^2(s)[t]$ 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 4.2: For any given fixed topology and load, consider a sequence of adjusted signaling power vectors, $(\vec{P}_s^A[t] : s = 1, \ldots, F, t = 0, 1, \ldots,)$. 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 4.2 corresponds to the condition that "good" high priority TXs (i.e., valid TXs at stage 1) are guaranteed to be successful. Now, Theorem 4.1 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. Due to space limitations, we skip the proof, which is available in [20].

Theorem 4.1: For any given fixed topology and load, suppose that RCAMA satisfies HPC with $(\vec{P}_s^A[t])$, then

(i) RCAMA satisfies FSC and FIC. Thus, it is throughput-optimal from Theorem 3.1.

(ii) RCAMA satisfies HPC with *any* $(\vec{Q}_s^A[t])$, where $\vec{Q}_s^A[t] \geq \vec{P}_s^A[t]$, $s = 1, \ldots, F, t = 0, 1, \ldots,$ in element-wise.

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

> **RCAMA-MAX:** All the powers in the signaling power adjustment (i.e., $\vec{P}_s^A[t]$) 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 on existence of P_{max} is reasonable for wireless multi-hop networks deployed in a finite size of plane. Now, we have the following immediate corollary:

Corollary 4.1 (RCAMA-MAX): For any fixed topology and feasible load, RCAMA-MAX satisfies HPC, and thus it is throughput-optimal.

Remark 4.1: 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 4.1 implies that there exists a distributed throughput-optimal scheduling algorithm that does not need such centralized topology information.

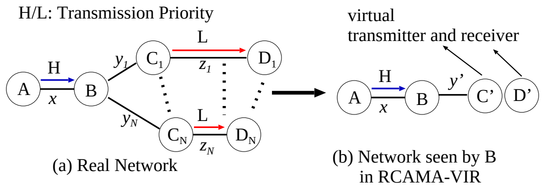
In spite of the provable throughput-optimality and the fully distributed nature of RCAMA-MAX, it may not be a practical algorithm, since for a large-scale multi-hop network, P_{max} should be very large. This is not a desirable feature due to low efficiency of energy utilization and poor transient throughput. 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.

E. RCAMA-VIR

The main idea in RCAMA-VIR is to use a sufficiently high power (but not so 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)[4], *(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 the path loss exponent (which may

[4] Note 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.



(a) Real Network

(b) Network seen by B in RCAMA-VIR

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, $i = 1, \ldots, N$. 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'.

depend on the node-pair), and each node knows just its (lower and upper) bounds (i.e., $\underline{\alpha} \leq \alpha(i, j) \leq \bar{\alpha}$), *(iii)* the system is interference-limited[5].

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

RCAMA-VIR:

(i) $d(l)$ ($s(l)$) 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 IV-F for discussion on estimation of the aggregate interference).

(ii) $d(l)$ ($s(l)$) 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).

(iii) 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 4.2).

F. 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 let the total received signal power on RTS and CTS slots at stage 1 by $d(l)$ and $s(l)$ be $\hat{R}^1_{d(l)}$ and $\hat{C}^1_{s(l)}$, respectively. Similarly, at stage 2, we use the notations $\hat{R}^2_{d(l)}$ and $\hat{C}^2_{s(l)}$. Then, we use the following to estimate the interference by low priority transmitters and receivers:

$$\hat{I}^r_{d(l)} = \hat{R}^2_{d(l)} - \hat{R}^1_{d(l)}, \qquad \hat{I}^c_{s(l)} = \hat{C}^2_{s(l)} - \hat{C}^1_{s(l)}.$$

Using the above method for estimation, we have

$$\hat{I}^r_{d(l)} \leq I^r_{d(l)}, \qquad \hat{I}^c_{s(l)} \leq I^c_{s(l)}, \tag{3}$$

where $I^r_{d(l)}$ and $I^c_{s(l)}$ are the exact aggregate low priority interference to $d(l)$ and $s(l)$, respectively. 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. The proof of (3) and more technical details are presented in [20].

Theorem 4.2 (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 $\sqrt[2\underline{\alpha}]{N_{int}}(d_{int})^{\bar{\alpha}/(2\underline{\alpha})} \leq d_{min}$, where d_{min} is the minimum distance between two nodes, then RCAMA-VIR satisfies HPC. Thus, it is throughput-optimal from Theorem 4.1.

Theorem 4.2 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. See [20] for the complete proof.

Numerical Example 4.1: 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 rage is set to be d_{min}) for different values of bounds on path-loss exponents and N_{int}, given by:

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

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} for provable guarantee.

V. 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

[5] In 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 $0 \leq \eta_j \leq \epsilon \times$ (interference), where ϵ is the ratio of thermal noise to the total interference.

changes. As shown in Proposition 5.1 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 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 $\vec{w}^l[t] = (w_s^l[t] : \;\; s = 1.\cdots, F,)$.

Proposition 5.1: For any fixed topology and feasible load and any positive integer $m < \infty$, in ARCAMA with history m, Theorems 4.1 and 4.2 still holds.

We skip the proof for brevity, since it is similar to that of RCAMA.

VI. Simulations

In this section, we evaluate the performance of RCAMA and ARCAMA algorithm 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 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.

Prior to presenting simulation results, we comment on the control overhead of the RCAMA/ARCAMA algorithm. Our approach has additional overheads as compared to a standard contention based MAC protocol (which has only one RTS/CTS signaling phase). Suppose that a MAC packet has 1000 bytes of data (note that in the 802.11 MAC, the size limit is 2312 bytes). The overhead of each RTS/CTS message pair with RCAMA is no more than 30 bytes (6 bytes each for source/destination addresses, and 3 bytes for signaling such as RTS priority level, stage, etc) will suffice for our algorithm. Thus, the additional overhead is about 30 × 2 bytes, which corresponds to approximately 6%. However, as we will show by simulation results, the performance increase is more than about 20%.

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. Figure 6(a) shows the network topology and link connectivity generated at random using the parameters above. More simulation results under various parameters and environments are available in [20].

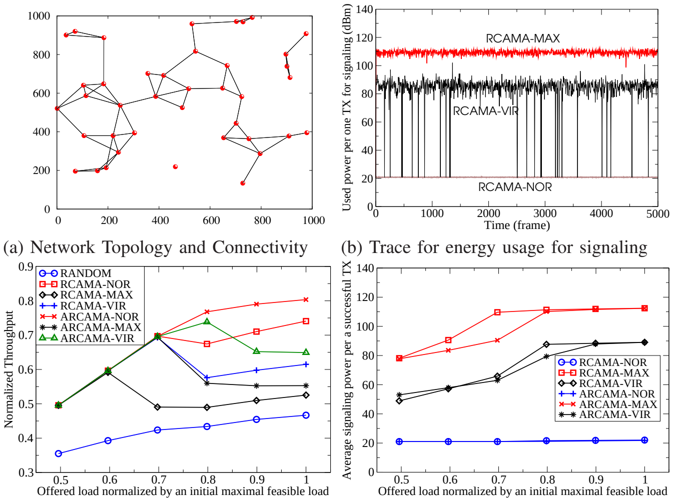
Weight Maintenance Algorithm: We use a simple weight maintenance algorithm based on three frame contention history in ARCAMA, where a slot status is defined for each outgoing links of a node, and we increase (decrease) its weight more aggressively for back-to-back failures (successes) on a slot over past three frames. We expect to see even better performance increase when more sophisticated maintenance algorithms are used. Note that the slot access probability is set to be inversely proportional to the current weight. 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 skip the details, which is described in [20].

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(c) shows the performance of RCAMA and ARCAMA algorithms for a normalized load by a randomly chosen maximally feasible load[6], 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. Figure 6(b) shows a trace of the used powers for signaling per one transmission for different RCAMA versions. 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 savings of energy.

Note that, in practice, we may need lower powers than that used by RCAMA-VIR, and the condition on d_{min} in Theorem 4.2 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 in Section VI, 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*

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



(a) Network Topology and Connectivity

(b) Trace for energy usage for signaling

(c) Throughput for different loads

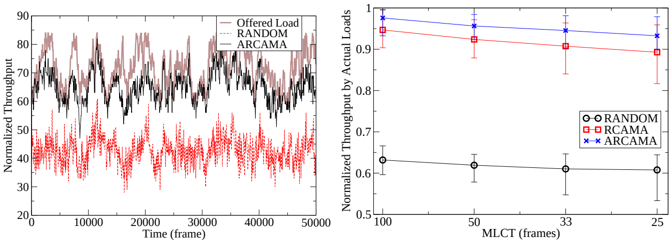
(d) Average energy per one successful TX

Fig. 6. Steady-state throughput and energy usage

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 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%.



(a) Throughput Traces: MLCT= 25 frames

(b) Normalized throughput for different MLCTs

Fig. 7. Adaptation to load changes

References

[1] A. Behzad and I. Rubin, "On the performance of graph-based scheduling algorithms for packet radio networks," in *Proceeding of IEEE Globecom*, Dec 2003.

[2] G. Zhou, T. He, J. A. Stankovic, and T. Abdelzaher, "RID: Radio interference detection in wireless sensor networks," in *Proceedings of INFOCOM*, 2005.

[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] X. Lin and S. Rasool, "Constant-time distributed scheduling policies for ad hoc wireless networks," Purdue University, Tech. Rep., 2006.

[7] J. G. Dai and B. Prabhakar, "The throughput of data switches with and without speedup." in *INFOCOM*, 2000.

[8] S. Ramanathan, "A unified framework and algorithm for channel assignment in wireless networks," *Wireless Networks*, vol. 5, no. 2, pp. 81–94, 1999.

[9] E. Modiano, D. Shah, and G. Zussman, "Maximizing throughput in wireless networks via gossiping," in *Proceedings of ACM Sigmetrics*, New York, NY, USA, 2006.

[10] A. Eryilmaz, A. Ozdaglar, and E. Modiano, "Polynomial complexity algorithms for full utilization of multi-hop wireless networks," MIT, Tech. Rep., 2006.

[11] Y. Yi, G. de Veciana, and S. Shakkottai, "Learning contention patterns and adapting to load/topology changesin in a mac scheduling algorithm," in *Proceedings of IEEE WiMesh*, 2006.

[12] P. While, "RSVP and integrated services in the internet: A tutorial," *IEEE Communications Magazine*, May 1997.

[13] 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.

[14] 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.

[15] 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.

[16] G. Sharma, R. R. Mazumdar, and N. B. Shroff, "On the complexity of scheduling in wireless networks," in *Proceedings of MOBICOM*, 2006.

[17] T. Moscibroda and R. Wattenhofer, "The complexity of connectivity in wireless networks," in *Proceedings of INFOCOM*, 2006.

[18] R. Cruz and A. Santhanam, "Optimal routing, link scheduling and power control in multi-hop wireless networks," in *Proceeding of INFOCOM*, 2003.

[19] 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.

[20] Y. Yi, G. de Veciana, and S. Shakkottai, "On optimal MAC scheduling with physical interference," University of Texas at Austin, Tech. Rep., 2006.

[21] I. Rhee, A. Warrier, 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.

[22] 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.