2016 IEEE 55th Conference on Decision and Control (CDC)
ARIA Resort & Casino
December 12-14, 2016, Las Vegas, USA

Optimal Distributed Scheduling of Real-Time Traffic with Hard Deadlines

Ning Lu, Bin Li, R. Srikant, and Lei Ying

Abstract— **In this paper, we consider optimal distributed scheduling of real-time traffic with hard deadlines in an ad hoc wireless network. Specifically, we assume the links share a common wireless channel and interference is represented by a conflict graph. Periodic single-hop traffic is considered where packets arrive at the beginning of each frame and need to be delivered by the end of the frame (otherwise, packets will be dropped). Each link is required to guarantee a maximum allowable packet dropping rate. We show that the real-time scheduling problem is combinatorial and tends to be intractable as the network size increases. To solve the real-time scheduling problem, we propose a frame-based carrier-sense multiple access (CSMA) algorithm which is shown to be asymptotically optimal. Moreover, it can be implemented in a distributed manner with low complexity. Simulation results also demonstrate the ability of our algorithm to meet the QoS requirements on deadlines.**

I. Introduction

Scheduling real-time traffic over wireless links is an important task to provide quality of service (QoS) guarantees for mission-critical systems. Consider the scenario where vehicles are platooning on highways. Vehicles in each platoon exchange up-to-date driving status (i.e., position, speed, deceleration, etc.) wirelessly to maintain a small inter-vehicle distances for fuel economy. The data of driving status needs to be delivered in a timely manner, otherwise the data becomes outdated, which could lead to disturbance in the platooning system.

Scheduling deadline-constrained packets to guarantee a minimum portion of packets to be delivered on time in wireless networks has attracted substantial research interest, driven by real-time applications. Existing research has developed approximate solutions (e.g., [1]), optimal centralized solutions ([2] and [3]), and performance guarantees of sub-optimal algorithms ([4] and [5]). In [2] and [3], an optimization framework is proposed for ad hoc wireless networks with single-hop traffic, but the solutions obtained from this framework are centralized. In [4] and [5], the performance (i.e., efficiency ratio) of a sub-optimal low-complexity algorithm (proposed in [6]) is analyzed for ad hoc wireless networks under general interference, channel, and packet arrival models. Existing optimal distributed algorithms are only designed for fully connected networks (downlink [6] and peer-to-peer [7]) due to the significantly reduced scheduling decision space, i.e., only one link can be scheduled for data transmission at a time.

In this paper, we consider the problem of scheduling real-time traffic with hard deadlines in an ad hoc wireless network where the interference is represented by a conflict graph. Specifically, we consider the case where all links share one communication channel and channel access time is in frames, each of which consists of a fixed number of successive time-slots of equal duration. For each link, packets periodically arrive at the beginning of each frame and need to be delivered by the end of the frame (otherwise, the packets will be dropped). Each link needs to guarantee a maximum allowable packet dropping rate. The resulting real-time scheduling problem is combinatorial and tends to be intractable as the network size goes large [8]. The objective of this paper is to design an optimal algorithm which solves the underlying combinatorial optimization problem, and most importantly is amenable to low-complexity distributed implementation.

The main contributions of this paper are highlighted as follows.

- We show that the considered real-time scheduling problem is essentially a Maximum Weighted k-independent set (MWKIS) problem, where k independent sets are not necessarily disjoint (see discussions in Section III). For a fully connected network, a greedy algorithm that chooses the maximum weighted independent set (MWIS) at each time slot sequentially in a frame is optimal. However, we show that such a greedy algorithm is not an optimal solution to the MWKIS problem for ad hoc networks.
- We propose a frame-based carrier-sense multiple access (CSMA) algorithm for scheduling deadline-constrained packets. We show that by utilizing a time-reversible Markov chain the proposed algorithm is asymptotically optimal. Moreover, the frame-based CSMA is indeed low-complexity and can be implemented in a distributed manner. We note that designing a CSMA algorithm for the MWKIS problem is considerable harder than the corresponding result for the MWIS problem which has been studied previously. In particular, one of the main contributions of the paper is to design a CSMA algorithm which operates with a small state space, whereas a naive application of the traditional CSMA algorithm for the MWIS to the MWKIS problem will

Ning Lu (nlu@tru.ca) is with the Department of Computing Science, Thompson Rivers University, Kamloops, BC, Canada;
Bin Li (binli@uri.edu) is with the Department of Electrical, Computer and Biomedical Engineering, University of Rhode Island, Kingston, RI 02881, USA;
R. Srikant (rsrikant@illinois.edu) is with the Department of Electrical and Computer Engineering and the Coordinated Science Lab, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA;
Lei Ying (lei.ying.2@asu.edu) is with the School of Electrical, Computer and Energy Engineering, Arizona State University, Tempe, AZ 85287, USA.

result in a state space that is exponentially large in the number of arrivals per frame.

The remainder of the paper is organized as follows. Section II introduces the system model. Section III presents the optimal solution and shows that the greedy algorithm is not optimal generally. In Section IV, we propose the frame-based CSMA which is feasibility-optimal. Simulation results are given in Section V. Section VI provides concluding remarks.

II. System Model

We consider scheduling deadline-constrained packets in an ad hoc wireless network consisting of a set of links, denoted by $\mathcal{L} = \{1, 2, \cdots, L\}$, sharing a common communication channel. Assume that time is slotted and each consecutive T time slots are grouped into one frame. Let $A_l[k]$ denote the number of packet arrivals at link $l \in \mathcal{L}$ in frame k. We assume that packets arrive at each link only at the beginning of each frame and there is no packet arrival during the frame. Each packet has to be delivered by the end of the frame; otherwise, it will be dropped. It is required that each link guarantees a maximum allowable packet dropping rate $\gamma_l \in (0, 1)$ due to missing deadlines. Particularly, in this paper, we consider periodic traffic flow, i.e., $A_l[k] = A_l > 0$ over all frames. Denote $\mathbf{A} = \big[A_1, A_2, \cdots, A_L\big]$ and $\mathbf{\Gamma} = \big[\gamma_1, \gamma_2, \cdots, \gamma_L\big]$.

Since nearby links cannot be scheduled at the same time due to interference, scheduling is required to determine the subset of links that should transmit at each time slot. A *feasible schedule* is a set of links that can be scheduled simultaneously for transmitting packets without interfering with each other. In this paper, we use a conflict graph to model interference such that vertices of the interference graph represent wireless links and any two vertices are connected by an edge if and only if the two vertices cannot be scheduled at the same time. Under the interference graph model, finding a feasible schedule in the network is equivalent to finding an independent set in the corresponding conflict graph.

Denote by $\mathbf{S}[k] = \big(S_{lt}[k]\big)$, $l \in \mathcal{L}, t \in \{1, 2, \cdots, T\} = \mathcal{T}, S_{lt}[k] \in \{0, 1\}$, a global schedule ($L \times T$ matrix) for all links and time slots in frame k. Specifically, $S_{lt}[k] = 1$ if link l is scheduled at time slot t, and $S_{lt}[k] = 0$ otherwise. We further assume that if scheduled, each link can only transmit one packet at a time slot[1]. The set of all possible globe schedules is denoted by $\mathcal{S}$, i.e., $\mathbf{S}[k] \in \mathcal{S}, \forall k$. It is easy to verify that for any element $\mathbf{S} \in \mathcal{S}$, it needs to satisfy the following constraints: (i) $\sum_{t=1}^{T} S_{lt} \leq A_l, \forall l \in \mathcal{L}$; and (ii) for any time slot t, the set $\{l : S_{lt} = 1, \forall l \in \mathcal{L}\}$ is a feasible schedule (an independent set). For convenience, we further denote $\mathbf{S}[k]$ by $\{\mathbf{S}_{\cdot t}[k]\}$ and $\{\mathbf{S}_{l\cdot}[k]\}$, i.e., in the form of column vectors and row vectors respectively. The objective of real-time scheduling in our setting is to find $\{\mathbf{S}[k]\}_{k\geq 1}$ such that for each link the long-term time-average packet dropping rate due to missing deadlines is no more than the maximum allowable packet dropping rate.

[1]This assumption does not imply that channel conditions remain unchanged. Different modulation and coding schemes can be applied to combating channel fluctuation. What we really assume here is that each link is capable of transmitting at least one packet successfully at a time slot by using appropriate modulation and coding schemes; however, each link only transmits one packet if scheduled.

III. Optimal Global Schedule

We present an optimal centralized scheduling policy in this section, based on which we design the optimal distributed algorithm. A virtual queue technique in [9] can be used to obtain a schedule $\{\mathbf{S}^*[k]\}_{k\geq 1}$ which is optimal in the sense the requirement on packet dropping rate is fulfilled for all links.

Assume that each link l maintains a virtual queue to keep track of the number of dropped packets. Let $V_l[k]$ denote the queue length at the beginning of frame k. The dynamics of virtual queue is given by

$$V_l[k+1] = \Big(V_l[k] + D_l[k] - B_l[k]\Big)^+, \tag{1}$$

where $(\cdot)^+ = \max\{\cdot, 0\}$, and $D_l[k] \geq 0$ is the increase of the virtual queue and is equal to the number of dropped packets in frame k, i.e., $D_l[k] = A_l - \sum_{t=1}^{T} S_{lt}[k]$; and the decrease of the virtual queue is $B_l[k]$ with mean $\gamma_l A_l$, and $B_l[k] < B_{max}$ for some $B_{max} < \infty$. The requirements on packet dropping rates will be satisfied for all links if all virtual queues are *mean rate stable* (see [9] for details). According to [7], if there is a scheduling policy that can make all virtual queues mean rate stable, then there always exist non-negative numbers $\alpha(\mathbf{A}, \mathbf{\Gamma}; \mathbf{S})$ such that

$$\sum_{\mathbf{S}\in\mathcal{S}} \alpha(\mathbf{A}, \mathbf{\Gamma}; \mathbf{S}) = 1, \tag{2}$$

$$A_l(1 - \gamma_l) < \sum_{\mathbf{S}\in\mathcal{S}} \alpha(\mathbf{A}, \mathbf{\Gamma}; \mathbf{S}) \sum_{t=1}^{T} S_{lt}, \forall l. \tag{3}$$

The maximal satisfiable region is defined as follows:

$$\mathcal{C}(\mathbf{A}, \mathbf{\Gamma}) = \big\{(\mathbf{A}, \mathbf{\Gamma}) : \exists \alpha(\mathbf{A}, \mathbf{\Gamma}; \mathbf{S}) \geq 0, \text{ such that (2) and (3) hold}\big\}. \tag{4}$$

Definition 1: A scheduling algorithm is said to be *feasibility-optimal* if, for any $(\mathbf{A}, \mathbf{\Gamma}) \in \mathcal{C}(\mathbf{A}, \mathbf{\Gamma})$, it makes all virtual queues mean rate stable.

Remark: Note that the feasibility-optimal in the context of virtual queues is equivalent to the throughput-optimal in the context of data queues.

By applying the dual decomposition [10], an optimal centralized solution can be obtained as follows.

Algorithm 1 Optimal Centralized Algorithm

For any $(\mathbf{A}, \mathbf{\Gamma}) \in \mathcal{C}(\mathbf{A}, \mathbf{\Gamma})$, the global schedule in each frame k is given by

$$\mathbf{S}^*[k] \in \arg\max_{\mathbf{S}\in\mathcal{S}} \sum_{l=1}^{L} f(V_l[k]) \min\Big\{\sum_{t=1}^{T} S_{lt}[k], A_l\Big\}. \tag{5}$$

In (5), the choice of weight function $f(\cdot)$ facilitates flexible implementations. $f(\cdot)$ can be an increasing function given in [11]. For example, $f(\cdot) = \log\log(\cdot)$. We omit the proof of optimality of (5) here since it is almost the same as in [7].

It is well known that MaxWeight scheduling [12] is throughput-optimal for wireless networks with single-hop traffic, which essentially chooses the MWIS for transmission. Different from MWIS, (5) is an MWKIS problem, i.e., finding k independent sets (not necessarily disjoint) such that the total weight of their union is maximum. Both MWIS and MWKIS are combinatorial in nature and difficult to solve. Motivated by queue-length-based CSMA ([13], [14]) for solving the MWIS problem asymptotically, it is natural to consider a CSMA-based scheduling algorithm to solve the MWKIS problem. Further, if a greedy algorithm exists, which optimally solves MWKIS by solving MWIS sequentially at each time slot, we may directly apply existing CSMA algorithms for scheduling deadline-constrained packets.

To find a greedy algorithm, based on Lemma 5 in [7], we first obtain a cost-to-go form of dynamic programming for the optimal scheduler (5): at each time slot $t \in \mathcal{T}$ in frame k, given $V_l[k]$ for all $l \in \mathcal{L}$ and $\{\mathbf{S}_{\cdot\tau}[k]\}_{\tau=1}^{t-1}$, $\mathbf{S}^*_{\cdot t}[k]$ is given by

$$\mathbf{S}^*_{\cdot t}[k] \in \arg\max_{\mathbf{S}_{\cdot t}} \Bigg(\sum_{l \in \mathbf{S}_{\cdot t}} f(V_l[k]) \Bigg(A_l - \sum_{j=1}^{t-1} S_{lj}[k] \Bigg)^+ + \max_{\{\mathbf{S}_{\cdot\tau}[k]\}_{\tau=t+1}^{T}} \sum_{i=t+1}^{T} \sum_{l'=1}^{L} f(V_{l'}[k]) \cdot \Bigg(A_{l'} - \sum_{j=1}^{i-1} S_{l'j}[k] \Bigg)^+ S_{l'i}[k] \Bigg). \tag{6}$$

It can be seen that the optimal solution to (6) at each time slot depends on the future time slot. Moreover, the total weight of an independent set at a time slot may change depending on MWIS selection at previous time slots. For a fully connected network, a greedy algorithm of selecting MWIS at each time slot, which is given by

$$\mathbf{S}^G_{\cdot t}[k] \in \arg\max_{\mathbf{S}_{\cdot t}} \sum_{l \in \mathbf{S}_{\cdot t}} f(V_l[k]) \Bigg(A_l - \sum_{j=1}^{t-1} S_{lj}[k] \Bigg)^+, \ \forall t \in \mathcal{T}. \tag{7}$$

is optimal to (5) and (6), and can be implemented distributedly [3], [7]. However, such a greedy algorithm is not an optimal solution to (5) and (6) for ad hoc networks. We demonstrate this by giving a counter example to show that 'greedy does not always stay ahead'. Consider a conflict graph shown in Fig. 1. We set $T = 2$, and $A_l = 1$ for all links. Clearly, $\{l_1, l_3\}$, $\{l_1, l_5\}$, $\{l_2, l_4\}$ and $\{l_3, l_4\}$ are maximal feasible schedules. We simply let the weights associated to each link satisfy the following: $V_1[k] > V_3[k] > V_4[k] > V_5[k] > V_2[k]$. Then, by the greedy algorithm, $\{l_1, l_3\}$ is chosen for slot 1 and $\{l_2, l_4\}$ is chosen for slot 2, which yields a global schedule $\big\{\{l_1, l_3\}, \{l_2, l_4\}\big\}$ with a total weight of $V_1[k] + V_3[k] + V_2[k] + V_4[k]$. Since at most four different links can be scheduled in a frame, the optimal global schedule is $\big\{\{l_1, l_5\}, \{l_3, l_4\}\big\}$ with a total weight of $V_1[k] + V_5[k] + V_3[k] + V_4[k]$. Therefore, the greedy algorithm is not optimal in a particular time slot.

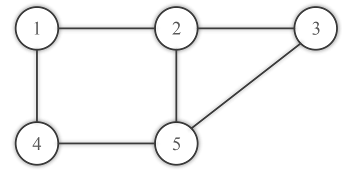

Fig. 1. Conflict graph of a 5-link network

IV. Distributed Algorithm

In this section, we propose a low-complexity CSMA-like distributed algorithm for scheduling deadline-constrained packets. We prove that our proposed algorithm solves the MWKIS problem asymptotically, and it supports the entire maximal satisfiable region, i.e., it is feasibility-optimal.

The idea is to construct a time-reversible Markov chain of global schedules, where the state space is $\mathcal{S}$, with an underlying stationary distribution that generates a global schedule asymptotically approaching MWKIS. The queue-length-based CSMA also resorts to a time-reversible Markov chain for solving MWIS. However, it is not straightforward for solving MWKIS due to the constraint that the number of scheduled time slots should be no more than the number of packet arrivals in a frame for each link.

We propose the frame-based CSMA algorithm as follows to distributedly generate global schedule $\mathbf{S}[k]$ in frame k based on $\mathbf{S}[k-1]$. A small control slot at the beginning of each frame is required to run the scheduling algorithm.

In Algorithm 2, a decision schedule $\mathbf{M}[k]$ is an independent set of non-interfering links in frame k. $\mathcal{M}$ denotes the set of all decision schedules. Refer to [14] for details of how to generate a decision schedule in a distributed manner. $F_l[t]$ indicates whether link l is allowed to change transmission status (i.e, transmit or not transmit) at time slot t. X_l denotes the number of time slots that are allowed to change transmission status in frame k based on the previous frame $k-1$. U_l denotes the maximum number of time slots that will be scheduled to link l depending on X_l and A_l. W_l is the number of time slots scheduled to link l in frame k.

Proposition 1: $\{\mathbf{S}[k]\}_{k \geq 1}$ is an irreducible, aperiodic, and finite-state Markov chain. The stationary distribution of $\{\mathbf{S}[k]\}_{k \geq 1}$ is given by

$$\begin{aligned} \pi(\mathbf{S}) &= \frac{1}{Z} \exp\Bigg(\sum_{l=1}^{L} \sum_{t=1}^{T} S_{lt} f(V_l[k]) \Bigg) \\ &= \frac{1}{Z} \exp\Bigg(\sum_{l=1}^{L} Y_l(\mathbf{S}) f(V_l[k]) \Bigg), \quad \forall \mathbf{S} \in \mathcal{S}, \end{aligned} \tag{8}$$

where $Y_l(\mathbf{S}[k]) \triangleq \sum_{t=1}^{T} S_{lt}[k]$ (i.e., the number of time slots scheduled to link l in frame k) and Z is a normalization factor, i.e., $Z = \sum_{\mathbf{S} \in \mathcal{S}} \exp\big(\sum_{l=1}^{L} Y_l(\mathbf{S}) f(V_l[k]) \big)$.

Algorithm 2 Frame-based CSMA (in control slot of frame k)

1. Choose a decision schedule $\mathbf{M}[k]$ randomly from $\mathcal{M}$
2. $\forall l \notin \mathbf{M}[k]$, $S_{lt}[k] = S_{lt}[k-1], \forall t \in \mathcal{T}$
3. $\forall l \in \mathbf{M}[k]$,
   $F_l[t] = 0, \forall t \in \mathcal{T}$
   Let $\mathcal{N}_l$ be the set of links interfering with l
   for $t = 1$ **to** T **do**
     if $\exists l' \in \mathcal{N}_l$ such that $S_{l't}[k-1] = 1$ **then**
       $S_{lt}[k] = S_{lt}[k-1]$
     else
       $F_l[t] = 1$
     end if
   end for
   $X_l(\mathbf{S}[k-1]) \triangleq \sum_{t=1}^{T} F_l[t]$
   $U_l(\mathbf{S}[k-1]) \triangleq \min(X_l(\mathbf{S}[k-1]), A_l)$
   Let $h(x,y,z) \triangleq \binom{x}{y} e^{yz}$, $x, y \in \mathbb{N}$, $y \le x$, and $z \ge 0$
   W_l randomly takes integer values in $[0, U_l(\mathbf{S}[k-1])]$ with probability distribution
$$P(W_l = w) = \frac{h(X_l(\mathbf{S}[k-1]), w, f(V_l[k]))}{\sum_{x=0}^{U_l(\mathbf{S}[k-1])} h(X_l(\mathbf{S}[k-1]), x, f(V_l[k]))}, \quad w = 0,1,2,\cdots,U_l(\mathbf{S}[k-1])$$
   Randomly and uniformly select W_l distinct slot indices in $\{t : t \in \mathcal{T}, F_l[t] = 1\}$, i.e., $t_1, t_2, \cdots, t_{W_l}$
   Set $S_{lt_i}[k] = 1$ for all $i = 1, 2, \cdots, W_l$
4. All links transmit according to $\mathbf{S}[k] = (S_{lt}[k])$ in frame k

Proof: Under Algorithm 2, $\{\mathbf{S}[k]\}_{k\ge1}$ is a discrete-time Markov chain since for any $k \ge 1$, global schedule $\mathbf{S}[k]$ only depends on $\mathbf{S}[k-1]$ and the decision schedule $\mathbf{M}[k]$. We can verify that $\{\mathbf{S}[k]\}_{k\ge1}$ is irreducible since every state can communicate with state $\mathbf{S} = \mathbf{0}$, and is aperiodic since state $\mathbf{0}$ is aperiodic. Therefore, $\{\mathbf{S}[k]\}_{k\ge1}$ has a unique stationary distribution [10]. Next, we will show that if a global schedule $\mathbf{S}$ can make a transition to another global schedule $\mathbf{S}'$, the stationary distribution given in (8) indeed satisfies the local balance equation. A transition from state $\mathbf{S}$ to state $\mathbf{S}'$ is possible if there exists a decision schedule $\mathbf{M} \in \mathcal{M}$ such that it can make this transition happen. A procedure to find such a decision schedule $\mathbf{M}$ is given in Procedure 3.

Procedure 3 Find a decision schedule for $\mathbf{S} \to \mathbf{S}'$

   Initialization: $\mathbf{M} = \emptyset$
   if $\mathbf{S}_{l\cdot} \neq \mathbf{S}'_{l\cdot}, \forall l \in \mathcal{L}$ **then**
     Add l to $\mathbf{M}$
   end if
   if $(\cup_{l\in\mathbf{M}} \mathcal{N}_l) \cap \mathbf{M} \neq \emptyset$ **then**
     return $\mathbf{M} = \emptyset$
   end if
   $\mathcal{R} \triangleq \mathcal{L} \setminus \cup_{l\in\mathbf{M}} (\mathcal{N}_l \cup \{l\})$
   $l \in \mathcal{R}$ can be in $\mathbf{M}$ arbitrarily as long as any $l' \in \mathcal{R} \cap \mathcal{N}_l$ is not added to $\mathbf{M}$
   return $\mathbf{M}$ {If $\mathbf{M} = \emptyset$, $\mathbf{S}$ cannot transit to $\mathbf{S}'$ directly}

Note that Procedure 3 is for the purpose of proving Proposition 1. Denote by $\mathcal{M}_{\mathbf{S}\to\mathbf{S}'}$ the set of all $\mathbf{M}$ generated by this procedure. It is easy to verify the following: (i) there does not exist $\mathbf{M} \notin \mathcal{M}_{\mathbf{S}\to\mathbf{S}'}$ that can make a transition from $\mathbf{S}$ to $\mathbf{S}'$; (ii) clearly, $\mathcal{M}_{\mathbf{S}\to\mathbf{S}'} = \mathcal{M}_{\mathbf{S}'\to\mathbf{S}}$; and (iii) $X_l(\mathbf{S}) = X_l(\mathbf{S}')$ if $l \in \mathbf{M}$ since $\forall l' \in \mathcal{N}_l$, $\mathbf{S}_{l'} = \mathbf{S}'_{l'}$. Note that $X_l(\mathbf{S})$ is the number of slots which are allowed to change the transmission state if link l is in a decision schedule.

There are two cases to make a transition from $\mathbf{S}$ to $\mathbf{S}'$.

1) For $l \in \mathbf{M}$, $\mathbf{S}_l$ transits to $\mathbf{S}'_l$ with probability
$$\frac{1}{\binom{X_l(\mathbf{S})}{Y_l(\mathbf{S}')}} \frac{h(X_l(\mathbf{S}), Y_l(\mathbf{S}'), f(V_l[k]))}{\sum_{x=0}^{U_l(\mathbf{S})} h(X_l(\mathbf{S}), x, f(V_l[k]))}. \tag{9}$$
2) For $l \in \mathcal{L}\setminus\mathbf{M}$, it has to keep the transmission status for the entire frame, which occurs with probability one.

Let $q(\mathbf{M})$ denote the arbitrary probability that the decision schedule $\mathbf{M}$ is chosen, and $\sum_{\mathbf{M}\in\mathcal{M}} q(\mathbf{M}) = 1$. Next, we plug in (8) and check the local balance equation for a transition from $\mathbf{S}$ to $\mathbf{S}'$:

$$\begin{aligned}
&\pi(\mathbf{S})P(\mathbf{S}\to\mathbf{S}') \\
&= \pi(\mathbf{S}) \sum_{\mathbf{M}\in\mathcal{M}_{\mathbf{S}\to\mathbf{S}'}} q(\mathbf{M}) \prod_{l\in\mathbf{M}} P(\mathbf{S}_{l\cdot}\to\mathbf{S}'_{l\cdot}) \\
&= \frac{1}{Z}\prod_{l\notin\mathbf{M}} e^{Y_l(\mathbf{S})f(V_l[k])} \sum_{\mathbf{M}\in\mathcal{M}_{\mathbf{S}\to\mathbf{S}'}} q(\mathbf{M}) \\
&\qquad \prod_{l\in\mathbf{M}} P(\mathbf{S}_{l\cdot}\to\mathbf{S}'_{l\cdot}) e^{Y_l(\mathbf{S})f(V_l[k])} \\
&= \frac{1}{Z}\prod_{l\notin\mathbf{M}} e^{Y_l(\mathbf{S})f(V_l[k])} \sum_{\mathbf{M}\in\mathcal{M}_{\mathbf{S}\to\mathbf{S}'}} q(\mathbf{M}) \\
&\qquad \prod_{l\in\mathbf{M}} \frac{1}{\binom{X_l(\mathbf{S})}{Y_l(\mathbf{S}')}} \frac{\binom{X_l(\mathbf{S})}{Y_l(\mathbf{S}')} e^{Y_l(\mathbf{S}')f(V_l[k])}}{\sum_{x=0}^{U_l(\mathbf{S})} \binom{X_l(\mathbf{S})}{x} e^{xf(V_l[k])}} e^{Y_l(\mathbf{S})f(V_l[k])} \\
&= \frac{1}{Z}\prod_{l\notin\mathbf{M}} e^{Y_l(\mathbf{S})f(V_l[k])} \sum_{\mathbf{M}\in\mathcal{M}_{\mathbf{S}\to\mathbf{S}'}} q(\mathbf{M}) \\
&\qquad \prod_{l\in\mathbf{M}} \frac{e^{(Y_l(\mathbf{S}')+Y_l(\mathbf{S}))f(V_l[k])}}{\sum_{x=0}^{U_l(\mathbf{S})} \binom{X_l(\mathbf{S})}{x} e^{xf(V_l[k])}} \\
&= \pi(\mathbf{S}')P(\mathbf{S}'\to\mathbf{S}),
\end{aligned} \tag{10}$$

where the first equality is due to the fact that each link $l \in \mathbf{M}$ makes the transition from $\mathbf{S}$ to $\mathbf{S}'$ independently from each other, and the last equality is due to the fact that $X_l(\mathbf{S}) = X_l(\mathbf{S}')$ and $U_l(\mathbf{S}) = U_l(\mathbf{S}')$ when making the transition from $\mathbf{S}'$ to $\mathbf{S}$. ■

Remark: Note that $(S_{lt}[k] : l \in \mathbf{M}[k], t \in \mathcal{T}, F_l[t] = 1)$ is a discrete-time Markov Chain conditional on $\mathbf{S}[k-1]$, for all k. Fig. 2 shows an example of how this Markov chain transits between states. Since the link weight may change over time according to the virtual queue length, the underlying Markov chain of $\mathbf{S}[k]$ is assumed to converge to its steady state before the link weight changes. This is the so-called timescale separation assumption in establishing the optimality of the queue-length-based CSMA. Although by

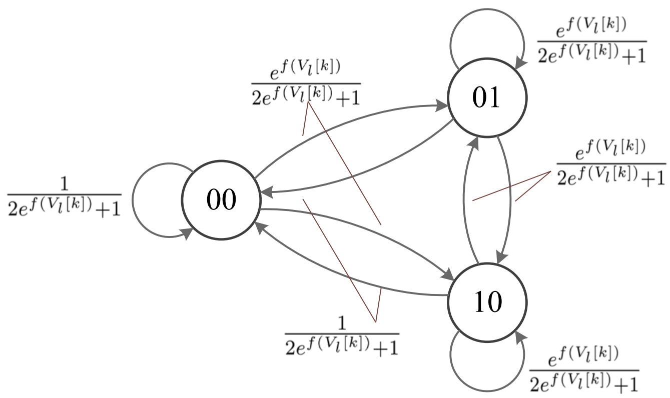

Fig. 2. Transitions between states $\{00, 01, 10\}$ for link $l \in \mathbf{M}$, given $A_l = 1, X_l(\mathbf{S}[k-1]) = 2$ (hence $U_l(\mathbf{S}[k-1]) = 1$).

appropriately choosing the weight function f the timescale separation assumption can be relaxed [11], we still rely on this assumption to establish the optimality of the frame-based CSMA scheduling for simplification.

Theorem 1: Frame-based CSMA algorithm is asymptotically optimal to (5) and (6), i.e., given any $\epsilon > 0$, $\delta < 1$, there exists a constant $C > 0$, whenever $\zeta(\mathbf{S}^*[k]) > C$ in frame k, we have

$$P\left\{\zeta(\mathbf{S}[k]) > (1-\epsilon)\zeta(\mathbf{S}^*[k])\right\} > 1-\delta, \tag{11}$$

where $\zeta(\mathbf{S}[k]) = \sum_{l=1}^{L} Y_l(\mathbf{S}[k]) f(V_l[k])$ is the weight associated with schedule $\mathbf{S}[k]$ and $\zeta(\mathbf{S}^*[k]) = \max_{\mathbf{S}[k]\in\mathcal{S}} \zeta(\mathbf{S}[k])$. Then, the frame-based CSMA algorithm is feasibility-optimal.

Proof: We denote $\mathcal{I}$ the set of schedules whose weights are less than $(1-\epsilon)\zeta(\mathbf{S}^*[k])$, i.e.,

$$\mathcal{I} = \left\{\mathbf{S}[k] \in \mathcal{S} : \zeta(\mathbf{S}[k]) < (1-\epsilon)\zeta(\mathbf{S}^*[k])\right\}. \tag{12}$$

From Proposition 1, we have

$$\begin{aligned} &P\left\{\zeta(\mathbf{S}[k]) < (1-\epsilon)\zeta(\mathbf{S}^*[k])\right\} \\ &\quad = \sum_{\mathbf{S}[k]\in\mathcal{I}} \pi(\mathbf{S}[k]) = \sum_{\mathbf{S}[k]\in\mathcal{I}} \frac{1}{Z}\exp\Big(\zeta(\mathbf{S}[k])\Big) \\ &\quad \leq \frac{|\mathcal{I}|}{Z}\exp\Big((1-\epsilon)\zeta(\mathbf{S}^*[k])\Big) \\ &\quad < |\mathcal{I}|\exp\Big(-\epsilon\zeta(\mathbf{S}^*[k])\Big) \\ &\quad < 2^{LT}\exp\Big(-\epsilon\zeta(\mathbf{S}^*[k])\Big), \end{aligned}$$

where the last two inequities are due to the fact that $\exp(\zeta(\mathbf{S}^*[k])) < Z$ and $|\mathcal{I}| < 2^{LT}$, respectively. Clearly, if we choose $C = \frac{1}{\epsilon}(LT\log 2 + \log\frac{1}{\delta})$, and whenever $\zeta(\mathbf{S}^*[k]) > C$, we will have (11). Since the feasibility optimality for virtual queues is equivalent to throughput optimality for data queues, according to Theorem 1 in [14], the frame-based CSMA algorithm is feasibility-optimal. ■

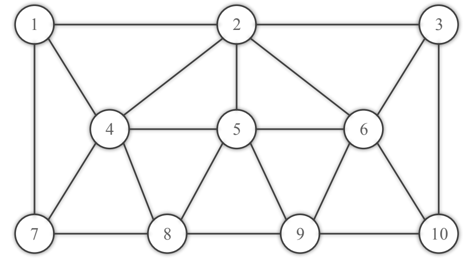

Fig. 3. Conflict graph of the simulated 10-link network

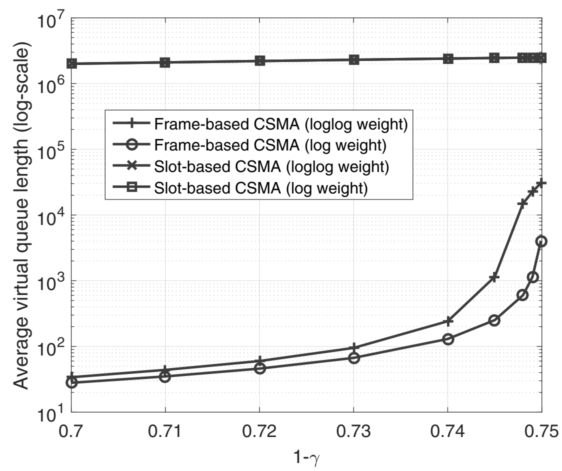

Fig. 4. Average virtual queue length vs $1-\gamma$ for the fully connected network

V. Simulation Results

In this section, we present two sets of simulations to compare the frame-based CSMA with a slot-based CSMA, which performs as the vanilla queue-length-based CSMA at each time slot with the weights given in (7).

In the first simulation, we consider a fully connected network (i.e., the conflict graph of the network is a complete graph) with $T = 15, L = 10$, and $A_l = 2, \gamma_l = \gamma$ for all l. The simulation has been run for 10^7 frames. Since at each time slot, only one link can be scheduled for transmission, at most 15 packets can be transmitted within a frame. Therefore, we have $\gamma > 0.25$. In the simulation, we examine the average virtual queue length with respect to $1-\gamma$, which can be interpreted as the QoS requirement on packet delivery ratio. We gradually push the network traffic load to the boundary of the maximal satisfiable region by increasing $1-\gamma$. The reason that we are particularly interested in the average virtual queue length is that the average virtual queue length is a good indicator of our ability to meet the QoS requirements on deadlines. A larger average virtual queue length indicates a relatively limited capability to meet the requirement. We compare both algorithms in terms of the virtual queue length averaged over all frames and all links. As shown in Fig. 4, the frame-based CSMA performs much better than the slot-based CSMA. The average virtual queue length under the slot-based CSMA goes very large even for

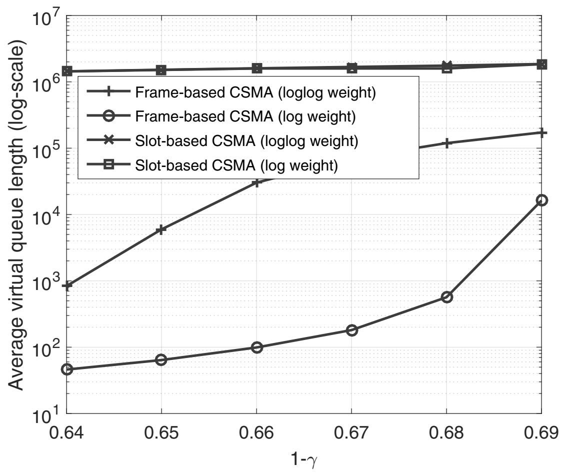

Fig. 5. Average virtual queue length vs $1-\gamma$ for the 10-link network

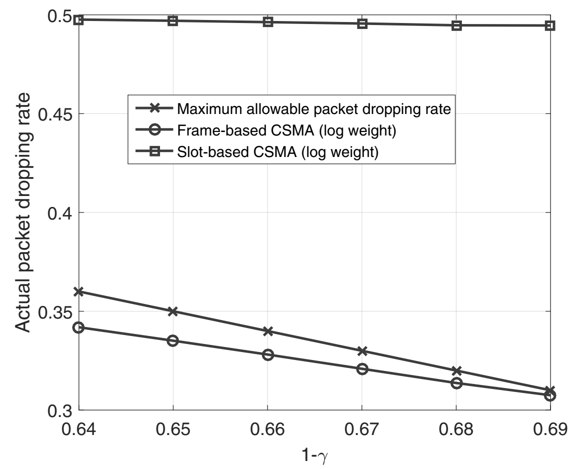

Fig. 6. Actual packet dropping rate for the 10-link network

$\gamma = 0.3$.

In the second simulation, we consider a 10-link network whose conflict graph is shown in Fig. 3. Different from fully connected networks, more than one link can be scheduled at a time slot. For example, link $1, 3, 5$ can transmit at the same time. In this case, the maximal satisfiable region is difficult to characterize due to inhomogeneous degrees of vertices in the conflict graph. We set $T = 5$, and $A_l = 2, \gamma_l = \gamma$ for all l. In this simulation, we also compare both algorithms in terms of the average virtual queue length. Similarly, as shown in Fig. 5, the frame-based CSMA outperforms the standard CSMA. Moreover, in both simulations, with a weight function f of $\log(\cdot)$, it is easier for the frame-based algorithm to achieve the maximal satisfiable region. This is because the scheduling algorithm reacts faster on virtual queue length changes with $f = \log(\cdot)$. Fig. 6 shows the actual packet dropping rate with respect to QoS requirement γ under both algorithms. It can be seen that the proposed frame-based CSMA is capable of delivering the required minimum portion of real-time traffic. While the slot-based CSMA algorithm performs much worse and cannot provide any QoS guarantees.

VI. Conclusion

In this paper, we considered the problem of scheduling real-time traffic with hard deadlines in a distributed manner for ad hoc wireless networks. We have shown that the real-time scheduling problem is essentially an MWKIS problem and an greedy algorithm of solving an MWIS at each time slot is not an optimal solution. We then proposed a frame-base CSMA algorithm which is feasibility-optimal and can be implemented distributedly. Simulations have been done to compare our algorithm to existing algorithms. In the future, we will consider the channel effect explicitly and more general traffic patterns in optimal algorithm design.

Acknowledgments

This work was supported by ARO MURI W911NF-12-1-0385, ARO Grant W911NF-16-1-0259, NSF Grant ECCS-1202065, NSF ECCS-1609202, and DTRA Grant HDTRA1-15-1-0003.

References

[1] A. Dua and N. Bambos, “Downlink wireless packet scheduling with deadlines,” *IEEE Transactions on Mobile Computing*, vol. 6, no. 12, pp. 1410–1425, 2007.

[2] J. J. Jaramillo and R. Srikant, “Optimal scheduling for fair resource allocation in ad hoc networks with elastic and inelastic traffic,” *IEEE/ACM Transactions on Networking*, vol. 4, no. 19, pp. 1125–1136, 2011.

[3] J. J. Jaramillo, R. Srikant, and L. Ying, “Scheduling for optimal rate allocation in ad hoc networks with heterogeneous delay constraints,” *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 5, pp. 979–987, 2011.

[4] X. Kang, W. Wang, J. J. Jaramillo, and L. Ying, “On the performance of largest-deficit-first for scheduling real-time traffic in wireless networks,” in *Proc. of ACM MobiHoc*, Bangalore, India, July 2013.

[5] X. Kang, I. Hou, L. Ying *et al.*, “On the capacity requirement of largest-deficit-first for scheduling real-time traffic in wireless networks,” in *Proc. of ACM MobiHoc*, Hangzhou, China, June 2015.

[6] I.-H. Hou, V. Borkar, and P. Kumar, “A theory of qos for wireless,” in *Proc. of IEEE INFOCOM*, Rio de Janeiro, Brazil, April 2009.

[7] B. Li and A. Eryilmaz, “Optimal distributed scheduling under time-varying conditions: A Fast-CSMA algorithm with applications,” *IEEE Transactions on Wireless Communications*, vol. 12, no. 7, pp. 3278–3288, 2013.

[8] M. Chen, S. C. Liew, Z. Shao, and C. Kai, “Markov approximation for combinatorial network optimization,” *IEEE Transactions on Information Theory*, vol. 59, no. 10, pp. 6301–6327, 2013.

[9] M. J. Neely, “Stochastic network optimization with application to communication and queueing systems,” *Synthesis Lectures on Communication Networks*, vol. 3, no. 1, pp. 1–211, 2010.

[10] R. Srikant and L. Ying, *Communication networks: an optimization, control, and stochastic networks perspective*. Cambridge University Press, 2013.

[11] J. Ghaderi and R. Srikant, “On the design of efficient CSMA algorithms for wireless networks,” in *Proc. of IEEE Conference on Decision and Control (CDC)*, Atlanta, Georgia, USA, December 2010.

[12] L. Tassiulas and A. Ephremides, “Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks,” *IEEE Transactions on Automatic Control*, vol. 37, no. 12, pp. 1936–1948, 1992.

[13] L. Jiang and J. Walrand, “A distributed csma algorithm for throughput and utility maximization in wireless networks,” *IEEE/ACM Transactions on Networking*, vol. 18, no. 3, pp. 960–972, 2010.

[14] J. Ni, B. Tan, and R. Srikant, “Q-CSMA: queue-length-based CSMA/CA algorithms for achieving maximum throughput and low delay in wireless networks,” *IEEE/ACM Transactions on Networking*, vol. 20, no. 3, pp. 825–836, 2012.