

Exploiting the Past to Reduce Delay in CSMA Scheduling: A High-order Markov Chain Approach

Jaewook Kwak, Chul-Ho Lee, and Do Young Eun
 Department of Electrical and Computer Engineering
 North Carolina State University
 Raleigh, NC 27695
 {jkwak, cle4, dyeun}@ncsu.edu

ABSTRACT

Recently several CSMA algorithms based on the Glauber dynamics model have been proposed for multihop wireless scheduling, as viable solutions to achieve the throughput optimality, yet are simple to implement [1, 2, 4–6]. However, their delay performances still remain unsatisfactory, mainly due to the nature of the underlying Markov chains that imposes a fundamental constraint on how the link state can evolve over time. In this paper, we propose a new approach toward better queueing and delay performance, based on our observation that the algorithm needs not be Markovian, as long as it can be implemented in a distributed manner, achieving the same throughput optimality and better delay performance. Our approach hinges upon utilizing past state information observed by local link and then constructing a high-order Markov chain for the evolution of the feasible link schedules. Our proposed algorithm, named *delayed CSMA*, adds virtually no additional overhead onto the existing CSMA-based algorithms, achieves the throughput optimality under the usual choice of link weight as a function of queue length, and also provides much better delay performance by effectively resolving temporal link starvation problem. From our extensive simulations we observe that the delay under our algorithm can be often reduced by a factor of 20 over a wide range of scenarios, compared to the standard Glauber-dynamics-based CSMA algorithm.

Categories and Subject Descriptors

C.2.1 [Network Architecture and Design]: Distributed networks, Wireless communication

General Terms

Algorithms, Performance

Keywords

CSMA scheduling, Glauber dynamics, high-order Markov chains, delay performance

1. NETWORK MODEL

We consider a wireless network with a *conflict graph* $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ where \mathcal{N} is the set of links (transmitter-receiver pair), and \mathcal{E} is the set of edges which represents conflict relationship between links. An edge $(i, j) \in \mathcal{E}$ exists between two

links i and j if simultaneous use of the two leads to failure of communications. We define a schedule by $\sigma = (\sigma_v)_{v \in \mathcal{N}} \in \{0, 1\}^{|\mathcal{N}|}$, which represents the set of transmitting links. A link v (or node v in the conflict graph \mathcal{G}) is active if it is included in the schedule, i.e., $\sigma_v = 1$, and is inactive if otherwise. A *feasible* schedule is a set of links that can be active at the same time slot according to the conflict relationship \mathcal{E} . Thus, a feasible schedule σ should satisfy the independent set constraint i.e., $\sigma_i + \sigma_j \leq 1$ for all $(i, j) \in \mathcal{E}$.

In our model, each link is associated with a queue fed by some exogenous traffic arrivals and serviced when the link is active. We consider that a packet arrives to the queue of link v at each time slot t according to a Bernoulli process $A_v(t)$, i.e., $A_v(t)$, $t = 1, 2, \dots$ are *i.i.d.* with $\mathbb{E}\{A_v(t)\} = \eta_v$. Let $\boldsymbol{\eta} = (\eta_v)_{v \in \mathcal{N}}$ be the set of arrival rates to the queues in the network. Let $Q(t) = (Q_v(t))_{v \in \mathcal{N}}$ be the number of packets in the queue at time t . Then the queue dynamics is governed by the recursion: $Q_v(t) = [Q_v(t-1) + A_v(t) - \sigma_v(t)]^+$, $t \geq 1$.

2. MAIN ALGORITHM

The basic idea of throughput-optimal CSMA is to utilize the Glauber dynamics as a link scheduling algorithm. Let $W_v(t)$ denote a weight associated with node v . Our proposed algorithm with a parameter $T \in \mathbb{N}_+$ is described in Algorithm 1.

Algorithm 1 Delayed CSMA

- 1: Initialize: for all links $i \in \mathcal{N}$, $\sigma_i(t) = 0$, $t = 0, 1, \dots, T-1$.
 - 2: At each time $t \geq T$: links find an independent set, $\mathcal{D}(t)$ through a randomized procedure*, and
 - 3: **for all** links $i \in \mathcal{D}(t)$ **do**
 - 4: **if** $\sum_{j \in \mathcal{N}_i} \sigma_j(t-T) = 0$ **then**
 - 5: $\sigma_i(t) = 1$ with probability $\frac{\exp(W_i(t))}{1 + \exp(W_i(t))}$
 - 6: $\sigma_i(t) = 0$ with probability $\frac{1}{1 + \exp(W_i(t))}$
 - 7: **else**
 - 8: $\sigma_i(t) = 0$
 - 9: **end if**
 - 10: **end for**
 - 11: **for all** links $j \notin \mathcal{D}(t)$ **do**
 - 12: $\sigma_j(t) = \sigma_j(t-T)$
 - 13: **end for**
-

*For instance, each link attempts to access the channel with an access probability a_v , $v \in \mathcal{N}$, and link v is then selected with probability $a_v \prod_{j \in \{w: (v,w) \in \mathcal{E}\}} (1 - a_j)$. More practical implementation tailored to IEEE 802.11 can be found in [4].

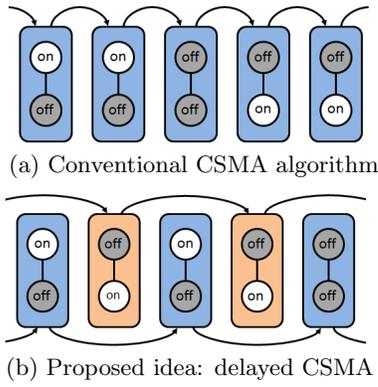


Figure 1: Comparison between the conventional CSMA and our proposed approach. A box indicates a schedule, and arrows indicate state transitions.

Note that if $T = 1$, our algorithm reduces to the conventional CSMA-based scheduling algorithm. Our motivation comes from the fact that the service process $\sigma_v(t)$ at link v under the standard CSMA policy is often heavily correlated over time. This is because once CSMA finds a schedule, it tends to stay in the same schedule or its similar set of schedules for a long time. The method we propose in this paper effectively resolves this problem. The main idea is as follows. Suppose we have two schedulers that respectively generate schedules independently, while preserving the feasibility constraint for each time slot. If we choose to use one scheduler at every odd time index, and the other one at every even time index (see Figure 1(b)), it is now possible to make a transition from ‘active-inactive’ directly to ‘inactive-active’ state, which would be impossible under the conventional CSMA (see Figure 1(a)). This alternate use of different schedulers produces more drastic change of states in consecutive time slots, thereby alleviating link starvation while maintaining the same long-term frequency of being active. Our idea is to generalize this concept by the use of multiple schedulers in a round-robin manner. In practice, this can be easily implemented in a distributed setting by having all links together update their schedules based on T -step-back state. For this purpose, each link only needs to remember its last T channel states. This way, the whole system behaves *as if* there are T separate schedulers (or chains) taking turns to generate next schedules.

3. THROUGHPUT OPTIMALITY

In our algorithm, we set $W_v(t)$ to be a function of the current queue length $Q_v(t)$ at time t , rather than T steps ago, such that the system can react more quickly by adjusting the parameter with the latest information. With the time-varying parameters, the state transition matrix becomes time-inhomogeneous, which we write as P_t , a function of $Q_v(t)$, $v \in \mathcal{N}$ at time t . For each given such P_t , let π_t be its unique stationary distribution (in a row vector form), i.e., $\pi_t = \pi_t P_t$, and μ_t be the actual distribution of the link schedules $\sigma(t)$ at time t under our algorithm with order parameter T . Then, we have $\mu_t = \mu_{t-T} P_{t-1}$. Similar to the steps via ‘network adiabatic’ theorem in [5, 6], a key step involved toward the throughput optimality is to show $\mu_t \approx \pi_t$ for sufficiently large queue lengths under suitably chosen weight functions. Note that under our algorithm, the speed of convergence of μ_t is roughly T times slower, how-

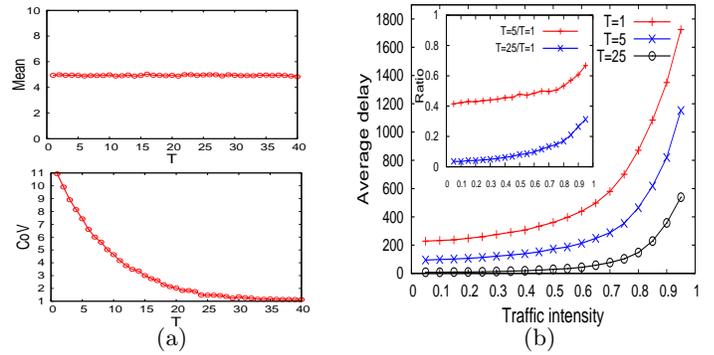


Figure 2: Impact of order parameter T : (a) Mean and CoV of ‘off’ duration. (b) Delay performance.

ever since T is finite, we expect that μ_t is still able to catch up the slowly varying target π_t in time. We verified that this is indeed the case, and see [3] for the detailed proof.

4. SIMULATION RESULTS

To evaluate the performance of the delayed CSMA algorithm, we have run simulations in a random network scenario with 25 nodes (The detailed simulation setup is given in [3]). To understand the benefits of having more drastic changes in the service process of a link under the delayed CSMA algorithm we look at the distribution of its ‘off’ duration U , the duration from an active slot to the next active slot. We measured its mean $\mathbb{E}\{U\}$ and the coefficient of variation (CoV) $\sqrt{\text{Var}\{U\}}/\mathbb{E}\{U\}$ as we increase the order parameter T . Figure 2(a) shows that the first-order statistics of the link state doesn’t change with T , while its variability decreases for larger T , implying that our algorithm with larger T effectively removes link starvation. The Figure 2(b) shows the delay improvement over the conventional CSMA algorithm under different traffic intensity, where the inset figure displays the ratio of the delay under our algorithm with chosen T to that of standard CSMA. We note that the performance improvement is quite remarkable. For instance, with $T = 5$, the delay reduces by half, and with $T = 25$, it reduces by a factor of 20 compared to the conventional CSMA algorithm.

5. REFERENCES

- [1] J. Ghaderi and R. Srikant. On the design of efficient csma algorithms for wireless networks. arXiv report <http://arxiv.org/abs/1003.1364>, arXiv, 2010.
- [2] L. Jiang, M. Leconte, J. Ni, R. Srikant, and J. Walrand. Fast mixing of parallel glauber dynamics and low-delay csma scheduling. *IEEE Trans. on Information Theory*, 2012.
- [3] J. Kwak, C.-H. Lee, and D. Y. Eun. Exploiting the past to reduce delay in csma scheduling: A high-order markov chain approach. arXiv report <http://arxiv.org/abs/1302.3250>, arXiv, 2013.
- [4] 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 Trans. on Networking*.
- [5] S. Rajagopalan, D. Shah, and J. Shin. Network adiabatic theorem: An efficient randomized protocol for contention resolution. In *ACM SIGMETRICS/Performance*, Seattle, WA, June 2009.
- [6] D. Shah and J. Shin. Randomized scheduling algorithm for queueing networks. *Annals of Applied Probability*, 2012.