Delay Optimal Queue-based CSMA

Devavrat Shah
Massachusetts Institute of Technology
Cambridge, MA 02139, USA
devavrat@mit.edu

Jinwoo Shin*
Massachusetts Institute of Technology
Cambridge, MA 02139, USA
jinwoos@mit.edu

ABSTRACT

In the past year or so, an exciting progress has led to throughput optimal design of CSMA-based algorithms for wireless networks ([1][4][2][3]). However, such an algorithm suffers from very poor delay performance. A recent work [6] suggests that it is impossible to design a CSMA-like simple algorithm that is throughput optimal and induces low delay for any wireless network. However, wireless networks arising in practice are formed by nodes placed, possibly arbitrarily, in some geographic area.

In this paper, we propose a CSMA algorithm with per-node average-delay bounded by a constant, independent of the network size, when the network has geometry (precisely, polynomial growth structure) that is present in *any* practical wireless network. Two novel features of our algorithm, crucial for its performance, are (a) choice of access probabilities as an appropriate function of queue-sizes, and (b) use of local network topological structures. Essentially, our algorithm is a queue-based CSMA with a minor difference that at each time instance a very small fraction of *frozen* nodes do not execute CSMA. Somewhat surprisingly, appropriate selection of such *frozen* nodes, in a distributed manner, lead to the delay optimal performance.

Categories and Subject Descriptors

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

General Terms

Algorithms, Performance, Design

Keywords

Wireless multi-access, Markov chain, Mixing time, Aloha

1. NETWORK MODEL

Our network is a collection of n queues. Each queue has a dedicated exogenous arrival process through which new work arrives in the form of unit-sized packets. Each queue can be potentially serviced at unit rate, resulting in departures of packets from it upon completion of their unit service requirement. The network will be assumed to be *single-hop*, i.e. once work leaves a queue, it leaves the network.

Let $t \in \mathbb{R}_+$ denote the (continuous) time and $\tau = \lfloor t \rfloor \in \mathbb{N}$ denote the corresponding discrete time slot. Let $Q_i(t) \in \mathbb{R}_+$ be the amount of work in the ith queue at time t.

Arrival process is assumed to be discrete-time with unit-sized packets arriving to queues, for convenience. Let $A_i(\tau)$ denote the total packets that arrive to queue i in $[0, \tau]$ with assumption that arrivals happen at the end in each time slot, i.e. arrivals in time slot τ happen at time $(\tau+1)^-$ and are equal to $A_i(\tau+1) - A_i(\tau)$ packets. For simplicity, we assume $A_i(\cdot)$ are independent Bernoulli processes with parameter λ_i. That is, $A_i(\tau+1) - A_i(\tau) \in \{0, 1\}$ and $\Pr(A_i(\tau+1) - A_i(\tau) = 1) = \lambda_i$ for all i and τ. Denote the arrival rate vector as $\lambda = [\lambda_i]_{1 \le i \le n}$.

The queues are offered service as per a continuous-time (i.e. asynchronous/non-slotted) scheduling algorithm. Each of the n queues is associated with a wireless transmission-capable device. Under any reasonable model of communication deployed in practice (e.g. 802.11 standards), transmissions of two devices may interfere with each other. Thus the scheduling constraint here is that no two devices that might interfere with each other can transmit at the same time. This can be naturally modeled as an *independent-set* constraint on a graph (called the *interference graph*) $G = (V, E)$ with $V = \{1, \ldots, n\}$ representing n nodes and $E = \{(i, j) : i \text{ and } j \text{ interfere with each other}\}$.

Interference graphs of our interest are of polynomial growth with constant rate ρ, which can be defined as follows.

Definition 1 (Graphs with Polynomial Growth). *$G = (V, E)$ is a polynomial growth graph with rate ρ if there exists a universal constant B such that for any $r \in \mathbb{N}$ and $v \in V$,*

$$|\{w \in V : d_G(w, v) \le r\}| \le B \cdot r^\rho,$$

where $d_G(u, v)$ denotes the length of the shortest path between u and v in G.

One can easily verify that the wireless interference network G in practice (i.e. in $\mathbb{R}^3$) has polynomial growth if the minimum distance of two devices is lower bounded by some constant and two far-away devices do not interfere with each other.

Let $\mathcal{N}(i) = \{j \in V : (i, j) \in E\}$ denote the neighbors of node i. We assume that if node i is transmitting, then all of its neighbors in $\mathcal{N}(i)$ can "listen" to it. Given this, let $\boldsymbol{\sigma}(t) = [\boldsymbol{\sigma}_i(t)]$ denote the collective scheduling decision at time $t \in \mathbb{R}_+$, with $\boldsymbol{\sigma}_i(t)$ being the rate at which node i is transmitting. The queueing dynamics induced under the

*All authors are with Laboratory for Information and Decision Systems, MIT.

SIGMETRICS'10, June 14–18, 2010, New York, New York, USA.
ACM 978-1-4503-0038-4/10/06.

above described model can be summarized by the following equation: for any $0 \le s < t$ and $1 \le i \le n$,

$$Q_i(t) = Q_i(s) - \int_s^t \sigma_i(y) \mathbf{1}_{\{Q_i(y)>0\}} dy + A_i(s,t),$$

where $\boldsymbol{A}(s,t) = [A_i(s,t)]$ denotes the cumulative arrival in time interval $[s,t]$ and $\mathbf{1}_{\{x\}}$ denotes the indicator function.

2. MAIN RESULT

In a nutshell, our algorithm is an asynchronous CSMA in which each wireless node adapts its medium access probabilities as function of its current queue-size. That is, a node attempts transmission in a regular, asynchronous manner: if it finds the medium busy then it does not transmit, else it transmits with probability that depends on its queue-size. In addition, at each time each node is in one of two states, *frozen* (colored red) or *unfrozen* (colored green). The frozen (i.e. red) nodes do not change their transmission state but unfrozen (i.e. green) nodes execute the queue-based CSMA mentioned above and details are given in Section 2.1.

As we shall find, the performance of such an algorithm is crucially determined by the precise choices of (a) frozen nodes and (b) queue-based access probabilities. We describe these in Sections 2.2 and 2.3, respectively. We note that while the algorithm is inspired by and similar to that in [4, 5], the choice of weights differs crucially in addition to the selection of frozen nodes.

2.1 CSMA Algorithm

As explained above, the CSMA algorithm will be executed by green or unfrozen nodes. Let $t \in \mathbb{R}_+$ denote the time and $\boldsymbol{W}(t) = [W_i(t)]$ denote the vector of weights at nodes at time t. As explained in Section 2.3, the weights $\boldsymbol{W}(t)$ will be essentially some function of the queue-sizes $\mathbf{Q}(t)$.

Each node i has an independent Exponential clock of rate 1. Upon a clock tick of node i at time t, it does the following:

- Node i "listens to" or "senses" the medium.
- If any neighbor is transmitting, i.e. medium is busy, then $\sigma_i(t^+) = 0$.
- Else (i.e. medium is free), set $\sigma_i(t^+) = 1$ with probability $\frac{\exp(W_i(t))}{1+\exp(W_i(t))}$, and $\sigma_i(t^+) = 0$ otherwise.

Note that due to the property of continuous random variables, no two clock ticks at different nodes will happen at the same time (with probability 1).

2.2 Coloring Scheme

In this section, we provide details of the randomized coloring or *freezing* decisions. They are updated regularly L time apart. That is, the decisions are made at times T_k, where $T_k = kL,\ k \in \mathbb{Z}_+$. Here we present the centralized description of this algorithm that is run at each time T_k, but one can easily find its simple distributed implementation.

Initially, all nodes are uncolored. Repeat the following until all nodes are colored by green or red:

(a) Choose an uncolored node $u \in V$ uniformly at random.

(b) Draw a random integer $R \in [1, K]$ according to a distribution described below that depends on K and parameter $\varepsilon > 0$.

(c) Color all nodes in $\{w \in V : d_G(u,w) < R\}$ as green.

(d) Color all nodes in $\{w \in V : d_G(u,w) = R\}$ as red.

Note that a node may be re-colored multiple times until the loop terminates. In above, K and ε is some constants, which will be decided later, and shall affect the performance of the algorithm. The distribution of R used in step (b) above, parameterized by K and $\varepsilon > 0$, is essentially a truncated (at K) Geometric with parameter ε:

$$\Pr[R = i] \quad = \quad \begin{cases} \varepsilon(1-\varepsilon)^{i-1} & \text{if } 1 \le i \le K \\ (1-\varepsilon)^{K-1} & \text{if } i = K \end{cases}.$$

2.3 Design of Weight

By removing red nodes from G, the graph is partitioned into connected components of green nodes. For each green node i, $Q_{\max,i}(t)$ denotes the maximum queue size at time t in the partition containing it. Based on this notation, the weight at green node i in the kth time interval $[T_k, T_{k+1})$ is defined as

$$W_i(t) \;=\; C \frac{Q_i(T_k)}{Q_{\max,i}(T_k)}, \qquad \text{for} \quad t \in [T_k, T_{k+1}),$$

where C is a constant and we use notation $0/0 = 1$. Thus, the weight of each node is updated regularly L time apart, at times $T_k,\ k \in \mathbb{Z}_+$.

Here we have assumed that every node i knows the maximum queue-size in the virtual partition that it belongs. This can be computed using a simple distributed mechanism.

2.4 Optimality: Throughput & Delay

These weight and coloring scheme with proper choices of L, C, K and ε lead the following throughput optimality and optimal delay property of the algorithm.

THEOREM 1. *Suppose $\boldsymbol{\lambda}$ is in the capacity region and G is a polynomial growth graph of rate ρ. Then, there exist constants*

$$L \triangleq L(\rho,\delta), \quad C \triangleq C(\rho,\delta), \quad K \triangleq K(\rho,\delta), \quad \varepsilon \triangleq \varepsilon(\delta),$$

such that the (appropriately defined) network Markov process is positive Harris recurrent with its unique stationary distribution $\boldsymbol{\pi}$. Further,

$$\sum_{i \in V} \mathbb{E}_{\boldsymbol{\pi}}\left[Q_i\right] = O(n).$$

3. REFERENCES

[1] L. Jiang and J. Walrand. A distributed csma algorithm for throughput and utility maximization in wireless networks. In *Proceedings of 46th Allerton Conference on Communication, Control, and Computing, Urbana-Champaign, IL*, 2008.

[2] J. Liu, Y. Yi, A. Proutiere, M. Chiang, and V. Poor. Convergence and tradeoff of utility-optimal csma. *Submitted, IEEE Communication Letters*, February 2009.

[3] J. Ni and R. Srikant. Distributed CSMA/CA algorithms for achieving maximum throughput in wireless networks. In *Proc. of Information Theory and Applications Workshop*, 2009.

[4] S. Rajagopalan, D. Shah, and J. Shin. A network adiabatic theorem: an efficient randomized protocol for contention resolution. In *ACM Sigmetrics/Performance*, 2009.

[5] D. Shah and J. Shin. Randomized scheduling algorithm for queueing networks. *CoRR*, abs/0908.3670, 2009.

[6] D. Shah, D. N. C. Tse, and J. N. Tsitsiklis. Hardness of low delay network scheduling. *Submitted to IEEE Transactions on Information Theory*, 2009.