Explicit Back-off Rates for Achieving Target Throughputs in CSMA/CA Networks

Benny Van Houdt
Dept. Mathematics and Computer Science
University of Antwerp, Belgium
benny.vanhoudt@uantwerpen.be

ABSTRACT

CSMA/CA networks have often been analyzed using a stylized model that is fully characterized by a vector of back-off rates and a conflict graph. We present an explicit formula for the unique vector of back-off rates $\vec{\nu}(\vec{\theta})$ needed to achieve any achievable throughput vector $\vec{\theta}$ provided that the network has a *chordal* conflict graph. These back-off rates are such that the back-off rate of a node only depends on its own target throughput and the target throughput of its neighbors and can be determined in a *distributed* manner. We also introduce a distributed chordal approximation algorithm for general conflict graphs which is shown (using numerical examples) to be more accurate than the Bethe approximation.

1. INTRODUCTION

An often studied model for CSMA/CA networks is the so-called *ideal* model [2, 4, 5, 6, 8, 9, 11, 12]. The ideal CSMA/CA model considers a network with a fixed set of n nodes and is fully characterized by a fixed conflict graph G and a fixed vector of back-off rates $\vec{\nu} = (\nu_1, \ldots, \nu_n)$. The conflict graph G identifies the pairs of nodes that interfere with each other, while the vector $(\nu_1, \ldots, \nu_n)$ determines the mean time that the nodes have to sense the channel idle before they are allowed to start a transmission.

One of the key assumptions of the ideal CSMA/CA model is that sensing is instantaneous, which implies that collisions cannot occur (as the probability that two nodes start transmitting at exactly the same time is zero). Another important assumption is that each of the n nodes always has packets ready for transmission, that is, the network is assumed to be saturated. Further perfect sensing and packet transmission is assumed.

While the product form solution for the steady state probabilities of the ideal CSMA/CA model has been established long ago [2] and the set Γ of achievable throughput vectors $\vec{\theta} = (\theta_1, \ldots, \theta_n)$ has been identified in [6], very few explicit results are available on how to set the back-off rates $\vec{\nu}$ to achieve a given vector $\vec{\theta} \in \Gamma$ (where Γ clearly depends on the conflict graph G). In [9] an explicit formula was presented to achieve fairness in a line network where each node interferes with its β left and right nodes. More recently, by relying on some existing results in statistical physics, explicit formulas for the back-off vector needed to achieve a given throughput vector were presented in case the conflict graph is a tree [12]. The existence of a unique vector of back-off rates for each achievable throughput vector was proven in [8].

In this paper[1] we identify a set of conflict graphs G for which simple explicit expressions can be obtained for the vector of back-off rates $\vec{\nu}$ that achieves a given throughput vector $\vec{\theta} \in \Gamma$. We show that an explicit expression can be obtained for any *chordal* conflict graph, thereby generalizing existing results for line networks and networks that have a tree as a conflict graph. These explicit expressions are such that the back-off rate ν_i of node i only depends on its own target throughput θ_i and the target throughput of its neighbors in the conflict graph G. Further we present a distributed chordal approximation for general conflict graphs that is more accurate than the Bethe approximation [12].

2. MODEL DESCRIPTION

Consider a network consisting of n nodes that is fully characterized by a vector of back-off rates $(\nu_1, \ldots, \nu_n)$ and an *undirected* conflict graph $G = (V(G), E(G))$, with $V(G) = \{1, \ldots, n\}$. A node is either active or inactive at any point in time. The conflict graph G specifies which pairs of nodes cannot be simultaneously active, that is, nodes i and j cannot be active simultaneously if and only if $(i, j) \in E(G)$. When a node becomes active, it remains active for some time before becoming inactive again. An inactive node can only become active if none of its neighbors in G are active. When a node becomes inactive it starts a back-off period. As soon as one of the neighbors of an inactive node in G becomes active, the back-off period of the inactive node is frozen and resumes when all of its neighbors are inactive again. A node becomes active when its back-off period ends.

If the duration of the active period is exponential (with mean 1) and the back-off period is exponentially distributed with mean $1/\nu_i$ for node i, it is well-known [2] that this network evolves as a reversible Markov chain on the state space

$$\Omega = \{(z_1, \ldots, z_n) \in \{0,1\}^n | z_i z_j = 0 \text{ if } (i,j) \in E(G)\},$$

where node i is active in state $(z_1, \ldots, z_n)$ if and only if $z_i = 1$. The steady state probabilities $\pi(\vec{z})$, with $\vec{z} = (z_1, \ldots, z_n)$

[1] A full version of this paper is available at [10].

SIGMETRICS '16 June 14-18, 2016, Antibes Juan-Les-Pins, France

ACM ISBN 978-1-4503-4266-7/16/06.
DOI: http://dx.doi.org/10.1145/2896377.2901482

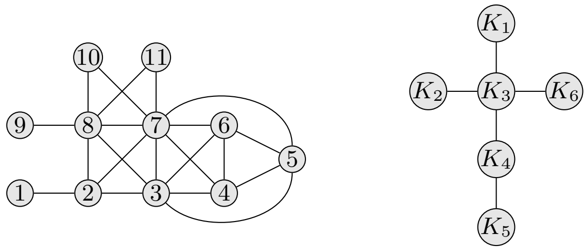

Figure 1: Example of a chordal graph G with $n = 11$ nodes (left) and one of its clique trees (right). This graph contains 6 maximal cliques $K_1 = \{1, 2\}, K_2 = \{3, 4, 5, 6, 7\}, K_3 = \{2, 3, 7, 8\}, K_4 = \{7, 8, 10\}, K_5 = \{8, 9\}$ and $K_6 = \{7, 8, 11\}$.

of this Markov chain are given by

$$\pi(\vec{z}) = \frac{1}{Z_n} \prod_{i=1}^{n} \nu_i^{z_i},$$

where $Z_n = \sum_{\vec{z}\in\Omega} \prod_{i=1}^{n} \nu_i^{z_i}$ is the normalizing constant. The throughput θ_i of node i equals $\theta_i = \sum_{\vec{z}\in\Omega, z_i=1} \pi(\vec{z})$, for $i = 1, \dots, n$. In [6] the set Γ of achievable throughput vectors $\vec{\theta} = (\theta_1, \dots, \theta_n)$ was shown to equal the interior of the convex hull of Ω and for each achievable vector $\vec{\theta} \in \Gamma$ there exists a unique vector $\vec{\nu} = (\nu_1, \dots, \nu_n)$ of back-off rates that achieves $\vec{\theta}$ [8].

3. MAIN RESULT

In this section we present an explicit formula for the back-off rates needed to achieve any achievable target throughput vector when the conflict graph G is chordal. A chordal graph $G = (V(G), E(G))$ is one in which all cycles consisting of more than 3 nodes have a *chord*. A chord of a cycle is an edge joining two nonconsecutive nodes of the cycle. A graph is chordal if and only if it has a perfect elimination ordering [1], which is an ordering of the nodes of the graph such that, for each $v \in V(G)$, v and the neighbors of v that occur after v in the order form a clique.

Let $\mathcal{K}_G = \{K_1, \dots, K_m\}$ be the set of maximal cliques of G. A clique tree $T = (\mathcal{K}_G, \mathcal{E})$ is a tree in which the nodes correspond to the maximal cliques and the edges are such that the subgraph of T induced by the maximal cliques that contain the node v is a subtree of T for any $v \in V$. An example of a chordal graph G and possible clique tree T is given in Figure 1. One can show that a graph G is chordal if and only if it has at least one clique tree (see Theorem 3.1 in [1]). A clique tree T of a chordal graph can be constructed in linear time and contains at most $V(G)$ nodes.

THEOREM 1. *Consider a network with a chordal conflict graph G. Let $\vec{\theta} = (\theta_1, \dots, \theta_n)$ be a positive vector with $T = \max_{j\in\mathcal{K}_G} \sum_{s\in K_j} \theta_s < 1$. The throughput of node i, for $i = 1, \dots, n$, in a network with conflict graph G matches θ_i if and only if the back-off rates are set as*

$$\nu_i(\vec{\theta}) = \theta_i \frac{\prod_{(K,K')\in\mathcal{E}, i\in K\cap K'} (1 - \sum_{s\in K\cap K'} \theta_s)}{\prod_{K\in\mathcal{K}_G, i\in K} (1 - \sum_{s\in K} \theta_s)}, \qquad (1)$$

for $i = 1, \dots, n$, where $T = (\mathcal{K}_G, \mathcal{E})$ is any clique tree of G.

Input: A chordal conflict graph G
Output: Back-off rates $\nu_1(\vec{\theta}), \dots, \nu_n(\vec{\theta})$

Determine a perfect elimination ordering of G
for $i = 1$ *to* n **do**
  Let $\alpha(i)$ be the node in position i in this order;
end
for $i = 1$ *to* n **do**
  Let $\mathcal{M}_{\alpha(i)} = \mathcal{N}_{\alpha(i)} \cap \{\alpha(i+1), \dots, \alpha(n)\}$;
end
$\nu_{\alpha(n)}(\vec{\theta}) = \theta_{\alpha(n)}/(1 - \theta_{\alpha(n)})$;
for $i = n - 1$ **down to** 1 **do**
  $\nu_{\alpha(i)}(\vec{\theta}) = \theta_{\alpha(i)}/(1 - \theta_{\alpha(i)} - \sum_{s\in\mathcal{M}_{\alpha(i)}} \theta_s)$;
  for $j \in \mathcal{M}_{\alpha(i)}$ **do**
    $\nu_j(\vec{\theta}) = \nu_j(\vec{\theta}) \frac{1-\sum_{s\in\mathcal{M}_{\alpha(i)}} \theta_s}{1-\theta_{\alpha(i)}-\sum_{s\in\mathcal{M}_{\alpha(i)}} \theta_s}$;
  end
end

Algorithm 1: Algorithm to determine the unique back-off rates for any $\vec{\theta} \in \Gamma$ in a network with a chordal conflict graph G.

The next property shows that the back-off rates as specified by (1) can be computed using Algorithm 1. The first step of this algorithm exists in determining a perfect elimination ordering of G, which can be achieved in $O(|V(G)| + |E(G)|)$ time using the maximum cardinality search (MCS) algorithm [7]. The MCS algorithm determines the perfect elimination ordering by picking $\alpha(n)$ at random and subsequently determines $\alpha(i)$ by selecting the node with the most neighbors in $\{\alpha(i+1), \dots, \alpha(n)\}$, breaking ties arbitrarily.

PROPOSITION 1. *The back-off rates $\nu_i(\vec{\theta})$ as computed by Algorithm 1 are equal to* (1) *irrespective of the perfect elimination ordering used.*

The back-off rates $\nu_i(\vec{\theta})$ can also be determined in a fully distributed manner with limited message passing. More specifically, it suffices for node i to discover its set of neighbors $\mathcal{N}_i$ in the conflict graph G, their target throughputs $\{\theta_j | j \in \mathcal{N}_i\}$ as well as the set of neighbors $\mathcal{N}_j$ for each $j \in \mathcal{N}_i$. With this information node i can construct the subgraph $G[\mathcal{N}_i^+]$ of the conflict graph G induced by i and its neighbors. To obtain its back-off rate $\nu_i(\vec{\theta})$ node i executes Algorithm 1 on the graph $G[\mathcal{N}_i^+]$ and sets its own back-off rate accordingly.

PROPOSITION 2. *The back-off rate for node $i \in V$ given by executing Algorithm 1 on the conflict graph G is identical to the rate obtained for node i when executing Algorithm 1 on the subgraph $G[\mathcal{N}_i^+]$ of G induced by the nodes in $\mathcal{N}_i \cup i$.*

4. CHORDAL APPROXIMATION

In this section we present a *chordal* approximation for the rates $\nu_i(\vec{\theta})$ for a general conflict graph G that is exact when the conflict graph is chordal. To study the accuracy of the proposed approximation we considered a class of conflict graphs obtained by placing a set of n nodes in a random manner in the unit square and assumed that node i and j are in conflict when the Euclidean distance between node i and j was below some threshold (see Figure 2).

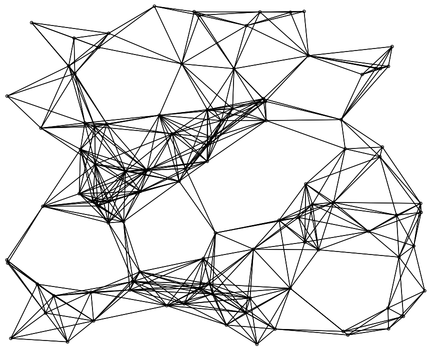

Figure 2: Non-chordal conflict graph with $n = 100$ nodes and 485 edges.

Input: A general conflict graph $G = (V, E)$
Output: A maximal chordal subgraph $\tilde{G} = (V, \tilde{E})$ and peo α

1 **for** $v \in V$ **do**
2 | $C(v) = \emptyset$;
3 **end**
4 $k = |V|$; $\tilde{E} = \emptyset$;
5 Select any $v_0 \in V$; set $S_k = \{v_0\}$; $\alpha(k) = v_0$;
6 **for** $u \in V \setminus S_k$ *with* $(u, v_0) \in E$ **do**
7 | **if** $C(u) \subseteq C(v_0)$ **then**
8 | | $C(u) = C(u) \cup \{v_0\}$;$\tilde{E} = \tilde{E} \cup (u, v_0)$;
9 | **end**
10 **end**
11 Let $v_0 \in V \setminus S_k$ with $|C(v_0)| \geq |C(v)|$ for $v \in V \setminus S_k$;
12 Set $\alpha(k-1) = v_0$; $S_{k-1} = S_k \cup \{v_0\}$; $k = k - 1$;
13 **if** $k > 1$ **then**
14 | Go to line 6;
15 **end**

Algorithm 2: MAXCHORD algorithm of [3].

The idea behind the proposed distributed chordal approximation exists in letting node i determine its back-off rate by computing a maximal chordal subgraph G_i of $G[\mathcal{N}_i^+]$ and computing its back-off rate using Algorithm 1 on G_i. To determine the subgraph G_i node i runs the MAXCHORD algorithm of [3] on $G[\mathcal{N}_i^+]$ with $v_0 = i$ in line 5 (see Algorithm 2). We refer to this approximation as the *local* chordal subgraph approximation. When the graph G is chordal this distributed algorithm corresponds to the distributed algorithm of Section 3 and is therefore exact.

Note the Bethe approximation of [12] corresponds to using the subtree consisting of the edges (i, j) with $j \in \mathcal{N}_i$, instead of the maximal chordal subgraph G_i and applying Algorithm 1 on this subtree. As such the local chordal subgraph approximation takes more conflicts into account when determining the back-off rates and it is expected to be more accurate than the Bethe approximation. This was confirmed using numerical experiments, with Figure 3 showing one particular example when using both approximations on the conflict graph of Figure 2 and setting the target throughput of each node equal to 0.04.

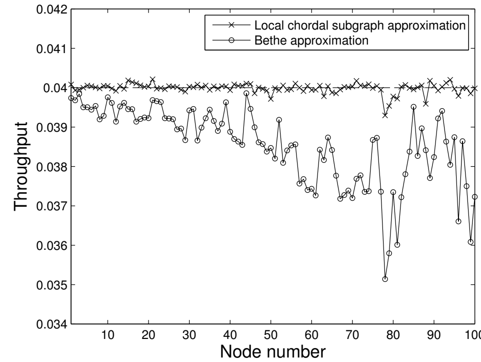

Figure 3: Throughputs of *local* chordal subgraph and Bethe approximation for the conflict graph in Figure 2 (by simulation) when the target throughput $\theta_i = 0.04$ for all i.

5. REFERENCES

[1] J. Blair and B. W. Peyton. An introduction to chordal graphs and clique trees. In *Graph Theory and Sparse Matrix Computation*, volume 56 of *The IMA Volumes in Mathematics and its Applications*, pages 1–29. Springer New York, 1993.

[2] R. Boorstyn, A. Kershenbaum, B. Maglaris, and V. Sahin. Throughput analysis in multihop CSMA packet radio networks. *IEEE Transactions on Communications*, 35(3):267–274, 1987.

[3] P. Dearing, D. Shier, and D. Warner. Maximal chordal subgraphs. *Discrete Applied Mathematics*, 20(3):181 – 190, 1988.

[4] M. Durvy, O. Dousse, and P. Thiran. On the fairness of large CSMA networks. *IEEE Journal on Selected Areas in Communications*, 27(7):1093–1104, 2009.

[5] L. Jiang, D. Shah, J. Shin, and J. Walrand. Distributed random access algorithm: scheduling and congestion control. *IEEE Trans. Inform. Theory*, 56(12):6182–6207, 2010.

[6] L. Jiang and J. Walrand. A distributed CSMA algorithm for throughput and utility maximization in wireless networks. *IEEE/ACM Transactions on Networking*, 18(3):960–972, June 2010.

[7] R. Tarjan. Maximum cardinality search and chordal graphs. Unpublished Lecture Notes CS 259, 1976.

[8] P. M. van de Ven, A. J. E. M. Janssen, J. S. H. van Leeuwaarden, and S. C. Borst. Achieving target throughputs in random-access networks. *Perform. Eval.*, 68(11):1103–1117, Nov. 2011.

[9] P. M. van de Ven, J. S. H. van Leeuwaarden, D. Denteneer, and A. J. E. M. Janssen. Spatial fairness in linear random-access networks. *Perform. Eval.*, 69(3-4):121–134, Mar. 2012.

[10] B. Van Houdt. Explicit back-off rates for achieving target throughputs in CSMA/CA networks. *CoRR*, abs/1602.08290, 2016.

[11] X. Wang and K. Kar. Throughput modeling and fairness issues in CSMA/CA based ad-hoc networks. In *Proc. of IEEE INFOCOM*, 2005.

[12] S. Yun, J. Shin, and Y. Yi. CSMA using the Bethe approximation: scheduling and utility maximization. *IEEE Transactions on Information Theory*, 61(9):4776–4787, 2015.