A New Look at Wireless Scheduling with Delayed Information

Matthew Johnston and Eytan Modiano
Laboratory for Information and Decision Systems
Massachusetts Institute of Technology
Cambridge, MA
Email: {mrj, modiano}@mit.edu

***Abstract*—The performance of wireless scheduling algorithms directly depends on the availability and accuracy of channel state information (CSI) at the scheduler. As CSI updates must propagate across the network, they are delayed as they arrive at the controller. In this paper, we analyze the effect that delayed CSI has on the throughput performance of scheduling in wireless networks. By accounting for the delays in CSI as they relate to the network topology, we revisit the comparison between centralized and distributed scheduling, which is analyzed as a trade-off between using delayed CSI and making imperfect scheduling decisions. In particular, we prove that there exist conditions under which distributed scheduling outperforms the optimal centralized scheduling policy. We characterize the point at which distributed scheduling outperforms centralized scheduling for tree networks, illustrating the impact of topology on throughput.**

I. Introduction

To achieve maximum throughput in a wireless network, a centralized controller must opportunistically schedule transmissions based on the current state of time-varying channels [1]. The channel state of a link can be measured by its adjacent nodes, who forward this channel state information (CSI) across the network to the scheduler. For example, CSI updates can be piggy-backed on top of data transmissions. Due to the transmission and propagation delays over wireless links, it may take several time-slots for the scheduler to collect CSI throughout the network, and in that time the network state may change.

There has been extensive work on wireless scheduling [1], [2], [3], although a great deal of the literature solves the optimal scheduling problem with a centralized algorithm requiring global CSI. Centralized scheduling yields high theoretical performance, since the central entity uses current network-wide CSI to compute a globally optimal schedule. However, maintaining current CSI is impractical, due to the latency in acquiring CSI throughout the network.

An alternative is a distributed approach, in which each node makes an independent transmission decision based on the CSI locally available at that node. Typically, distributed algorithms achieve only a fraction of the throughput of their ideal centralized counterparts, because they make locally optimal decisions [4]. An example distributed scheme is Greedy Maximal Scheduling [5], [6], which is known to achieve only a fraction of the centralized throughput depending on the topology. Distributed scheduling schemes that approach the centralized throughput region are proposed in [7], [8], but require higher complexity to implement. Additionally, several authors have applied random-access scheduling approaches to maximize throughput in a fully distributed manner [9], [10].

In practice, the available CSI for centralized scheduling is a delayed view of the network state. Furthermore, the delay in CSI is proportional to the distance of each link to the controller, since CSI updates often must traverse the network. These delays reduce the attainable throughput of centralized scheduling [11]. In [12], Ying and Shakkottai study throughput optimal scheduling and routing with delayed CSI. In their work, the authors assume arbitrary delays and do not consider the dependence of delay on the network topology. In contrast, by accounting for the relationship between CSI delay and network topology, we are able to effectively compare centralized and distributed scheduling.

In this paper, we propose a new model for delayed CSI, under which nodes have more accurate CSI pertaining to neighboring links, and progressively less accurate CSI for distant links. We show that as a result of the delays in CSI, in some scenarios there exist distributed scheduling algorithms that outperform the optimal centralized scheduling scheme. We develop sufficient conditions under which there exists a distributed scheduling policy that outperforms the optimal centralized policy in tree networks, illustrating the impact of topology on achievable throughput. We provide simulation results to demonstrate the performance on different topologies.

II. Model and Problem Formulation

Consider a network consisting of a set of nodes $\mathcal{N}$, and a set of links $\mathcal{L}$. Time is slotted, and in each slot, a set of links is chosen to transmit. This set of activated links must satisfy an interference constraint. In this work, we use a primary interference model, in which each node is constrained to only activate one neighboring link. In other

This work was supported by NSF Grant CNS-1217048 and ONR Grant N00014-12-1-0064.

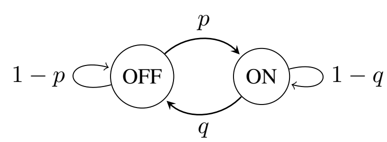


Fig. 1: Markov Chain describing the channel state evolution of each independent channel.

words, the set of activated links forms a matching[1].

Each link $l \in \mathcal{L}$ has a time-varying channel state $S_l(t) \in \{0, 1\}$, and is governed by the Markov Chain in Figure 1. The state of the channel at link l represents the rate at which data can be transmitted over that link. An ON channel can support a unit throughput (single packet transmission), while transmissions over an OFF channel fail.

A. Delayed Channel State Information

We assume that every node has CSI pertaining to each link, delayed by an amount of time proportional to the distance between the node and the link. Specifically, a node n has k-step delayed information of links in $\mathcal{N}_{k+1}(n)$, where $\mathcal{N}_k(n)$ is the set of links that are k hops away from n. In other words, each node has current CSI pertaining to its adjacent links, 1-hop delayed CSI of its 2-hop neighboring links, and so on. This models the effect of propagation and transmission delays on the process of collecting CSI.

B. Centralized Scheduling

A *centralized scheduling algorithm* consists of a single entity making a global scheduling decision for the entire network. In this work, one node is appointed to be the centralized decision-maker, referred to as the *controller*. The controller has delayed CSI of each link, where the delay is relative to that link's distance from the controller, and makes a scheduling decision based on the delayed CSI. This decision is then broadcasted across the network. Throughout this paper we assume the controller broadcasts the schedule to the other nodes instantaneously. In practice, the decision takes a similar amount of time to propagate from the controller as the time required to gather CSI, which effectively doubles the impact of delay in the CSI. Therefore, the theoretical performance of the centralized scheduling algorithm derived in this work provides an upper bound on the performance achievable in practice.

Let $d_r(l)$ be the distance (in hops) of link l from the controller r. The controller has an estimate of $S_l(t)$ based on the delayed CSI. Define the *belief* of a channel to be the probability that a channel is ON given the available CSI at the controller. For link l, the belief $x_l(t)$ is given by

$$x_l(t) = \mathbf{P}\big(S_l(t) = 1 \big| S_l(t - d_r(l))\big). \tag{1}$$

The belief is derived from the k-step transition probabilities of the Markov chain in Figure 1. Namely,

$$\mathbf{P}\big(S(t) = j \big| S(t-k) = i\big) = p_{ij}^k. \tag{2}$$

[1]A matching is a set of links such that no two links share an endpoint.

As the CSI of a channel grows stale, the probability that the channel is ON is given by the stationary distribution of the chain in Figure 1, and denoted as π.

$$\lim_{k\to\infty} p_{01}^k = \lim_{k\to\infty} p_{11}^k = \pi = \frac{p}{p+q}. \tag{3}$$

Since the objective is to maximize the expected sum-rate throughput, the optimal scheduling decision at each time slot is given by the maximum likelihood (ML) rule, which is to activate the links that are most likely to be ON, i.e. the links with the highest belief. Under the primary interference constraint, a set of links can only be scheduled simultaneously if that set forms a matching. Let $\mathcal{M}$ be the set of all matchings in the network. The maximum expected sum-rate is formulated as

$$\max_{m\in\mathcal{M}} \mathbb{E}\Big[\sum_{l\in m} S_l(t) \Big| \{S_l(t - d_r(l))\}_{l\in\mathcal{L}}\Big] \tag{4}$$

$$= \max_{m\in\mathcal{M}} \sum_{l\in m} \mathbb{E}\big[S_l(t) \big| S_l(t - d_r(l))\big] = \max_{m\in\mathcal{M}} \sum_{l\in m} x_l(t). \tag{5}$$

Thus, the optimal schedule is a maximum weighted matching, where the weight of each link is equal to the controller's belief of that link.

C. Distributed Scheduling

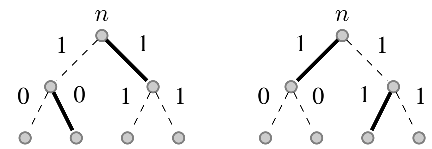
A *distributed scheduling* algorithm consists of multiple entities making independent decisions without coordination. Each node makes a transmission decision for its neighboring links using only the CSI of adjacent links; hence, the performance of distributed scheduling is unaffected by the delay in CSI. The drawback of such policies is that local scheduling decisions may not be globally optimal.

We consider distributed policies in which decisions are made sequentially to avoid collisions. If a node begins transmission, neighboring nodes detect this transmission and can activate a non-conflicting link rather than an interfering link, in a manner similar to collision avoidance in CSMA/CA[2]. This allows us to focus on the sub-optimality resulting from making a local instead of a global decision, rather than the transmission coordination needed to avoid collisions[3].

As mentioned above, the drawback of distributed scheduling is that local decisions can be suboptimal. For example, in Figure 2, node n can choose to schedule either of its neighboring links; if it schedules its right child link, then the total sum rate of the resulting schedule is 1, as in Figure 2a, whereas scheduling the left link results in a sum rate of 2, as in Figure 2b. In a distributed framework, node n is unaware of the state of the rest of the network, so it makes an arbitrary decision resulting in a throughput loss. Moreover, the loss in efficiency due to suboptimal

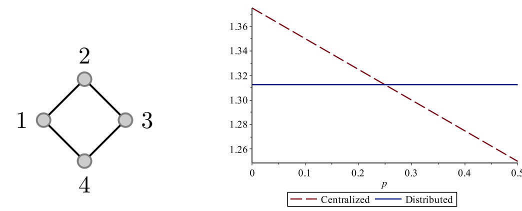
[2]Here we assume a small propagation delay, such that nodes can immediately detect if a neighbor is transmitting.

[3]Alternative transmission coordination schemes are also possible based on RTS/CTS exchanges [13].



(a) Suboptimal Matching (b) Optimal Matching

Fig. 2: Example network: All links are labelled by their channel state at the current time. Bold links represent activated links.



(a) Topology. (b) Expected throughput as a function of channel transition probability p.

Fig. 3: Four-node ring network example.

decisions becomes more pronounced when moving beyond the simple two-state channel model.

III. Centralized vs. Distributed Scheduling

In the previous section, we introduced two primary classes of scheduling policies: distributed and centralized policies. It is known that a centralized scheme using perfect CSI outperforms distributed schemes, due to the aforementioned loss of efficiency in localized decisions. However, these results ignore the effect of delays in collecting CSI. In this section, we revisit the comparison between centralized and distributed scheduling. We show that for sufficiently large CSI delays, there exist distributed policies that perform at least as well as the optimal centralized policy.

As an example, consider the four node network in Figure 3a, and a symmetric channel state model satisfying $p = q$. Without loss of generality, assume node 1 is the controller. In a centralized scheduling scheme, node 1 chooses a schedule based on current CSI for links $(1,2)$ and $(1,4)$, and 1-hop delayed CSI for links $(2,3)$ and $(3,4)$. The resulting expected throughput is computed by first conditioning on the state of the links adjacent to the controller, then on the delayed state of the remaining links, and computing the optimal expected throughput conditioned on this CSI.

$$\begin{aligned} C(p) &= \tfrac{1}{4}(\tfrac{3}{4}(1-p) + \tfrac{1}{4}p) + \tfrac{1}{2}\cdot\tfrac{3}{2} + \tfrac{1}{4}(1 + \tfrac{3}{4}(1-p) + \tfrac{1}{4}p) \\ &= \tfrac{11}{8} - \tfrac{1}{4}p. \end{aligned} \tag{6}$$

Now consider a distributed schedule, in which node 1 makes a scheduling decision based on the state of adjacent links $(1,2)$ and $(1,4)$. After this decision is made, node 3 makes a non-conflicting decision based on the state of links $(3,1)$ and $(3,4)$. The resulting expected throughput is given by conditioning on the event that node 1 has an ON adjacent link to activate.

$$D = \frac{1}{4}\cdot\frac{3}{4} + \frac{3}{4}\cdot\frac{3}{2} = \frac{21}{16} \tag{7}$$

The expected throughput for centralized and distributed scheduling in (6) and (7) is plotted in Figure 3b. As the channel transition probability p increases, the memory in the channel decreases, and the expected throughput of a centralized scheduler decreases. The distributed scheduler, on the other hand, is unaffected by the channel transition probability, as it only uses non-delayed local CSI. For channel transition probabilities $p \geq \frac{1}{4}$, distributed scheduling outperforms centralized scheduling over this network.

Next, we extend this result to general topologies. The throughput degradation of the centralized scheme is a function of the memory in the channel state process. Let μ be a metric reflecting this memory. In the case of a two-state Markov chain, we define

$$\mu \triangleq 1 - p - q. \tag{8}$$

Note that μ is the second eigenvalue of the state transition matrix for the two-state Markov chain, and thus represents the rate at which the chain converges to its steady state distribution [14].

Theorem 1. *For a fixed steady-state probability π, there exists a threshold μ^* such that if $\mu \leq \mu^*$, there exists a distributed scheduling policy that outperforms the optimal centralized scheduling policy.*

Theorem 1 is proven by combining the following four Lemmas.

Lemma 1. *For a fixed steady-state probability π, and state transition probabilities p and $q = \frac{\pi}{1-\pi}p$, the expected sum-rate of any distributed policy is independent of the channel memory μ.*

Lemma 1 follows because the distributed policy does not use delayed CSI.

Lemma 2. *The expected sum-rate of the optimal centralized policy is greater than or equal to that of any distributed policy when $\mu = 1$.*

Lemma 2 follows because when $\mu = 1$, the controller has perfect (non-delayed) CSI.

Lemma 3. *There exists a distributed policy with sum rate greater than or equal to the sum rate of the optimal centralized policy when $\mu = 0$.*

The proof of Lemma 3 follows by showing that when $\mu = 0$, a centralized policy only has CSI pertaining to the links adjacent to the controller. Thus, one can construct a distributed policy that returns the same schedule as the centralized policy.

Lemma 4. *Let $C(p,q)$ be the sum-rate of the optimal centralized algorithm as a function of the channel transition probabilities p and q. For a fixed value of π, $C(p,q)$ is monotonically increasing in $\mu = 1 - p - q$.*

Theorem 1 follows by combining Lemmas 1 - 4 to prove that there exists a value of μ where distributed and centralized achieve the same expected throughput. Thus, there exists a threshold μ^*, such that for $\mu \leq \mu^*$, distributed scheduling outperforms the optimal centralized scheduler. The value of μ^* depends on the topology, and in general, this threshold is difficult to compute. In the following, we characterize the value of the threshold for tree networks.

IV. Tree Topologies

In this section, we characterize the expected throughput over networks with tree topologies. The acyclic nature of these graphs makes them amenable to analysis. We focus on rooted trees, such that one node is the root and every other node has a *depth* equal to the distance from the root. Furthermore, for any node v, the nodes that are connected to v but have depth greater than v are referred to as *children* of v. If u is a child of v, then v is the *parent* of u. This familial nomenclature is standard in the graph-theoretic literature, and simplifies description of the algorithms over tree networks. A *complete* k-ary tree of depth n is a tree such that each node of depth less than n has k children, and the nodes at depth n are *leaf* nodes, i.e. they have no children. This section focuses on symmetric channel models such that $p = q$ to simplify the analysis, but the results are easily extended to asymmetric channels as well. Moreover, this paper provides results for complete binary trees $k = 2$, but the results extend to the case of $k > 2$ as well [15].

A. Distributed Scheduling on Tree Networks

Consider applying the distributed scheduling algorithm over a complete binary tree of depth n, where transmission priorities are assigned in order of depth (lower depth has higher priority). The root node first makes a decision for its neighboring links. Then, the children of the root attempt to activate one of their child links, if this activation does not conflict with their parent's decision. Consequently, the average sum rate can be written recursively by conditioning on the event that the root has an ON child link.

Proposition 1. *Let D_n be the average sum rate of the distributed algorithm over a complete binary tree of depth n. The average sum-rate is computed recursively as*

$$D_n = \frac{3}{4} + \frac{5}{4}D_{n-1} + \frac{3}{2}D_{n-2}. \tag{9}$$

A closed-form expression is obtained by solving the recursion in (9).

$$D_n = -\frac{9}{77}\left(-\frac{3}{4}\right)^n + \frac{6}{11}\cdot 2^n - \frac{3}{7}. \tag{10}$$

The average sum-rate in (10) of the distributed scheduling algorithm is independent of the link transition probability p, as each node only uses the CSI of the neighboring links, which is available without delay. This follows from Lemma 1.

Consider the asymptotic per-link throughput as the number of links grows large. An n-level binary tree has $2^{n+1} - 2$ links. Using the expression in (10), and taking the limit as n grows large while dividing by the number of links, yields

$$\lim_{n\to\infty} \frac{D_n}{2^{n+1} - 2} = \frac{3}{11}. \tag{11}$$

Thus, the distributed priority algorithm achieves a throughput of at least $\frac{3}{11}$ per link.

B. Centralized Scheduling on Tree Topologies

The optimal centralized policy schedules a maximum weight matching over the network, where the weight of each link is the belief given the delayed CSI. For tree networks, the maximum-weight matching is the solution to a dynamic programming (DP) problem. While dynamic programming yields the optimal centralized schedule for a specific observation of delayed CSI, computing the average sum rate requires taking an expectation over the delayed CSI. For larger trees, this analysis becomes difficult; however, a recursive strategy can be used to bound the expected solution to the DP.

Proposition 2. *Let C_n be the average sum rate of the centralized algorithm over a complete binary tree of depth n, when the root node is chosen to be the controller. The expected sum-rate throughput of the optimal centralized controller is bounded recursively as*

$$C_n \leq \frac{3}{4} + \frac{3}{2}(1-2p)^1 C_{n-1} + \frac{1}{2}p(2^n + 1) + (1-2p)^2 C_{n-2} + \frac{2}{3}p(1-p)(2^{n-1}+1). \tag{12}$$

Proposition 2 is proven by bounding the effect of delay on centralized scheduling, and writing the expression for expected throughput by conditioning on the possible state of the links adjacent to the root, for which the optimal decision is computed. Solving the recursion in (12) yields a closed-form upper bound on the expected sum-rate throughput achievable by a centralized scheduler.

The limiting ratio of the centralized throughput to the number of links in the tree (for $p > 0$) is given by

$$\lim_{n\to\infty} \frac{C_n}{2^{n+1} - 2} = \frac{1}{6} \tag{13}$$

Note that this value is independent of p, since in the limit as n grows large, infinitely many nodes are sufficiently far from the root such that the controller has no knowledge of their current state. One third of these links are scheduled (size of a maximum cardinality matching) and they will be in the ON state with probability $\frac{1}{2}$. Hence, the limiting per-link throughput is $\frac{1}{6}$. Under distributed scheduling (11), the achievable throughput is $\frac{3}{11}$. Therefore, as the network grows large, distributed eventually outperforms centralized scheduling.

The threshold p^* beyond which distributed scheduling is optimal is computed by combining (12) and (9). Figure 4 plots p^* as a function of n. Note that as n gets large, this threshold approaches zero, implying that distributed is always better than centralized in large networks.

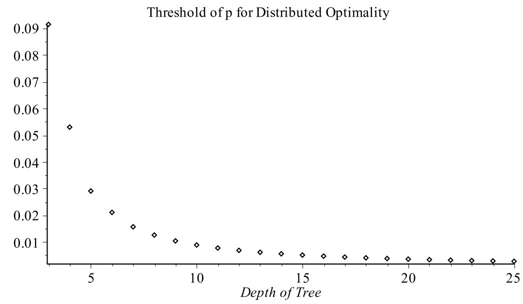


Fig. 4: Threshold value of $p^*(n)$ such that for $p > p^*(n)$, distributed scheduling outperforms centralized scheduling on n-level, binary tree.

V. Simulation Results and Conclusions

In this section, the performance of the distributed policy is compared to the performance of a centralized controller through simulation. For distributed scheduling, transmission priorities are assigned in reverse order of node degree. For each network, we simulate decisions over 100,000 time slots. To begin, consider a 10-node, fully-connected network. The average sum-rate throughput as a fraction of the perfect-CSI throughput[4] is plotted as a function of the channel state transition probability p in Figure 5. In Figure 6, the simulation is applied to a five-by-five grid network, where the centralized controller is located at the central-most node. In both topologies, the distributed expected throughput is constant, as in Lemma 1. These results show that for p small, i.e. channels with high degrees of memory, a purely centralized controller is optimal, as in Lemma 2. As p increases, the expected throughput of centralized scheduling decreases, as proven in Lemma 4, and eventually distributed scheduling outperforms the optimal centralized scheme.

In summary, this work studied wireless scheduling in networks where CSI updates are delayed proportional to distance. In particular, we showed that a centralized scheduling approach suffers from having delayed CSI, and the resulting performance is a function of the memory in the channel state process, as well as the network topology. If the degradation to throughput is significant, we show that distributed scheduling can outperform centralized scheduling, despite making suboptimal transmisison decisions.

[4]This is the throughput attainable by a centralized scheduler with perfect CSI.

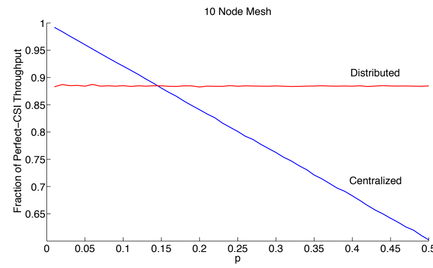


Fig. 5: The fraction of the perfect-CSI throughput obtained as a function of p for 10-node clique topology, over a horizon of 100,000 time slots.

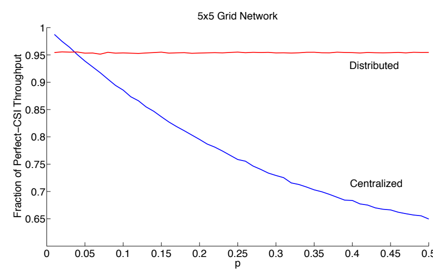


Fig. 6: The fraction of the perfect-CSI throughput obtained as a function of p for a 5 x 5 grid network, over a horizon of 100,000 time slots.

References

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

[2] M. Neely, E. Modiano, and C. Rohrs, "Dynamic power allocation and routing for time-varying wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 1, pp. 89–103, 2005.

[3] 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.

[4] X. Lin and N. B. Shroff, "The impact of imperfect scheduling on cross-layer rate control in wireless networks," in *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, vol. 3. IEEE, 2005, pp. 1804–1814.

[5] A. Dimakis and J. Walrand, "Sufficient conditions for stability of longest-queue-first scheduling: Second-order properties using fluid limits," *Advances in Applied probability*, pp. 505–521, 2006.

[6] C. Joo, X. Lin, and N. B. Shroff, "Understanding the capacity region of the greedy maximal scheduling algorithm in multihop wireless networks," *IEEE/ACM Transactions on Networking (TON)*, vol. 17, no. 4, pp. 1132–1145, 2009.

[7] E. Modiano, D. Shah, and G. Zussman, "Maximizing throughput in wireless networks via gossiping," in *ACM SIGMETRICS Performance Evaluation Review*, vol. 34, no. 1. ACM, 2006, pp. 27–38.

[8] S. Sanghavi, L. Bui, and R. Srikant, "Distributed link scheduling with constant overhead," in *ACM SIGMETRICS Performance Evaluation Review*, vol. 35, no. 1. ACM, 2007, pp. 313–324.

[9] A. Proutiere, Y. Yi, and M. Chiang, "Throughput of random access without message passing." in *CISS*, 2008, pp. 509–514.

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

[11] K. Kar, X. Luo, and S. Sarkar, "Throughput-optimal scheduling in multichannel access point networks under infrequent channel measurements," *Wireless Communications, IEEE Transactions on*, 2008.

[12] L. Ying and S. Shakkottai, "On throughput optimality with delayed network-state information," *Information Theory, IEEE Transactions on*, 2011.

[13] P. Karn, "Maca-a new channel access method for packet radio," in *ARRL/CRRL Amateur radio 9th computer networking conference*, vol. 140, 1990, pp. 134–140.

[14] R. G. Gallager, *Stochastic processes: theory for applications*. Cambridge University Press, 2013.

[15] M. Johnston, "The role of control information in wireless link scheduling," Ph.D. dissertation, MIT, 2015.