Controller Placement in Wireless Networks With Delayed CSI

Matthew Johnston and Eytan Modiano, *Fellow, IEEE*

***Abstract*— We consider the impact of delayed state information on the performance of centralized wireless scheduling algorithms. Since state updates must be collected from throughout the network, they are inevitably delayed, and this delay is proportional to the distance of each respective node to the controller. In this paper, we analyze the optimal controller placement resulting from this delayed state information. We propose a dynamic controller placement framework, in which the controller is relocated using delayed queue length information at each node, and transmissions are scheduled based on channel and queue length information. We characterize the throughput region under such policies, and find a policy that stabilizes the system for all arrival rates within the throughput region.**

***Index Terms*— Communication networks, optimal scheduling, wireless networks.**

I. Introduction

In ORDER to schedule transmissions to achieve maximum throughput, a centralized scheduler must opportunistically make decisions based on the current state of the time-varying channels [1]. The channel state of a link can be measured by its adjacent nodes, who convey this channel state information (CSI) across the network to the scheduler. Due to the transmission and propagation delays over wireless links, the time required for the scheduler to collect CSI throughout the network is significant, and in that time the network state may change relative to the CSI.

There has been extensive work on wireless scheduling, in which centralized approaches are used to control the network [1]–[3]. In theory, centralized scheduling, where a single entity makes a scheduling decision for the entire network, yields high throughput because it is assumed that current CSI is used to compute a globally optimal schedule. However, in practice, the available CSI for centralized scheduling is a delayed view of the network state. Furthermore, the delay in CSI is often proportional to the distance of each link to the controller, since CSI updates must traverse the network.

Several works have considered scheduling with delayed state information. In [4], the authors consider a system in which CSI and QLI (Queue Length Information) updates are only reported once every T time-slots, but the transmitter makes a scheduling decision every slot, using delayed information. They show that delays in the CSI reduce the achievable throughput region, while delays in QLI do not adversely affect throughput. In [5], Ying and Shakkottai study throughput optimal scheduling and routing with delayed CSI and QLI. They show that the throughput optimal policy activates a max-weight schedule, where the weight on each link is given by the product of the delayed queue length and the conditional expected channel state given the delayed CSI. This work is extended in [6], where the authors account for the uncertainty in the state of the network topology as well. Lastly, the work in [7] characterizes the impact of delayed CSI as a function of the network topology, when delays are proportional to distance.

In order to implement a centralized scheduling scheme, one node is assigned the role of a controller, and collects CSI from the rest of the network. Then, the controller uses this CSI to select a set of nodes to transmit at each time slot, in order to maximize throughput while avoiding interference between neighboring links [7]. However, the information available at the controller is delayed an amount of time proportional to the distance of the links from the controller. Since this delay directly impacts the throughput of the scheduling algorithm, the placement of the controller affects network performance.

This paper studies the impact of the controller placement on network performance. To begin, we analyze the static controller placement problem, in which the controller placement is computed a priori, and remains fixed over time. We formulate the optimal controller placement problem, and develop a computationally tractable heuristic to locate the controller in large networks. Next, we propose a dynamic controller placement framework, in which the location of the controller is changed over time. This allows for the controller to be moved to a congested region of the network to increase throughput to this region and provide stability.

The main contribution of this work is the design and analysis of a queue-length based dynamic controller placement algorithm. We prove this algorithm is throughput optimal over all controller placement policies which do not depend on CSI. Additionally, we extend this framework to include controller placement policies which use delayed CSI that is globally available in the network. We provide extensive simulation results to quantify the improvement in dynamic controller placement, and verify the throughput optimality of the proposed policies.

The remainder of this paper is organized as follows. In Section II, we provide the network and information models used in this work. In Section III, we analyze the static controller placement problem. This is extended to a dynamic controller placement in Section IV, where we present an

Manuscript received January 4, 2016; revised September 30, 2016 and December 21, 2016; accepted January 5, 2017; approved by IEEE/ACM Transactions on Networking Editor Y. Yi. Date of publication January 31, 2017; date of current version June 14, 2017. This work was supported by the NSF uder Grant CNS-1217048 and CNS-1524317, and by the ONR under Grant N00014-12-1-0064. This paper was presented at the WiOpt '15.

The authors are with the Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: modiano@mit.edu).

Digital Object Identifier 10.1109/TNET.2017.2651808

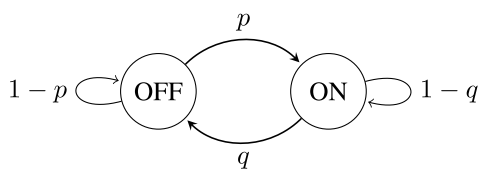


Fig. 1. Markov Chain describing the channel state evolution of each independent channel.

optimal controller placement algorithm. Simulation results are shown in Section V, and conclusions are presented in Section VII.

II. System Model

Consider a network $G(\mathcal{N}, \mathcal{L})$ consisting of a set of nodes $\mathcal{N}$ and links $\mathcal{L}$. Each link is associated with an independent, time-varying channel. Let $S_l(t) \in \{\text{OFF}, \text{ON}\}$ be the state of the channel at link l at time t. Throughout this paper, we use $S_l(t) \in \{0, 1\}$ interchangeably. Assume the channel state evolves over time according to the Markov chain in Figure 1, with transition probabilities p and q. Throughout this work, we assume that $1 - p - q \geq 0$, corresponding to channels with "positive memory." The positive memory property ensures that a channel that was ON k slots ago is more likely to be ON at the current time, than a channel that was OFF k slots ago. This allows the transmitter to make efficient scheduling decisions using delayed CSI. Let p_{ij}^k be the k-step transition probability of the Markov chain, and let π be the steady state probability that the channel is in the ON state.

$$\lim_{k\to\infty} p_{01}^k = \lim_{k\to\infty} p_{11}^k = \pi = \frac{p}{p+q}. \tag{1}$$

One of the nodes is assigned to be the controller, and in each time slot, activates a subset of links for transmission. Assume a primary interference constraint in which a link activation is feasible if the activation is a matching, i.e. no two neighboring links are activated simultaneously. If link l is activated, and $S_l(t) = \text{ON}$, then a packet is successfully transmitted at that time slot. On the other hand, if the channel at link l is OFF, then the transmission fails.

We assume that packets arrive externally to each node i, destined for neighboring node j, according to an i.i.d. Bernoulli arrival process $A_{ij}(t)$ of rate λ_{ij}, and are stored in a queue at that node to await transmission. We assume that packets are destined for a neighboring node, but our results easily extend to the case of multi-hop communication. Let $Q_l(t)$ be the total packet backlog of link l at time t. If a link is scheduled for transmission, there is a packet to transmit, and the corresponding channel is ON, then a packet departs the system from that link.

Let Π be the set of joint controller placement and scheduling policies. The primary objective of this work is to determine a policy $\mathcal{P} \in \Pi$ to stabilize the system of queues. We proceed with a number of important definitions that are standard of the scheduling literature [2].

Definition 1: A link with backlog $Q_l(t)$ is *stable* under policy $\mathcal{P}$ if

$$\limsup_{n\to\infty} \frac{1}{n} \sum_{t=0}^{n-1} \mathbb{E}[Q_l(t)] < \infty \tag{2}$$

The complete network is stable if all queues are stable.

Definition 2: The *throughput region or stability region* Λ is the closure of the set of all rate vectors λ that can be stably supported over the network by a policy $\mathcal{P} \in \Pi$.

Definition 3: A policy is said to be *throughput optimal* if it stabilizes the system for any arrival rate $\lambda \in \Lambda$.

A. Delayed Information Model

In order to determine the subset of links to activate, the controller obtains CSI from each link in the network, and uses the CSI to compute a feasible link activation with maximum expected throughput. Due to the physical distance between nodes, the propagation delay across each link and the need to relay transmissions over multiple hops, the CSI updates received at the controller are delayed an amount of time that is proportional to the distance between each link and the controller [7]. In particular, let $d_i(l)$ be the distance in hops between node i and link l.[1] At time t, each node i has delayed CSI pertaining to link l from time-slot $t - d_i(l)$. In other words, node i has state information $S_l(t - d_i(l))$ for link l. In addition to each node i having delayed CSI, it has delayed queue length information (QLI) $Q_l(t - d_i(l))$ for every link. While in an actual system there is also a delay in distributing the schedule to the nodes, in our formulation we assume that the extra delay on the return path can be captured in the channel state process. (i.e., if the one-way delay is T slots, then the round-trip delay is 2T slots, and this can be effectively captured by channel state transition process).

III. Static Controller Placement

In this section, we consider an off-line controller placement, such that the controller remains fixed over time. We show that the optimal controller placement depends on the network topology as well as the channel transition probabilities. Throughout this section, we consider a saturated system, in which each node as an infinite backlog of packets to be transmitted to each neighbor.

Intuitively, scheduling an ON link at the current time-slot should take priority over scheduling a link that was ON at a previous time slot, but may not be ON currently. Mathematically, this follows from the monotonicity of the k-step transition probability for the Markov chain in Figure 1. Thus, the expected throughput depends on the topology of the network with respect to the controller location, in terms of how many links are 1 hop away, 2 hops away, etc. and which links can be activated simultaneously.

A. Static Placement Example

As a motivational example, consider the topology in Figure 2, and compare the expected sum-throughput attainable by placing the controller at node A, node B, or node C. When the controller is placed at node B, the CSI of the two links adjacent to node B is available without delay, while the CSI corresponding to the other $2k$ links are available at a 1 time-slot delay. Intuitively, if one of two interfering links can be scheduled, and both links are ON, a higher expected throughput is earned by scheduling the link with the smaller

[1] By convention, node i is a distance of 0 hops from its adjacent links.

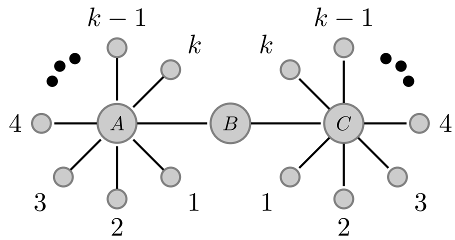


Fig. 2. Barbell Network. Node A and Node C each have a degree of $k+1$.

delay. This intuition leads to a mathematical characterization of the expected throughput, when the controller is located at node B.

For simplicity, assume a symmetric Markov channel in Figure 1, i.e. $p = q$. The probability that k links are OFF is given by $\gamma = (\frac{1}{2})^k$. To compute the expected throughput as a function of controller placement, we first condition on the state of the two links adjacent to node B. Thus, placing the controller at node B results in an expected throughput of

$$\begin{aligned}\text{thpt}_B &= \frac{1}{4}\cdot 2\big((1-\gamma)p_{11}^1 + \gamma p_{01}^1\big)\\ &+ \frac{1}{2}\big(1 + (1-\gamma)p_{11}^1 + \gamma p_{01}^1\big)\\ &+ \frac{1}{4}\big(1 + (1-\gamma^2)p_{11}^1 + \gamma^2 p_{01}^1\big). \qquad (3)\end{aligned}$$

The above expression follows from conditioning on the state of the two adjacent links to B. The first term corresponds to the expected throughput when both links are OFF, in which ON links adjacent to A or C should be scheduled, the second corresponds to the case when one is ON and the other is OFF, in which the ON link should be scheduled, and the last term corresponds to both links being ON, in which one of the ON links are scheduled, based on the states of the other links.

On the other hand, when the controller is placed at node A, the state of the $k+1$ neighboring links are available without delay, the link between B and C is available at a 1 time-slot delay, and the remaining k links are available at a 2 time-slot delay. Placing the controller at node A yields the same expected throughput as placing the controller at node C, due to the symmetry of the network. The expected throughput from placing the controller at node A is derived by first conditioning on the state of the $k+1$ links adjacent to node A, and then conditioning on the state of the link from B to C.

$$\begin{aligned}\text{thpt}_A &= (1-\gamma)\left[1 + \frac{1}{2}p_{11}^1 + \frac{1}{2}\big((1-\gamma)p_{11}^2 + \gamma p_{01}^2\big)\right]\\ &+ \gamma\left[\frac{1}{2}\big(1 + (1-\gamma)p_{11}^2 + \gamma p_{01}^2\big) + \frac{1}{2}\big(\frac{1}{2}p_{11}^1\right.\\ &\left. + \frac{1}{2}\Big((1-\gamma)p_{11}^2 + \gamma p_{01}^2\big)\Big)\right] \qquad (4)\end{aligned}$$

As k grows to infinity, it can be seen that for $p \le \frac{1}{4}$, it is optimal to place the controller at node B, and for $p \ge \frac{1}{4}$ it is optimal to place the controller at either node A or C. This example highlights some important properties of the controller placement problem. In particular, it is clear that the optimal placement depends on the channel transition probabilities. When p is small, it is advantageous to place the controller to minimize the CSI delay throughout the network (e.g. node B in the above example). On the other hand, when p is close to $\frac{1}{2}$, delayed CSI is no longer useful, hence it is better to maximize the amount of local CSI at the controller (e.g. nodes A or C).

B. Optimal Controller Placement

From the previous example, it is clear that the throughput-maximizing controller placement is a function of the channel state transition probabilities p and q, as well as the network topology. Let $\mathcal{M}$ be the set of matchings in the network, i.e., $\forall M \in \mathcal{M}$, M is a set of links which can be scheduled simultaneously without interfering with one another. Under a throughput maximization objective, the controller schedules the set of links maximizing the expected sum-rate throughput with respect to the CSI delays at that node. Consequently, the controller placement optimization is a max-weight matching, where the weight on each link is the *belief* of that link, or the probability the channel is in the ON state.

$$c = \arg\max_{r\in\mathcal{N}} \mathbb{E}_S\left[\max_{M\in\mathcal{M}} \sum_{l\in M} \mathbb{E}\big[S_l(t)\big|S_l(t-d_r(l)) = S_l\big]\right] \qquad (5)$$

$$= \arg\max_{r\in\mathcal{N}} \mathbb{E}_S\left[\max_{M\in\mathcal{M}} \sum_{l\in M} p_{S_l,1}^{d_r(l)}\right] \qquad (6)$$

In equation (6), $p_{i,j}^k$ is the k-step transition probability of the Markov chain in Figure 1. Equation (6) follows since the channel state satisfies $S_l(t) \in \{0,1\}$. Computing a maximum matching requires solving an integer linear program (ILP) and it is known to be solvable in $\mathcal{O}(|\mathcal{L}|^3)$-time [9]. However, computing the optimal controller position in (6) requires computing the expectation of the maximum matching, which involves solving the ILP for every state sequence $\mathbf{S}(t) \in \{0,1\}^{|\mathcal{L}|}$.

C. Controller Placement Heuristic

The mathematical formulation for computing the optimal controller location given in (6) depends on the distance between nodes, as well as the channel state statistics. However, this computation has a complexity that grows exponentially with the size of the network. Hence, we propose a computationally tractable heuristic for computing the optimal controller location, which is shown to be near-optimal in terms of the resulting expected throughput.

In this heuristic each node is assigned a weight based on its degree. As the memory in the channel process decreases, the best controller location is the node most likely to have an ON neighboring link, i.e. the node with the highest degree. To model this, node n is assigned a weight of $(1-(1-\pi)^{\Delta_n})$, where Δ_n is the degree of node n, which is equal to the probability of having an adjacent ON link.

The controller is placed at the location maximizing the information about the network. Intuitively, the controller should be "close" to as many highly weighted nodes as possible. However, "closeness" must reflect the memory in the system. Thus, we apply a scaling factor, which is a function of the distance to the controller, given by $(1-p-q)^{d_i(n)}$, where $d_i(n)$ is used in this context to refer to the distance between

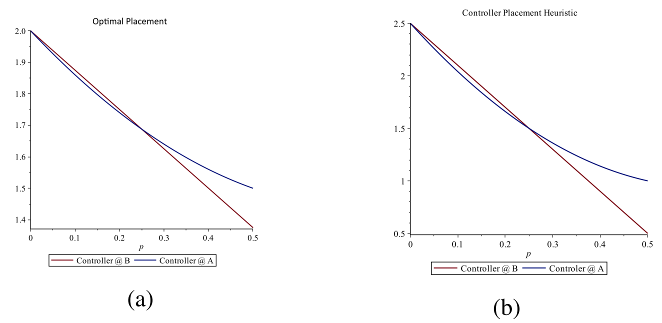


Fig. 3. Evaluation of the controller placement heuristic for the barbell network and various channel transition probabilities $p = q$. (a) Expected Throughput. (b) Heuristic Weight.

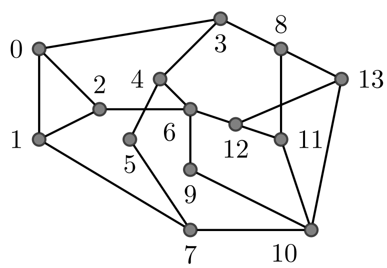


Fig. 4. Sample 14 Node network topology.

two nodes i and n. This scaling represents the transition rate of the Markov Chain in Figure 1 as a function of the distance to the controller. Thus, this scaling factor represents the effect on expected throughput in having delayed information. Our heuristic maximizes the weighted sum-distance to each node as shown in (7). Therefore, the controller is placed according to:

$$c = \arg\max_{r} \sum_{n \in \mathcal{N}} (1-p-q)^{d_r(n)}(1-(1-\pi)^{\Delta_n}). \quad (7)$$

Placing the controller according to (7) preserves the important properties of the optimal controller placement in (6).

The heuristic in (7) is very similar to the well-known p-median problem [10], for $p = 1$. The 1-median problem seeks to find the node that minimizes the sum distance to all other nodes. In contrast, the controller placement assigns weights to nodes and uses a convex function of distance in this computation. These differences ensure that the controller is placed at the location that yields high throughput, which may not be the same as the solution to the 1-median problem.

Figure 3 shows the expected throughput for the network of Figure 2, when the controller is located at node A and node B, as well as the value of the heuristic objective in (7). These results show the controller placement in (7) is similar in terms of throughput to the optimal placement.

In general, the heuristic returns a controller location that yields throughput close to that of the optimal placement. Consider the topology in Figure 4, for which the heuristic of (7) is applied and compared to the optimal controller placement, shown in Table I. We find that the heuristic controller placement is often the same as the throughput-optimal controller placement. Furthermore, in instances where the throughput-optimal location differs from the heuristic location, the controller is placed at a location yielding an average throughput within 1% of optimal.

TABLE I

RESULTS OF CONTROLLER PLACEMENT PROBLEM OVER THE TOPOLOGY OF FIGURE 4. OPTIMAL PLACEMENT IS COMPUTED BY SOLVING (6) VIA BRUTE FORCE, WHILE HEURISTIC REFERS TO (7). PERCENT ERROR REFERS TO THE DIFFERENCE BETWEEN THE OPTIMAL AND HEURISTIC PLACEMENTS

Strategy	Optimal Placement	Heuristic Placement	% Error
$p = 0.05$	6	6	0
$p = 0.1$	6	6	0
$p = 0.15$	6	6	0
$p = 0.2$	6	6	0
$p = 0.25$	6	6	0
$p = 0.3$	6	6	0
$p = 0.35$	10	6	.0289
$p = 0.4$	10	6	.2974
$p = 0.45$	10	6	.5704
$p = 0.5$	10	6	.7937

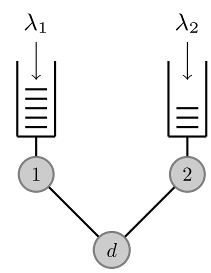


Fig. 5. Example 2-node system model. Packets arrive at nodes 1 and 2 for transmission to node d.

IV. DYNAMIC CONTROLLER PLACEMENT

For a fixed controller location, the links physically close to the controller obtain a higher throughput than those far from the controller due to the delay in CSI. By relocating the controller, the throughput in different regions of the network can be improved. In this section, we consider policies which recompute the controller location dynamically in order to balance the throughput throughout the network. We characterize the throughput region of the controller placement and scheduling problem above, and propose a throughput optimal joint controller placement and scheduling policy based on the information available at each node.

A. Two-Node Example

In order to illustrate the effect of dynamic controller relocation, consider the three-node system in Figure 5, where nodes 1 and 2 compete over a shared medium to send packets to destination d. Let λ_i is the arrival rate of packets at node i destined for d. Each node has instantaneous CSI pertaining to its channel at the current time, and 1-step delayed CSI of the other channel. Let Λ_1 be the throughput region when the controller is fixed at node 1, and let Λ_2 be the throughput region when the controller is fixed at node 2. The throughput regions Λ_r are characterized for each controller location r by

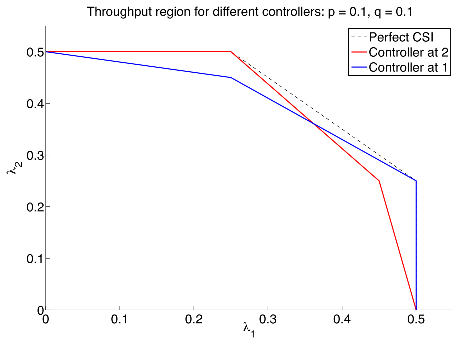


Fig. 6. Throughput regions for different controller scenarios. Assume the channel state model satisfies $p = 0.1, q = 0.1$, and $d_1(2) = d_2(1) = 1$.

the following linear program (LP).

$$\begin{aligned}&\text{Maximize: } \epsilon\\&\text{Subject To: } \lambda_i + \epsilon \le \sum_{(s_1,s_2)\in\mathcal{S}} \times \mathbf{P}((S_1(t-d_r),S_2(t-d_r))=(s_1,s_2))\\&\qquad \cdot\, \alpha_i(s_1,s_2)\mathbb{E}[S_i(t)|S_i(t-d_r(i))=s_i] \quad \forall i\in\{1,2\}\\&\sum_{i=1}^{M}\alpha_i(s_1,s_2)\le 1 \quad \forall \mathbf{s}\in\mathcal{S}\\&\alpha_i(s_1,s_2)\ge 0 \quad \forall \mathbf{s}\in\mathcal{S},\ i\in 1,2 \end{aligned} \tag{8}$$

In the above LP, $\alpha_i(s_1, s_2)$ represents the fraction of time link i is scheduled when delayed CSI at the controller is (s_1, s_2). To maintain stable queue lengths, the arrival rate to each queue must be less than the service rate at that queue, which is determined by the fraction of time the node transmits, and the expected throughput obtained over that link. For the case when the controller is at node r, Λ_r is the set of arrival rate pairs $\lambda = (\lambda_1, \lambda_2)$ such that there exists a solution to (8) satisfying $\epsilon > 0$. The proof that Λ_r is in fact the stability region of the system is found in [5].

The throughput regions Λ_1 and Λ_2 are plotted in Figure 6 for the case when $p = q = 0.1$. The throughput region is larger in the dimension of the controller, as a higher throughput is obtained at the node for which current CSI is available. The other node cannot attain the same throughput due to the CSI delay. Now consider a time-sharing policy, alternating between placing the controller at node 1 and node 2. The resulting throughput region Λ is given by the convex hull of Λ_1 and Λ_2, which is shown as the dotted black line in Figure 6. Time-sharing between controller placements allows for higher throughputs than if the controller is fixed at either node. For example, the point $(\lambda_1, \lambda_2) = (\frac{3}{8} - \epsilon, \frac{3}{8} - \epsilon)$, for ϵ small, is not attainable by any fixed controller placement; however, this throughput point is achieved by an equal time-sharing between controller locations.

The optimal time sharing between controller placements depends on the arrival rate and channel transition probabilities. This information is usually unavailable to the controller, and the control policy must stabilize the system even if the parameters change. Thus, we propose a dynamic controller placement and scheduling policy which achieves the full throughput region Λ using only delayed QLI for controller placement, and delayed CSI and QLI for scheduling, with no information pertaining to the arrival rates.

B. Queue Length-Based Dynamic Controller Placement

We design our dynamic controller placement policy based on the same intuition as the static case. As described previously, one node is assigned the role of the controller. In order to compute the new controller location in a distributed fashion, each node must be able to compute the controller location based only on globally available information. In particular, we consider algorithms that are based on sufficiently delayed QLI, and do not consider CSI in deciding where to place the controller,[2] since it is known that delayed QLI does not affect the throughput performance of the system [4]. After the controller is selected, the controller chooses the transmission schedule based on the delayed CSI and QLI. Assume that in every slot, the location of the controller is recomputed. Practically, it may be desirable to restrict controller relocation to a longer interval, and this extension is addressed in Section VI.

For simplicity, throughout this section we assume that each link mutually interferes with each other link. The results in this section can be easily extended to arbitrary interference models, but doing so complicates the analysis considerably, without providing additional insight to the problem.

Let Π be the set of policies which make a controller-placement decision based on QLI and not CSI, and make a scheduling decision based on the delayed CSI and QLI at the controller. In this section, we characterize the throughput region under such policies, and propose the *dynamic controller placement and scheduling (DCPS)* policy, which is proven to stabilize the system for all arrival rates within the throughput region.

Theorem 1 shows that the throughput region is characterized by the following LP.

$$\begin{aligned}&\text{Max. } \epsilon\\&\text{S.t.: } \lambda_i + \epsilon \le \sum_{\mathbf{s}\in\mathcal{S}} \mathbf{P}_S(\mathbf{s}) \sum_{r=1}^{M} \beta_r \alpha_i^r(\mathbf{s})\mathbb{E}\\&\qquad \times \big[S_i(t)\big|S_i(t-d_r(i))=s_i\big] \quad \forall i\in\{1,\dots,M\}\\&\sum_{i=1}^{M}\alpha_i^r(\mathbf{s})\le 1 \quad \forall \mathbf{s}\in\mathcal{S}\\&\sum_{r=1}^{M}\beta_r\le 1\\&\alpha_i^r(\mathbf{s})\ge 0, \beta_r\ge 0 \quad \forall \mathbf{s}\in\mathcal{S}, i, r\in 1,\dots,M \end{aligned} \tag{9}$$

This LP is an extension of the LP given in (8) to M nodes, with the addition of a time-sharing between controller locations. The optimization variables β_r and $\alpha_i^r(\mathbf{s})$ correspond to controller placement and link scheduling policies respectively. The variables β_r represent the fraction of the time that node r is elected to be a controller, and $\alpha_i^r(\mathbf{s})$ is the fraction of time

[2]For networks with a large diameter, the common CSI may be too stale to be used in controller placement; thus, we restrict our attention to policies which utilize QLI to make controller placement decisions, but not CSI.

Algorithm 1 Dynamic Controller Placement and Scheduling (DCPS) Policy

1: Choose a controller by solving the following optimization as a function of the delayed queue backlogs $Q_i(t-\tau_Q)$.

$$r^* = \arg\max_r \times \left(\sum_{\mathbf{s}\in\mathcal{S}} \mathbf{P}_\mathbf{S}(\mathbf{S}(t-d_r)=\mathbf{s}) \max_i Q_i(t-\tau_Q) p_{s_i,1}^{d_r(i)} \right) \tag{10}$$

where $P_S(\mathbf{s})$ is the steady state probability of the channel-state process.

2: Controller observes delayed CSI: $S(t-d_{r^*}(i)) = \mathbf{s}$ for each i.

3: Controller schedules the following queue to transmit.

$$i^* = \arg\max_i Q_i(t-\tau_Q) p_{\mathbf{s}_i,1}^{d_{r^*}(i)} \tag{11}$$

that controller r schedules node i when the controller observes a delayed CSI of $\mathbf{S}(t-d_r)=\mathbf{s}$. Note that $\mathbf{P}_S(\mathbf{s})$ is the stationary probability of the Markov chain in Figure 1. The throughput region Λ, is the set of all non-negative arrival rate vectors λ such that there exists a feasible solution to (9) for which $\epsilon \geq 0$. This implies that there exists a stationary policy such that the effective service rate at each queue is greater than the arrival rate to that queue.

Theorem 1 (Throughput/Stability Region): For any non-negative arrival rate vector λ, the system can be stabilized by some policy $\mathcal{P} \in \Pi$ if and only if $\lambda \in \Lambda$, as given by (9).

Necessity is shown in Lemma 1, and sufficiency is shown in Theorem 2 by proposing a throughput optimal joint scheduling and controller placement algorithm, and proving that for all $\lambda \in \Lambda$, that policy stabilizes the system.

Lemma 1: Suppose there exists a policy $\mathcal{P} \in \Pi$ that stabilizes the network for all $\lambda \in \Lambda$. Then, there exists a β_r and $\alpha_i^r(\mathbf{s})$ such that (9) has a solution with $\epsilon \geq 0$.

The proof of Lemma 1 is given in the Appendix. Lemma 1 shows that for all $\lambda \in \Lambda$, there exists a stationary policy $\Pi_{\text{STAT}} \in \Pi$ that stabilizes the system, which places the controller at node r with probability β_r, and schedules node i to transmit when the delayed CSI at controller r is S with probability $\alpha_i^r(\mathbf{s})$ in (9).

Next, we propose the dynamic controller placement and scheduling (DCPS) policy, and show that this policy stabilizes the network whenever the arrival rate vector is interior to the capacity region Λ. This proves the sufficient condition of Theorem 1.

Algorithm 1 computes a controller location as a function of τ_Q, the delay to QLI. τ_Q is taken to be large enough that all nodes have access to $Q(t-\tau_Q)$ and can therefore calculate r^* without requiring additional communication. Then, the selected controller computes a max weight schedule based on delayed CSI and QLI.

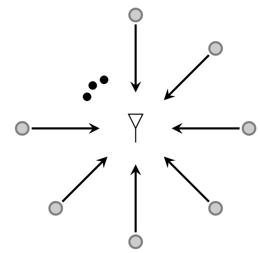


Fig. 7. Example star network topology where each node measures its own channel state instantaneously, and has d-step delayed CSI of each other node.

Theorem 2: Let $T_{SS}(\epsilon)$ be a large constant defined for some $\epsilon > 0$, such that

$$\left| \mathbf{P}\big(\mathbf{S}(t)=\mathbf{s}\big|\mathbf{S}(t-T_{SS}(\epsilon))\big) - \mathbf{P}\big(\mathbf{S}(t)=\mathbf{s}\big) \right| \leq \frac{\epsilon}{2|\mathcal{S}|} \tag{12}$$

For any arrival rate λ, and $\epsilon > 0$ satisfying $\lambda + \epsilon\mathbf{1} \in \Lambda$, the DCPS policy stabilizes the system if $\tau_Q \geq d_{max} + T_{SS}(\epsilon)$.

Under policy DCPS, the controller is placed at the node maximizing the expected max weight schedule, over all possible states. Then, the controller observes the delayed CSI and schedules the max-weight schedule for transmission as in [2] and [5]. Moving the controller to nodes with high backlog increases the throughput at those nodes, keeping the system stable. The proof of Theorem 2 follows by defining the Lyapunov drift, and showing that as the system backlogs grow large, the Lyapunov drift in becomes negative, implying system stability [2]. In particular, we consider the Lyapunov drift over a T-slot window, where T is large enough that the system reaches its steady state distribution. The proof is given in the supplemental material.

The throughput optimal controller placement uses delayed QLI $\mathbf{Q}(t-\tau_Q)$. The delay τ_Q must be sufficiently large such that $\mathbf{Q}(t-\tau_Q)$ is available at every node, i.e. $\tau_Q \geq d_{\max}$. Moreover, we require that $\tau_Q \geq d_{\max} + T_{SS}(\epsilon)$, where $T_{SS}(\epsilon)$ is the same order as the time required for the channel state process to approach its steady state distribution (i.e. the mixing time of the Markov process). Note that in making placement decisions, even though more recent QLI is available, an older version of the QLI is needed for throughput optimality. This counter-intuitive result stems from the observation that longer queues are correlated with bad channel states, thus, placing the controller at nodes with longer queues would bias the placement toward nodes with bad channel state. By using delayed QLI, the channel state is decoupled from the queue-length process.

The throughput optimal controller placement given by Theorem 2 takes on a simpler form for specific topologies. In particular, consider the hub topology in Figure 7.

Corollary 1: Consider a system of M nodes, where only one can transmit at each time. Assume the controller has full knowledge of its own channel state and d-slot delayed CSI for each other channel, as in Figure 7. At time t, the DCPS policy places the controller at the node with the largest backlog at time $t-\tau_Q$.

$$r^* = \arg\max_r Q_r(t-\tau_Q) \tag{13}$$

Corollary 1 follows due to the symmetry of the system. The complete proof is given in the Appendix. For the topology in Figure 7, the throughput optimal controller placement policy is simply to place the controller at the node with the longest backlog. Note the queue lengths in the above corollary must still be delayed according to Theorem 2.

C. Controller Placement With Delayed CSI

In the previous section, the throughput optimal joint controller placement and scheduling policy is presented with the restriction that only delayed QLI is used for controller placement. The motivation behind this restriction is that delayed QLI can be available at each node, allowing the controller location to be computed without further communication between nodes. In this vein, the channel state of each node $d_{\max}$ slots in the past can also be globally available, since $d_{\max}$ is the largest CSI delay in the network. If the network has a small diameter, or a high degree of memory, the additional CSI has significant impact on performance. In this section, we characterize the throughput region under controller placement policies using both delayed QLI and CSI, and propose an extension to the DCPS policy which stabilizes the system for all arrival rates within this stability region.

Let Π_{CS} be the set of policies which use delayed QLI and globally delayed CSI for controller placement. The new throughput region is characterized by the following LP.

Maximize: ϵ

Subject To:

$$\lambda_i + \epsilon \leq \mathbb{E}_{\mathbf{s}}\Big[\sum_{r=1}^{M} \beta_r(\mathbf{s}) \cdot \sum_{\mathbf{s}' \in \mathcal{S}} \mathbf{P}(S(t - d_r(i) = \mathbf{s}') \; |S(t - d_{\max}) = \mathbf{s})\alpha_i^r(\mathbf{s}') p_{s_i',1}^{d_r(i)}\Big] \quad \forall i \in \{1, \dots, M\}$$

$$\sum_{i=1}^{M} \alpha_i^r(\mathbf{s}') \leq 1, \alpha_i^r(\mathbf{s}) \geq 0 \quad \forall \mathbf{s} \in \mathcal{S}$$

$$\sum_{r=1}^{M} \beta_r(\mathbf{s}) \leq 1, \beta_r(\mathbf{s}) \geq 0 \quad \forall \mathbf{s} \in \mathcal{S} \tag{14}$$

The above LP is an extension of (9) allowing β_r to be a function of $\mathbf{S}(t - d_{\max})$. The optimization variables $\beta_r(\mathbf{s})$ and $\alpha_i^r(\mathbf{s}')$ correspond to controller placement and link scheduling policies respectively. Note that $p_{s_i,1}^{d_r(i)}$ is the k-step transition probability (where $k = d_r(i)$) of the Markov channel state (Figure 1). The throughput region, Λ_{CS}, is the set of all non-negative arrival rate vectors λ such that there exists a feasible solution to (9) for which $\epsilon \geq 0$. Note, Λ_{CS} is the set of arrival rate vectors such that some policy $\mathcal{P}' \in \Pi_{CS}$ can stabilize the system (policies using both QLI and CSI). This is in contrast to Λ of Section IV-B, in which the arrival rate vectors must be stabilized by a policy $\mathcal{P} \in \Pi$ (policies in which controller placement only uses QLI). Further, since any policy $\mathcal{P} \in \Pi$ also belongs to the policy space Π_{CS} (a controller placement policy that has access to CSI, but ignores it), we can conclude that the stability regions satisfy $\Lambda \subset \Lambda_{CS}$.

Algorithm 2 Dynamic Controller Placement and Scheduling With CSI (DCPS-CSI) Policy

1: Choose a controller by solving the following optimization as a function of the delayed queue backlogs $Q_i(t - \tau_Q)$ and delayed CSI $\mathbf{S}(t - d_{\max})$.

$$r^* = \arg\max_r \sum_{\mathbf{s} \in \mathcal{S}} \mathbf{P}\big(\mathbf{S}(t - d_r(i)) = \mathbf{s} \big| \mathbf{S}(t - d_{\max})\big) \cdot \max_i Q_i(t - \tau_Q) p_{s_i,1}^{d_r(i)} \tag{15}$$

where $P_S(\mathbf{s})$ is the steady state probability of the channel-state process.

2: Controller observes delayed CSI: $S(t - d_{r^*}(i)) = \mathbf{s}$

3: Controller schedules the following queue to transmit.

$$i^* = \arg\max_i Q_i(t - \tau_Q) p_{\mathbf{s}_i,1}^{d_{r^*}(i)} \tag{16}$$

Theorem 3 (Throughput/Stability Region): For any non-negative arrival rate vector λ, the system is stabilized by some policy $\mathcal{P} \in \Pi_{CS}$ if and only if $\lambda \in \Lambda_{CS}$.

Necessity is shown in Lemma 2 and sufficiency is shown in Theorem 4 by proposing a throughput optimal joint scheduling and controller placement algorithm, and proving that for all $\lambda \in \Lambda_{CS}$, that policy stabilizes the system.

Lemma 2: Suppose there exists a policy $\mathcal{P} \in \Pi_{CS}$ that stabilizes the system. Then, there exists variables $\beta_r(\mathbf{s})$ and $\alpha_i^r(\mathbf{s}')$ such that (9) has a solution with $\epsilon \geq 0$.

The proof of Lemma 2 follows the same procedure as that of Lemma 1. Lemma 2 shows that for all $\lambda \in \Lambda_{CS}$, there exists a stationary policy $\Pi_{\text{STAT}} \in \Pi_{CS}$ that stabilizes the system, by placing the controller at r when the maximally delayed CSI is $\mathbf{s}$ with probability $\beta_r(\mathbf{s})$, and schedules i to transmit when the delayed CSI at controller r is S' with probability $\alpha_i^r(\mathbf{s}')$ as in (14).

The DCPS policy of Theorem 2 is extended in Algorithm 2 to utilize globally delayed CSI as well as delayed QLI for controller placement, such that the extended policy is throughput optimal. This proves the sufficient condition of Theorem 3. Algorithm 2 is similar to algorithm 1, except that all nodes also have access to delayed CSI $\mathbf{S}(t - d_{\max})$, so this is incorporated into the controller location algorithm.

Theorem 4: The DCPS-CSI policy in (15) and (16) is throughput optimal if $\tau_Q > d_{max}$.

The proof of Theorem 4 follows according to the steps of the proof of Theorem 2 with modifications made to the conditioning throughout the proof.

Under policy DCPS-CSI, the controller is placed at the node maximizing the expected max-weight schedule, over all possible states, where this expectation is conditioned on globally available delayed CSI. Then the controller observes the delayed CSI and schedules the max-weight activation for transmission as in [2] and [5]. Note that for controller placement policy which only uses QLI, a very large delay is required for the DCPS policy, as the channel state must be independent from the queue length at that time. On the other hand, when the controller placement policy also depends on the delayed CSI $\mathbf{S}(t-d_{\max})$, the queue length delay only needs to be larger than $d_{\max}$, as is the case with policy DCPS-CSI. This follows because the CSI takes away the dependence of the channel state on the delayed QLI.

While the DCPS-CSI policy provides the optimal controller placement and scheduling decisions, the computation in Algorithm 2 is computationally intensive, similar to other work in the max-weight scheduling literature. The heuristics in Section III, along with typical heuristics for throughput optimal scheduling, can be combined to generate computationally tractable approaches for controller placement and scheduling. On one hand, when the queue lengths are similar, the optimal location is that of the static controller location formulation. This depends on the shape of the topology, and the channel transition probabilities. On the other hand, when the queue lengths are very uneven, or when the topology is such that the information delays are the same across the network, then the queue length will dictate the optimal controller placement. Controller placement heuristics should weigh both queue lengths, as well as the network parameters in order to determine the best controller location.

V. Simulation Results

To begin, we simulate a 6-node network with a topology given in Figure 7, and Bernoulli arrival processes of different rates. Assume the controller has instantaneous CSI for its channel, and homogeneously delayed (2 slots) CSI of each other channel. For each symmetric arrival rate vector λ, we simulate the evolution of the system over 100,000 time slots, and compute the average system backlog over those time slots. The results are plotted in Figure 8. Clearly, for small arrival rates, the average queue length remains very small. As the arrival rates increase towards the boundary of the stability region, the average system backlog starts to slightly increase. When the arrival rate grows beyond the stability region, the average queue length increases greatly, since packets arrive faster than they can be served in the system, implying that the system is unstable in this region.

Figure 8 compares several controller placement policies: a fixed controller, as in Section III, a policy that chooses a controller at each time uniformly at random, which is optimal when the arrival rate is the same to each node as it represents the correct stationary policy to stabilize the system, the DCPS policy using delayed QLI for controller placement, and the DCPS policy using delayed QLI and CSI.

Figure 9 repeats the above experiment for a channel that is more likely to be ON than OFF, specifically, for $p = 0.4$, and $q = 0.1$. In this Simulation, we compare a fixed controller policy, a policy that chooses a controller at each time uniformly at random, the DCPS policy using delayed QLI for controller placement, and the Longest Delayed Queue (LDQ) policy. The LDQ policy is taken from (13) and places the controller at the node with the longest delayed QLI. In this case, we see that the DCPS offers a much higher throughput than the other policies. This also highlights the effect that the channel statistics have on the optimal policy, over the LDQ policy.

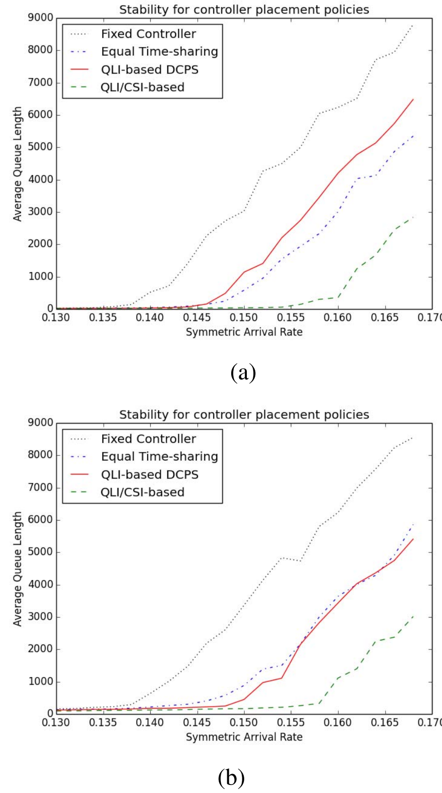


Fig. 8. Simulation results for different controller placement policies, with channel model parameters $p = 0.1, q = 0.1$. (a) 2-Step Delayed QLI. (b) 100-Step Delayed QLI.

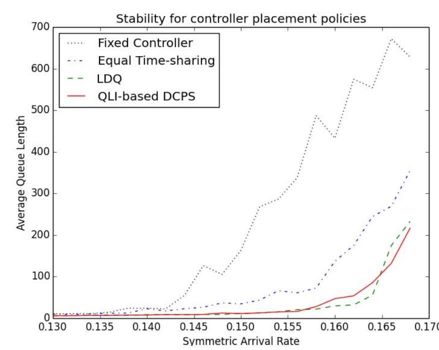


Fig. 9. Simulation results for different controller placement policies, with channel model parameters p = 0.4, q = 0.1.

In Figure 8a, the DCPS policy uses 2-step delayed QLI to place the controller. In this case, the DCPS policy fails to stabilize the system for the same set of arrival rates as the time-sharing policy, implying that the DCPS policy is not throughput optimal. However, in Figure 8b, the delay on the QLI is increased to 100 time-slots. In this scenario, the DCPS policy does stabilize the system for all symmetric arrival rates in the stability region, demonstrating the fact that significantly delayed queue-length information is required for throughput optimality. In contrast, note that when the CSI is also used for controller placement, the throughput does not benefit from additional delay in the QLI. In this example, dynamically changing the controller location using QLI provides a 7% increase in capacity region over the static controller placement, while using both QLI and CSI results in around an additional 6% improvement.

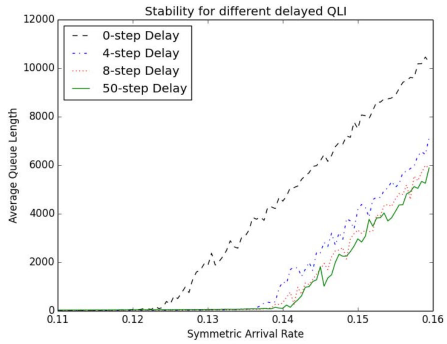


Fig. 10. Effect of QLI-delay on system stability, for $p = q = 0.1$. Each curve corresponds to a different value of τ_Q.

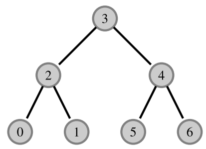


Fig. 11. Two-level binary tree topology.

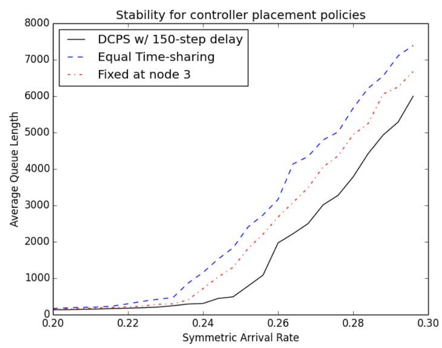
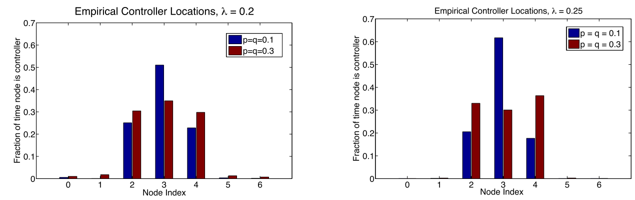


Fig. 12. Average queue length versus symmetric arrival rate for tree network in Figure 11.

Figure 10 illustrates the effect of the delay in QLI on the stability of the system. This figure presents four different values for τ_Q, the delay in the QLI used by the controller placement policy. As can be seen from the figure, the stability region increases with τ_Q. As τ_Q increases, the improvements to the stability region become smaller, as the stability region of the policy approaches the full stability region. In this example, using sufficiently delayed QLI yields a 16% increase in the stability region of the system over the policy which uses current QLI. However, if delay is a concern, less delayed QLI can be used to recover most of the throughput region.

Next, we simulate the controller placement algorithm over the network in Figure 11, to compare the dynamic controller placement with the static controller placement in Section III. Figure 12 analyzes the stability of the system over different controller placement policies. The black solid curve represents the DCPS policy, with QLI delay $\tau_Q = 150$. This policy is compared with the policy that randomly selects the controller and the policy that places the controller at node 3. These results show that relocating the controller according to the DCPS policy gives improvements over both the static placement, and a equal time-sharing between controller locations.



(a) (b)

Fig. 13. Fraction of time each node is selected as the controller under DCPS for the topology in Figure 11. Blue bars correspond to system with $p = q = 0.1$, and red bars correspond to system with $p = q = 0.3$. (a) Symmetric arrival rate $\lambda = 0.2$. (b) Symmetric arrival rate $\lambda = 0.25$.

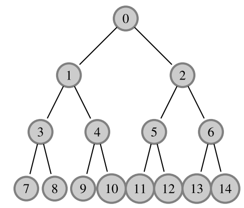
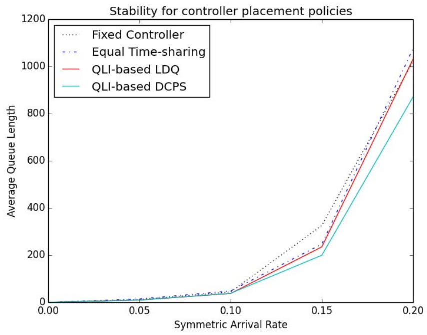


Fig. 14. 4-Level Binary Tree Topology.

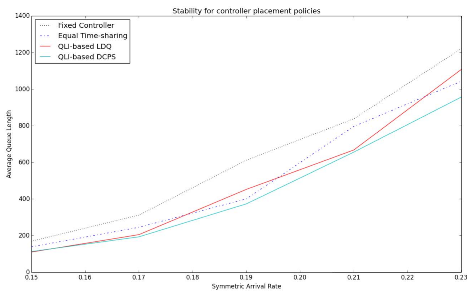
Figure 13 shows the fraction of time each node is selected as the controller under the DCPS policy for the binary-tree topology of Figure 11. For small transition probabilities (e.g. $p = q = 0.1$), the central node 3 is chosen as the controller most frequently. When the transition probabilities increase (e.g. $p = q = 0.3$), then more time is spent with nodes 2 and 4 as controllers. This corresponds to the insight gained from the solution to the static controller placement shown in Section III.

To better model practical scenarios, we simulate the effect of dynamic controller relocation on larger topologies with a variety of interference patterns. First, consider the topology in Figure 11. We simulate the controller placement and scheduling framework over this topology under both a one-hop and two hop interference constraint, where k-hop interference is defined as a constraint that no two nodes within k hops of each other can transmit simultaneously. Figure 15 shows a comparison of average queue length (from which we can derive delay) for a fixed controller policy, equal time sharing policy, LDQ policy, and DSCP policy. These results show that the QLI-based policies outperform the other controller placement policies in terms of throughput region. We note that for interference constraints where many links can be scheduled simultaneously, the location of the controller has a smaller effect on throughput, and relocating the controller dynamically is less important than in the high-interference regime. Similar results can be seen for the larger tree topology in Figure 14, as shown in Figure 16. For the large topology, the DSCP policy requires a significant computation time, so the LDQ policy is used to illustrate the effect of QLI-dependent controller placement. We expect the LDQ and DSCP policies to achieve similar performance based on the experiments on the topology in Figure 11.

Lastly, we study the distribution of the controller location through the simulation for these topologies. Consider the topology in Figure 14. When running the LDQ controller

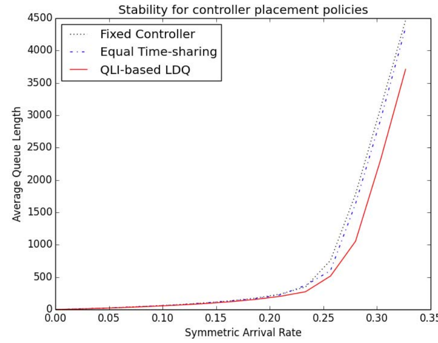


(a)

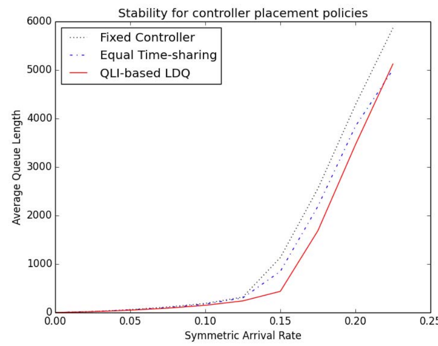


(b)

Fig. 15. Average queue length versus symmetric arrival rate for tree network in Figure 11 under various interference patters. (a) One-hop Interference. $p = 0.1, q = 0.4$. (b) Two-hop Interference. $p = q = 0.3$.

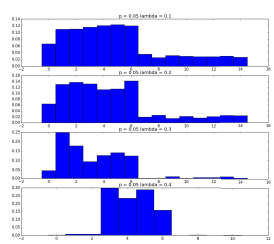


(a)

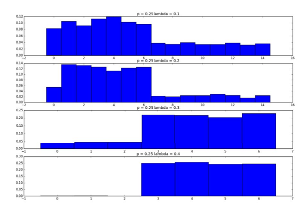


(b)

Fig. 16. Average queue length versus symmetric arrival rate for tree network in Figure 14 under various interference patters. (a) One-hop interference. (b) Two-hop interference.



(a)



(b)

Fig. 17. Controller Placement Distribution under the LDQ policy for the tree topology in 14, under one-hop interference. (a) $p = q = 0.05$. (b) $p = q = 0.3$.

placement policy, we measure the frequency that each node is selected as the controller, shown in Figure 17. Each figure consists of a separate histogram for arrival rates $\lambda = 0.1, 0.2, 0, 3$, and 0.4 to show the effect of increased arrival rate on the controller placement. We assume one-hop interference, and run the experiment for a slowly varying channel, $p = q = 0.05$, and a quickly varying channel, $p = q = 0.3$. Figure 17 shows that the controller is placed primarily at the most central nodes in the topology. When the arrival rate is small, the controller moves to the leaf nodes as well, where as this is suboptimal for larger arrival rate vectors. Moreover, when the channel has less memory, the controller is frequently placed further away from the topological center of the graph, and when there is a high degree of memory, the controller is more frequently located in the center of the topology. This echoes the results in the static controller placement analysis.

VI. Infrequent Controller Relocation

Throughout this paper, we assume that a new controller placement occurs at every time slot. This can be done by ensuring that the controller placement algorithm depends only on information that is available to each node in the network. Thus, there is no additional communication overhead required to compute the controller placement. Thus, as all nodes use the same policy, they will arrive at the same decision. However, there may be an additional cost associated with relocating the controller due to the computation required, and relocating the controller at every time-slot may not be practical. Therefore, in this section, we consider the case in which the controller placement occurs less often. Note that while infrequent controller placement may reduce the cost associated with relocation, there is an increased occurrence of the underlying unfairness caused by the delayed arrival of network state information to the fixed controller, as discussed in Section IV.

Consider the controller placement problem, in which the controller is relocated every N time slots. As discussed in Section IV, the throughput region is not affected by infrequent controller placement. Lemma 1 shows that any arrival rate $\lambda \in \Lambda$ corresponds to a stationary policy which stabilizes the system. The throughput region Λ is formed by a time-sharing between controller placements. Consequently, the frequency of changing the controller placement does not affect throughput, but rather the overall fraction of time spent in each controller location.

The DCPS policy of Section IV extends directly to the case of infrequent controller placement as follows.

Algorithm 3 DCPS With Infrequent Controller Placement (DCPS-N) Policy

1: At each time $t = k * N$, choose a controller by solving the following optimization as a function of the delayed queue backlogs $Q_i(kN - \tau_Q)$.

$$r^* = \arg\max_r \left(\sum_{\mathbf{s}\in\mathcal{S}} \mathbf{P}_{\mathbf{S}}(\mathbf{s}) \max_i Q_i(kN - \tau_Q) p_{s_i,1}^{d_r(i)} \right) \quad (17)$$

where $P_S(\mathbf{s})$ is the steady state probability of the channel-state process.

2: At subsequent time slots $t = kN + j$, the controller observes CSI $S(kN + j - d_{r^*}(i)) = \mathbf{s}$.

3: Controller schedules the following queue to transmit.

$$i^* = \arg\max_i Q_i(kN - \tau_Q) p_{\mathbf{s}_i,1}^{d_{r^*}(i)} \quad (18)$$

Theorem 5: For any arrival rate λ, and $\epsilon > 0$ satisfying $\lambda + \epsilon\mathbf{1} \in \Lambda$, the DCPS-N policy given in Algorithm 3 stabilizes the system if $\tau_Q \geq d_{max} + T_{SS}(\epsilon)$ for $T_{SS}(\epsilon)$ defined in (12).

The DPCS-N policy differs from the DCPS policy in that controller placement decisions are only made in time slots which are multiples of N, but the controller placement calculation is the same as in DCPS. The scheduling portion of DCPS-N uses the delayed QLI with respect to the time at which the controller was placed, rather than the current time slot. This additional delay in QLI does not affect the throughput optimality of the policy. The proof of Theorem 5 follows similarly to the proof of Theorem 2, except using a T-slot drift argument at every time slot $t = kN$ rather than every time slot.

VII. Conclusion

We studied the impact of controller location on achievable throughput in wireless networks. First, we formulated the static controller placement problem whereby a controller location is chosen a-priori with the objective of maximizing network throughput. Then, we consider dynamically placing controllers, using QLI to relocate the controller to heavily congested areas of the network based on queue-backlog information. We characterize the throughput region under dynamic controller placement, and propose a throughput optimal joint controller placement and scheduling policy. This policy uses significantly delayed QLI to place the controllers, and the CSI available at the controller to schedule links. These results were extended to controller placement policies which use both delayed CSI and delayed QLI. Using CSI in controller placement improves the throughput region, as well as eliminates the requirement for using highly delayed QLI.

An interesting observation is that when the controller placement depends only on delayed QLI, the throughput optimal policy uses a delayed version of the QLI, even if more recent QLI is available. This is due to the fact that QLI correlated with channel states, especially when there is a high degree of memory in the system. A related result was observed in [11] in the context of scheduling with hidden channel state information where, it was observed that the optimal scheduling decision uses highly delayed QLI to maximize throughput.

Appendix

A. Proof of Lemma 1

Lemma 1: Suppose there exists a policy $\mathcal{P} \in \Pi$ that stabilizes the network for all $\lambda \in \Lambda$. Then, there exists a β_r and $\alpha_i^r(\mathbf{s})$ such that (9) has a solution with $\epsilon^* \geq 0$.

Proof: Suppose the system is stabilized with some control policy $\mathcal{P}$, consisting of functions $\beta_r(t)$, which chooses a controller independent of channel state, and $\alpha_i^r(t)$ which chooses a link activation based on delayed CSI at the controller. Without loss of generality, let $\beta_r(t)$ be an indicator function signaling whether node r is the controller at time t, and let $\alpha_i(t)$ be an indicator signaling whether link i is scheduled for transmission at time t. Under any such scheme, the following relationship holds between arrivals, departures, and backlogs for each queue:

$$\sum_{\tau=1}^{t} A_i(\tau) \leq Q_i(t) + \sum_{\tau=1}^{t} \mu_i(\beta_r(\tau), \alpha_i^r(\tau)), \quad (19)$$

where μ_i is the service rate of the i^{th} queue as a function of the control decisions. Expanding μ_i in terms of the decision variables $\beta_r(t)$ and $\alpha_i^r(t)$ yields

$$\sum_{\tau=1}^{t} A_i(\tau) \leq Q_i(t) + \sum_{\tau=1}^{t}\sum_{r=1}^{M} \beta_r(\tau)\alpha_i^r(\tau) \times \mathbb{E}[S_i(\tau)|S_i(\tau - d_r(i))]. \quad (20)$$

Let T_r be the subintervals of $[1, t]$ over which r is the controller. Further, let T_S^r be the subintervals of T_r such that the controller r observes delayed CSI $S(t - d_r(i)) = S$. Let $|T_r|$ and $|T_S^r|$ be the aggregate length of these intervals. Since the arrival and the channel state processes are ergodic, and the number of channel states and queues is finite, there exists a time t_1 such that for all $t \geq t_1$, the empirical average arrival rates and state occupancy fractions are within ϵ of their expectations.

$$\frac{1}{t}\sum_{\tau=1}^{t} A_i(\tau) \geq \lambda_i - \epsilon \quad (21)$$

$$\frac{1}{|T_r|}|T_{\mathbf{S}}^r| \leq \mathbf{P}(S_i(t) = \mathbf{S}|r) + \epsilon = \mathbf{P}(S_i(t) = \mathbf{S}) + \epsilon \quad (22)$$

The above equations hold with probability 1 from the strong law of large numbers [12]. Furthermore, since the system is stable under the policy $\mathcal{P}$, [2] shows that there exists a V such that for an arbitrarily large t,

$$\mathbf{P}\left(\sum_{i=1}^{M} Q_i(t) \leq V\right) \geq \frac{1}{2}. \quad (23)$$

Thus, let t be a large time index such that $t \geq t_1$ and $\frac{V}{t} \leq \epsilon$. If $\sum_{i=1}^{M} Q_i(t) \leq V$, the inequality in (20) can be rewritten by dividing by t.

$$\sum_{\tau=1}^{t} \frac{A_i(\tau)}{t} \leq \frac{V}{t} + \frac{1}{t}\sum_{\tau=1}^{t}\sum_{r=1}^{M} \beta_r(\tau)\alpha_i(\tau) \times \mathbb{E}[S_i(\tau)|S_i(t-d_r(\tau))]. \tag{24}$$

$$\lambda_i - \epsilon \leq \epsilon + \sum_{r=1}^{M}\frac{1}{t}\sum_{\tau=1}^{t}\beta_r(\tau)\alpha_i(\tau) \times \mathbb{E}[S_i(\tau)|S_i(t-d_r(\tau))]. \tag{25}$$

The lower bound in (25) follows from (21). Since $\beta_r(\tau) = 1$ if and only if $\tau \in T_r$, the inequality in (25) is equivalent to

$$\lambda_i \leq 2\epsilon + \sum_{r=1}^{M}\frac{1}{t}\sum_{\tau\in T_r}\alpha_i(\tau)\mathbb{E}[S_i(\tau)|S_i(\tau - d_r(i))] \tag{26}$$

$$= 2\epsilon + \sum_{r=1}^{M}\frac{|T_r|}{t}\frac{1}{|T_r|}\sum_{\tau\in T_r}\alpha_i(\tau)\mathbb{E}[S_i(\tau)|S_i(\tau - d_r(i))] \tag{27}$$

$$= 2\epsilon + \sum_{r=1}^{M}\overline{\beta}_r\frac{1}{|T_r|}\sum_{\tau\in T_r}\alpha_i(\tau)\mathbb{E}[S_i(\tau)|S_i(\tau - d_r(i))] \tag{28}$$

The last equation follows from defining

$$\overline{\beta}_r \triangleq \frac{|T_r|}{t}, \tag{29}$$

the empirical fraction of time that r is the controller. Now, break the summation over T_r into separate summations over the sub-intervals T_S^r for each observed S. Note that $\mathbb{E}[S_i(\tau)|S_i(\tau - d_r(i))]$ is the k-step transition probability of the Markov chain in Figure 1 for $k = d_r(i)$.

$$\lambda_i \leq 2\epsilon + \sum_{r=1}^{M}\overline{\beta}_r\sum_{S\in\mathcal{S}}\frac{1}{|T_r|}\sum_{\tau\in T_S^r}\alpha_i(\tau)p_{S_i,1}^{d_r(i)} \tag{30}$$

$$= 2\epsilon + \sum_{r=1}^{M}\overline{\beta}_r\sum_{S\in\mathcal{S}}\frac{|T_S^r|}{|T_r|}\frac{1}{|T_S^r|}\sum_{\tau\in T_S^r}\alpha_i(\tau)p_{S_i,1}^{d_r(i)} \tag{31}$$

$$= 2\epsilon + \sum_{r=1}^{M}\overline{\beta}_r\sum_{S\in\mathcal{S}}\frac{|T_S^r|}{|T_r|}\overline{\alpha}_i^r(S)p_{S_i,1}^{d_r(i)} \tag{32}$$

$$\leq \sum_{r=1}^{M}\overline{\beta}_r\sum_{S\in\mathcal{S}}\mathbf{P}(S_i(t) = S)\overline{\alpha}_i^r(S)p_{S_i,1}^{d_r(i)} + \epsilon(2 + |\mathcal{S}|) \tag{33}$$

where (32) follows from defining the fraction of time that link i is scheduled given r and S as

$$\overline{\alpha}_i^r(S) \triangleq \frac{1}{|T_S^r|}\sum_{\tau\in T_{\mathbf{S}}^r}\alpha_i(\tau), \tag{34}$$

and (33) follows from (22) and the fact that controller placement is independent of channel state. Because the control functions satisfy $\sum_r \beta_r(t) \leq 1$ and $\sum_i \alpha_i(t) \leq 1$, it follows that $\overline{\beta_r}$ and $\overline{\alpha_i^r}$ satisfy those same constraints. Furthermore, the fraction of time node r is the controller, $\overline{\beta}_r$, is independent of the CSI.

The above inequality assumes $\sum_{i=1}^{M} Q_i(t) \leq V$, which holds with probability greater than $\frac{1}{2}$ by (23). Hence, there exists a set of stationary control decisions β_r and α_r^i satisfying the necessary constraints such that (33) holds for all i. If there did not exist such a stationary policy, than this inequality would hold with probability 0. Therefore, λ is arbitrarily close to a point in the region Λ, implying the constraints imposed by Λ are necessary for stability. ∎

Corollary 1: Consider a system of M nodes, where only one can transmit at each time. Assume the controller has full knowledge of its own channel state and d-slot delayed CSI for each other channel, as in Figure 7. At time t, the DCPS policy places the controller at the node with the largest backlog at time $t - \tau_Q$.

$$r^* = \arg\max_r Q_r(t - \tau_Q) \tag{35}$$

Proof: Recall the optimal policy at each time is the DCPS policy in Theorem 2, where the controller is chosen to maximize the expected maximum weight schedule. Let $Q^{(1)}, \ldots, Q^{(M)}$ be the ordering of delayed queue lengths $\mathbf{Q}(t - \tau_Q)$, such that $Q^{(1)} \geq Q^{(2)} \ldots \geq Q^{(M)}$. Consider placing the controller at the node corresponding to $Q^{(1)}$. Let k_2 be the largest index i such that $Q^{(i)}p_{11}^d \geq Q^{(2)}p_{01}^d$. The expected max-weight is a random variable, which takes values determined by the CSI. Let MW_i be the weight of the schedule activated by a controller at the i^{th} largest queue. The expected max weight of a controller at $Q^{(1)}$ is given by

$$\mathbb{E}[MW_1] = Q^{(1)}\pi + \sum_{i=2}^{k_2} Q^{(i)}\pi(1-\pi)^{i-1}p_{11}^d + Q^{(2)}p_{01}^d(1-\pi)^{k_2} \tag{36}$$

Equation (36) is derived as follows. Since $Q^{(1)}$ is the largest queue, if that channel is ON the max-weight policy transmits from $Q^{(1)}$. If that channel is OFF, then the belief of that channel is zero, and it will not be used. Transmitting from $Q^{(j)}$ is optimal only if $Q^{(i)}$ is OFF for all $j > i$, since $Q^{(i)}$ are sorted in decreasing order. By the definition of k_2, for $j > k_2$, $Q^{(j)}p_{11}^d < Q^{(2)}p_{01}^d$, so it is optimal to schedule $Q^{(2)}$ when $Q^{(i)}$ is OFF for all $i \leq k_2$.

Now consider placing the controller at the node corresponding to queue $Q^{(j)}$, for $j \geq 2$. Let k_1 be the largest index such that $Q^{(k_1)}p_{11}^d \geq Q^{(1)}p_{01}^d$. Similarly, define k_j' to be the largest index such that $Q^{(k_j')}p_{11}^d \geq Q^{(j)}$. The expected max weight is computed for two cases, depending on the relationship between k_1 and k_j'.

First, consider the case where $Q^{(j)} \leq Q^{(1)}p_{01}^d$, i.e. $k_1 \leq k_j'$. In this case, it is never optimal to transmit over the channel corresponding to $Q^{(j)}$, regardless of its delayed CSI. The expected max-weight is given by

$$\mathbb{E}[MW_j] = \pi p_{11}^d\sum_{i=1}^{k_1} Q^{(i)}(1-\pi)^{i-1} + p_{01}^d Q^{(1)}(1-\pi)^{k_1} \tag{37}$$

Compare the expected max weight between the controller at $Q^{(1)}$ and $Q^{(j)}$.

$$\begin{aligned}
&\mathbb{E}[MW_1 - MW_j] \\
&= Q^{(1)}\pi + \sum_{i=2}^{k_2} Q^{(i)}\pi(1-\pi)^{i-1}p_{11}^d + Q^{(2)}p_{01}^d(1-\pi)^{k_2} \\
&\quad - Q^{(1)}\pi p_{11}^d - \sum_{i=2}^{k_1} Q^{(i)}(1-\pi)^{i-1}\pi p_{11}^d \\
&\quad - Q^{(k_1)}(1-\pi)^{k_1}p_{01}^d \qquad (38)\\
&= Q^{(1)}\pi(1-p_{11}^d) + p_{11}^d \sum_{i=k_1+1}^{k_2} Q^{(i)}\pi(1-\pi)^{i-1} \\
&\quad + Q^{(2)}p_{01}^d(1-\pi)^{k_2} - Q^{(1)}(1-\pi)^{k_1}p_{01}^d \qquad (39)\\
&\geq Q^{(1)}\pi p_{10}^d - Q^{(1)}(1-\pi)^{k_1}p_{01}^d + Q^{(2)}p_{01}^d(1-\pi)^{k_2} \qquad (40)\\
&= Q^{(1)}\pi p_{10}^d - Q^{(1)}\pi(1-\pi)^{k_1-1}p_{10}^d + Q^{(2)}p_{01}^d(1-\pi)^{k_2} \qquad (41)\\
&\geq Q^{(1)}\pi p_{10}^d\big(1-(1-\pi)^{k_1-1}\big) + Q^{(2)}p_{01}^d(1-\pi)^{k_2} \geq 0 \qquad (42)
\end{aligned}$$

where (40) follows from $Q^{(i)} \geq 0$, and (41) follows from the identity $\pi p_{10}^d = (1-\pi)p_{01}^d$.

Now consider the case where $Q^{(j)} \geq Q^{(1)}p_{01}^d$. In this case, there exists a state such that it is optimal to transmit over $Q^{(j)}$. The max-weight expression is given by

$$\begin{aligned}
\mathbb{E}[MW_j] &= \pi p_{11}^d \sum_{i=1}^{k_j'} Q^{(i)}(1-\pi)^{i-1} + Q^{(j)}\pi(1-\pi)^{k_j'} \\
&\quad + \pi p_{11}^d \sum_{i=k_j+1}^{k_1} Q^{(i)}(1-\pi)^{i} \\
&\quad + p_{01}^d Q^{(1)}(1-\pi)^{k_1+1} \qquad (43)
\end{aligned}$$

Comparing the expected max weight between the controller at $Q^{(1)}$ and $Q^{(j)}$.

$$\begin{aligned}
&\mathbb{E}[MW_1 - MW_j] \\
&= Q^{(1)}\pi + \sum_{i=2}^{k_2} Q^{(i)}\pi(1-\pi)^{i-1}p_{11}^d + Q^{(2)}p_{01}^d(1-\pi)^{k_2} \\
&\quad - Q^{(1)}\pi p_{11}^d - \pi p_{11}^d \sum_{i=2}^{k_j'} Q^{(i)}(1-\pi)^{i-1} - Q^{(j)}\pi(1-\pi)^{k_j'} \\
&\quad - \pi p_{11}^d \sum_{i=k_j'+1}^{k_1} Q^{(i)}(1-\pi)^{i} - p_{01}^d Q^{(1)}(1-\pi)^{k_1+1} \qquad (44)\\
&= Q^{(1)}\pi p_{10}^d - Q^{(j)}\pi(1-\pi)^{k_j'}p_{10}^d \\
&\quad + \pi p_{11}^d \sum_{i=k_j'+1}^{k_1} Q^{(i)}\pi(1-\pi)^{i} + p_{11}^d \sum_{i=k_1+1}^{k_2} Q^{(i)}\pi(1-\pi)^{i-1} \\
&\quad + Q^{(2)}p_{01}^d(1-\pi)^{k_2} - Q^{(1)}(1-\pi)^{k_1+1}p_{01}^d \qquad (45)
\end{aligned}$$

Equation (45) follows from combining like terms, and breaking up the summation over the interval $i = [2, k_2]$ into three intervals: $[2, k_j']$, $[k_j'+1, k_1]$, and $[k_1+1, k_2]$, as well as an additional term for $Q^{(j)}$. The summations are bounded as follows

$$\begin{aligned}
&\pi p_{11}^d \sum_{i=k_j'+1}^{k_1} Q^{(i)}\pi(1-\pi)^{i-1} + p_{11}^d \sum_{i=k_1+1}^{k_2} Q^{(i)}\pi(1-\pi)^{i-1} \\
&\quad \geq \pi Q^{(1)}p_{01}^d \sum_{i=k_j'+1}^{k_1} \pi(1-\pi)^{i-1} \\
&\qquad + Q^{(2)}p_{01}^d \sum_{i=k_1+1}^{k_2} \pi(1-\pi)^{i-1} \qquad (46)\\
&\quad = \pi Q^{(1)}p_{01}^d\big((1-\pi)^{k_j'} - (1-\pi)^{k_1}\big) \\
&\qquad + Q^{(2)}p_{01}^d\big((1-\pi)^{k_1} - (1-\pi)^{k_2}\big) \qquad (47)
\end{aligned}$$

The inequality in (47) follows from the fact that $Q^{(1)}p_{01}^d \leq Q^{(i)}p_{11}^d$ for $i \leq k_1$, and $Q^{(2)}p_{01}^d \leq Q^{(i)}p_{11}^d$ for $i \leq k_2$. Plugging this into equation (45)

$$\begin{aligned}
&\mathbb{E}[MW_1 - MW_j] \\
&\geq Q^{(1)}\pi p_{10}^d - Q^{(j)}\pi(1-\pi)^{k_j'}p_{10}^d + Q^{(2)}p_{01}^d(1-\pi)^{k_2} \\
&\quad - Q^{(1)}(1-\pi)^{k_1+1}p_{01}^d \\
&\quad + \pi Q^{(1)}p_{01}^d\big((1-\pi)^{k_j'} - (1-\pi)^{k_1}\big) \\
&\quad + Q^{(2)}p_{01}^d\big((1-\pi)^{k_1} - (1-\pi)^{k_2}\big) \\
&\geq Q^{(1)}\pi p_{10}^d - Q^{(j)}\pi(1-\pi)^{k_j'}p_{10}^d - Q^{(1)}(1-\pi)^{k_1+1}p_{01}^d \\
&\quad + Q^{(2)}p_{01}^d(1-\pi)^{k_1} \qquad (48)\\
&\geq Q^{(1)}\pi p_{10}^d(1-(1-\pi)^{k_1}) \\
&\quad - Q^{(j)}\pi(1-\pi)^{k_j'}p_{10}^d + Q^{(2)}\pi p_{10}^d(1-\pi)^{k_1-1} \qquad (49)\\
&\geq Q^{(2)}\pi p_{10}^d(1-(1-\pi)^{k_1}) \\
&\quad - Q^{(2)}\pi(1-\pi)^{k_j'}p_{10}^d + Q^{(2)}\pi p_{10}^d(1-\pi)^{k_1-1} \qquad (50)\\
&= Q^{(2)}\pi p_{10}^d\big[1-(1-\pi)^{k_1} - (1-\pi)^{k_j'} + (1-\pi)^{k_1-1}\big] \geq 0 \qquad (51)
\end{aligned}$$

The inequality in (48) follows from $k_j' \geq k_1$, and canceling out $Q^{(2)}$ terms, (49) follows from the identity $\pi p_{10}^d = (1-\pi)p_{01}^d$, and (50) holds since $Q^{(1)} \geq Q^{(2)} \geq Q^{(j)}$. Therefore, for all $j \geq 2$, placing the controller at the node corresponding to $Q^{(j)}$ results in a lower expected max weight than placing at the node corresponding to the longest queue. Thus, placing the controller at the longest queue is the optimal controller placement policy. ■

REFERENCES

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

[2] M. J. Neely, E. Modiano, and C. E. Rohrs, "Dynamic power allocation and routing for time-varying wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 1, pp. 89–103, Jan. 2005.

[3] M. J. Neely, "Stochastic network optimization with application to communication and queueing systems," *Synthesis Lectures Commun. Netw.*, vol. 3, no. 1, pp. 1–211, 2010.

[4] K. Kar, X. Luo, and S. Sarkar, "Throughput-optimal scheduling in multichannel access point networks under infrequent channel measurements," *IEEE Trans. Wireless Commun.*, vol. 7, no. 7, pp. 2619–2629, Jul. 2008.

[5] L. Ying and S. Shakkottai, "On throughput optimality with delayed network-state information," *IEEE Trans. Inf. Theory*, vol. 57, no. 8, pp. 5116–5132, Aug. 2011.

[6] L. Ying and S. Shakkottai, "Scheduling in mobile ad hoc networks with topology and channel-state uncertainty," *IEEE Trans. Autom. Control*, vol. 57, no. 10, pp. 2504–2517, Oct. 2012.

[7] M. Johnston and E. Modiano, "A new look at wireless scheduling with delayed information," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2015, pp. 1407–1411.

[8] M. Johnston, "The role of control information in wireless link scheduling," Ph.D. dissertation, Dept. Elect. Eng., Massachusetts Inst. Technol., Cambridge, MA, USA, 2015.

[9] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*. New York, NY, USA: Dover, 1998.

[10] M. S. Daskin, *Network and Discrete Location: Models, Algorithms, and Applications*. Hoboken, NJ, USA: Wiley, 2011.

[11] M. Johnston and E. Modiano, "Scheduling over time varying channels with hidden state information," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2015, pp. 1402–1406.

[12] D. P. Bertsekas, *Introduction to Probability*, D. P. Bertsekas and J. N. Tsitsiklis, Eds. Belmont, MA, USA: Athena Scientific, 2002.

[13] M. J. Neely, "Dynamic power allocation and routing for satellite and wireless networks with time varying channels," Ph.D. dissertation, Dept. Elect. Eng. Comput. Sci., Massachusetts Inst. Technol., Cambridge, MA, USA, 2003.

Matthew Johnston received the B.S. degree in electrical engineering and computer science from the University of California at Berkeley, Berkeley, CA, USA, in 2008, and the M.S. and Ph.D. degrees in electrical engineering from the Massachusetts Institute of Technology in 2010 and 2015, respectively. Since 2015, he has been a Research Scientist with Boeing Research and Technology. His research is on communication networks and protocols with an emphasis on network design, network control, and network robustness.

Eytan Modiano (S'90–M'93–SM'00–F'12) received the B.S. degree in electrical engineering and computer science from the University of Connecticut at Storrs in 1986, and the M.S. and Ph.D. degrees in electrical engineering from the University of Maryland at College Park, College Park, MD, USA, in 1989 and 1992, respectively. He was a Naval Research Laboratory Fellow from 1987 to 1992 and a National Research Council Post-Doctoral Fellow from 1992 to 1993. From 1993 to 1999, he was with the MIT Lincoln Laboratory. Since 1999, he has been serving on the faculty of MIT, where he is currently a Professor with the Department of Aeronautics and Astronautics and the Laboratory for Information and Decision Systems. His research is on communication networks and protocols with emphasis on satellite, wireless, and optical networks. He is an Associate Fellow of the AIAA, and served on the IEEE Fellows committee. He was a co-recipient of the MobiHoc 2016 Best Paper Award, the WiOpt 2013 Best Paper Award, and the SIGMETRICS 2006 Best Paper Award. He is currently an Editor-at-Large for the IEEE/ACM TRANSACTIONS ON NETWORKING, and served as an Associate Editor of the IEEE TRANSACTIONS ON INFORMATION THEORY and IEEE/ACM TRANSACTIONS ON NETWORKING. He was the Technical Program Co-Chair for the IEEE WiOpt 2006, the IEEE INFOCOM 2007, the ACM MobiHoc 2007, and the DRCN 2015.