

Reduced-State, Optimal Medium Access Control for Wireless Data Collection Networks

Avinash Mohan
Dept. of ECE,
Indian Institute of Science,
Bangalore, India.
avinashmohan@iisc.ac.in

Aditya Gopalan
Dept. of ECE,
Indian Institute of Science,
Bangalore, India.
aditya@iisc.ac.in

Anurag Kumar
Dept. of ECE,
Indian Institute of Science,
Bangalore, India.
anurag@iisc.ac.in

Abstract—Motivated by medium access control for resource-challenged wireless sensor networks whose main purpose is data collection, we consider the problem of queue scheduling with reduced queue state information. In particular, we consider a model with N sensor nodes, with pair-wise dependence, such that nodes i and $i + 1$, $1 \leq i \leq N - 1$ cannot transmit together. For $N = 3, 4$, and 5 , we develop new throughput-optimal scheduling policies requiring only the empty-nonempty state of each queue, and also revisit previously proposed policies to rigorously establish their throughput- and delay-optimality. For $N = 3$, there exists a sum-queue length optimal scheduling policy that requires only the empty-nonempty state of each queue. We show, however, that for $N \geq 4$, there is no scheduling policy that uses only the empty-nonempty states of the queues and is sum-queue length optimal uniformly over all arrival rate vectors. We then extend our results to a more general class of interference constraints, namely, a star of cliques. Our throughput-optimality results rely on two new arguments: a Lyapunov drift lemma specially adapted to policies that are queue length-agnostic, and a priority queueing analysis for showing strong stability.

Our study throws up some counterintuitive conclusions: 1) knowledge of queue length information is not necessary to achieve optimal throughput/delay performance for a large class of interference networks, 2) it is possible to perform throughput-optimal scheduling by merely knowing whether queues in the network are empty or not, and 3) it is also possible to be throughput-optimal by *not* always scheduling the maximum possible number of nonempty queues. We also show the results of numerical experiments on the performance of queue length agnostic scheduling vs. queue length aware scheduling, on several interference networks.

Index Terms—Wireless Sensor Networks, Medium Access Control (MAC) protocols, Optimal Polling, Delay Minimization, Hybrid MACs, Self-Organizing Networks, Internet of Things (IoT).

I. INTRODUCTION

The Internet of Things (IoT) paradigm is expected to make possible applications where vast numbers of devices coexist on a communication network. A typical example is a large-scale wireless sensor network comprising low-cost sensors that forward measurements from their respective locations. Nodes in these networks are typically energy-limited, and must communicate over a common, interference-constrained wireless medium. Extracting high performance from such resource-challenged wireless access networks entails the design of low coordination media access control (MAC) schemes, that can

apportion resources dynamically while keeping information exchange overheads (and thus energy) down to a minimum.

A natural approach for dynamic resource allocation is to use backlog or queue length information to schedule transmissions. One of the seminal contributions to scheduling in constrained queueing systems is the work of Tassiulas and Ephremides [1]. This paper introduces the model of a wireless network as a network of queues with pair-wise scheduling constraints (corresponding to wireless interferences, half-duplex operation, etc.), and several flows over the network, each with its ingress queue and egress queue. The pair-wise constraints are represented by an *interference graph* with the queues as the nodes and the pair-wise scheduling constraints being the edges. With stochastic arrivals to each flow to be routed from their ingress to egress points, the authors derive MaxWeight, a *centralized* scheduling algorithm which requires the queue lengths of all nodes. MaxWeight is known to be throughput-optimal, i.e., it stochastically stabilizes all queues under any stabilizable arrival rate.

Attempts to decentralize MaxWeight have included approximations based on message passing between nodes [2], [3], or using queue lengths to modulate backoff parameters in CSMA and ALOHA [4], [5]. Both of these methods, while being throughput-optimal, suffer from poor delay performance.

Another method to reduce the amount of information required for scheduling was proposed by Tassiulas and Ephremides [6], where, for two classes of constrained queueing systems, algorithms relying only on the empty-nonempty state of queues were proposed and analysed for delay performance. Our interest lies in the second half of [6], where a scheduling algorithm was proposed for a system of N parallel queues in which adjacent queues cannot be served simultaneously. The paper gave the optimal policy for $N = 3$. This was extended to $N = 4$ by Ji et al [7] where the heavy-traffic delay-optimality of the proposed policy was proved. One of the contributions of our work in this paper is to reexamine the delay and throughput optimality proofs for these policies. It is not yet clear if it is even possible to extend these algorithms to general wireless networks while preserving their performance guarantees.

A third class of strategies has focused on completely uncoordinated medium access, in contrast to methods using network

state information. Here the focus is on improving the saturation throughput of Abramson’s ALOHA protocol beyond $(1/e)$, without any queue length knowledge. The main idea is to allow collisions to take place, but use physical layer techniques like successive interference cancellation to decode the garbled messages over multiple time slots [8], [9]. These techniques like, Abramson’s ALOHA, are not throughput optimal (in that they can hope to achieve only the saturation throughput), but allow uncoordinated access. In contrast, we are interested in contention-free, low coordination MAC schemes for high throughput.

A. Our Contributions and Organization

In this work, we seek to develop throughput-optimal and low-delay¹ medium access control protocols that rely only on reduced state information, namely the empty-nonempty states of queues. Our specific contributions are as follows.

- We study in detail the space of reduced-state scheduling policies for path (linear) interference networks of sizes 3, 4 and 5 nodes (Sections V and VI). Our analysis, along with giving new throughput-optimal reduced-state scheduling policies, rigorously establishes the throughput- and delay-optimality of the policies proposed earlier in [6] and [7]. It also shows, rather surprisingly, that throughput-optimality is not confined only to Maximum Egress Rate (MER) policies – policies that schedule the maximum possible number of nonempty queues at each instant – and that delay-optimal reduced-state scheduling is impossible in path graphs of size 4 or higher (Sec. VI-C).
- A *clique* in an interference graph models a collection of wireless links each of whose transmissions interferes with those of every other link in the clique. We extend the theory and insights from scheduling in path interference graph networks to a more general class of interference graphs that consist of an *arbitrary* number of *arbitrary-sized* cliques connected to a central clique (Fig. 3). We call this the “star-of-cliques” model and such networks naturally arise in IoT applications [10]. We present new throughput-optimal reduced-state scheduling algorithms for these networks (Sec. VII).
- We then present numerical results (Sec. VIII) showing the performance of our proposed policies, and comparisons with standard, high-overhead state-based policies such as the MaxWeight- α family [11].
- We then provide some remarks on decentralized implementation of our reduced-state scheduling policies (Sec. IX), in which nodes in the vicinity of a transmitting node overhear its transmissions and schedule their own transmissions in a contention-free manner. Finally, we conclude the paper and present directions for future work.

II. SYSTEM MODEL

In sections III to VI the system we study is modelled by N parallel queues. Each queue models a radio link in a

¹By low delay, we mean a low sum queue length across the system.

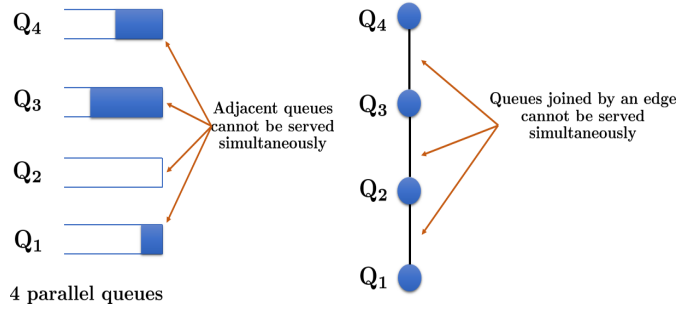


Figure 1: The basic queuing system with $N = 4$ queues (left) and its interference graph (right). **Adjacent queues cannot be served simultaneously.**

wireless network and represents a *transmitter-receiver pair*. The scheduling constraints are the same as the second model in [6], i.e., Queue i and Queue $i + 1$ cannot be served simultaneously for $1 \leq i \leq N - 1$. The interference graph [12] associated with the system is a *path graph*.

A *collocated star* network consists of a cluster of sensor nodes and a sink node to which all the sensors must send their data. By “collocated,” we mean that only one sensor can successfully send its data to the sink at any time. In sections VI-VII, we will consider a network comprising several *collocated star* networks such that the interference between them can be represented by a star; see Fig 2. For such networks, we will utilize the theory developed in sections III to VI to develop decentralized, throughput optimal, and low delay medium access control for such networks.

Time is assumed to be slotted and the *leading edges* of the slots are indexed $0, 1, 2, \dots$. Arrivals are embedded at slot boundaries, $t = 0, 1, 2, \dots$, with the number of packets arriving to Queue i at time t being denoted by the random variable $A_i(t)$. $A_i(t)$ is assumed iid² across time and independent across queues and is modelled as a Bernoulli random variable with mean λ_i . However, we will partially remove this restriction to include batch arrivals in Sec. VII. We use $\mathbf{Q}(t) = [Q_1(t), \dots, Q_N(t)]^T$ to denote the vector of all queue lengths at time t . The queue length process is embedded at the beginnings of time slots, so $Q_i(t), t \geq 0$, is measured at $t+$, i.e., just *after* the arrival. Packet transmissions are assumed to take exactly one time slot and succeed with probability³ 1. The random variable indicating the departure of a packet from Queue i at time t , $D_i(t)$, is such that $D_i(t) = 1$ if and only if Queue i is scheduled in slot t and $Q_i(t) > 0$, else $D_i(t) = 0$; here, the departure is assumed to end just before the leading edge of slot $(t + 1)$, i.e., at $(t + 1)-$. The *offered service* process to Queue i , $\{S_i(t), t \geq 0\}$, is defined as follows: $S_i(t) = 1$ whenever Queue i is given access to the channel, so that $D_i(t) = S_i(t)\mathbb{I}_{\{Q_i(t) > 0\}}, \forall t \geq 0, 1 \leq i \leq N$. The interference constraints enforce the rule $S_i(t) + S_{i+1}(t) \leq 1, \forall t \geq 0, 1 \leq i \leq N - 1$. The vector $\mathbf{S}(t) := [S_1(t), \dots, S_N(t)]$ is called an *activation vector*. With

²“iid” stands for *independent and identically distributed*.

³The effects of fading will be studied in future work.

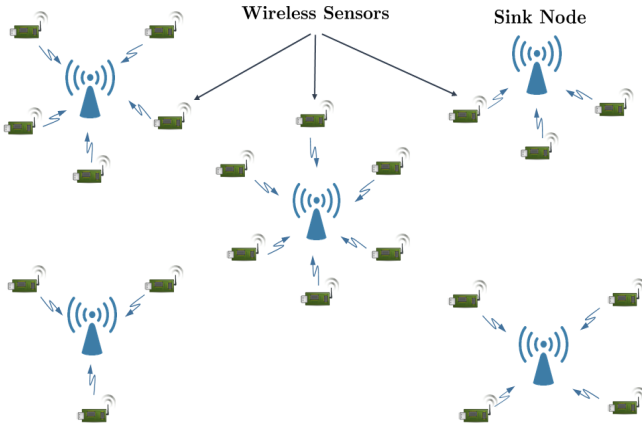


Figure 2: A wireless sensor network with multiple sinks. We do not impose any restrictions on the number of peripheral cliques or the number of nodes per clique. Fig. 3 shows the associated interference graph.

this embedding, for every i ,

$$\begin{aligned} Q_i(t+1) &= Q_i(t) - D_i(t) + A_i(t+1) \\ &= (Q_i(t) - S_i(t))^+ + A_i(t+1), \quad \forall t \geq 0. \end{aligned}$$

Denote by $\mathbf{i}(t) := [\mathbb{I}_{\{Q_1(t)>0\}}, \dots, \mathbb{I}_{\{Q_N(t)>0\}}]^T$ the system's *occupancy vector* at time t , i.e., the empty-nonempty state of each of the N queues. Let $V \subset \{0, 1\}^N$ be the set of all activation vectors. A scheduling policy π in general is defined as a time-indexed sequence of functions $\{\mu_0, \mu_1, \dots\}$ that decides which queues are allowed to transmit in each slot as a function of the available history \mathcal{H}_t , which comprises the past states and actions known to the controller, and the current known state. Specifically, $\mu_t : \mathcal{H}_t \rightarrow V$ is an $N \times 1$ vector, and $S_i(t) = \mu_t(i)$. By *stability* of the process $\{\mathbf{Q}(t), t \geq 0\}$ we will mean that

$$\limsup_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} \sum_{i=1}^N \mathbb{E}_\pi Q_i(t) < \infty. \quad (1)$$

This condition is also known in the literature as *strong stability* [13]. A policy π that ensures (1) is said to be stabilizing, and an arrival rate vector for which a stabilizing policy exists is called *stabilizable*. From [1], we know that for this network, the set of stabilizable rates is the interior of the set

$$\Lambda_N := \{\boldsymbol{\lambda} \in \mathbb{R}_+^N \mid \lambda_i + \lambda_{i+1} \leq 1, \forall 1 \leq i \leq N-1\}. \quad (2)$$

A policy that is stabilizing for every arrival rate vector in Λ^o is called *throughput-optimal* (T.O.).

III. MAXIMUM EGRESS RATE (MER) SCHEDULING

Definition. If, in every slot t , a policy schedules as many nonempty queues for transmission as the interference constraints allow, then the policy is said to be a *Maximum Egress Rate (MER)* policy.

For example, if $N = 7$, and $\mathbf{Q}(t) = [1, 2, 0, 0, 4, 3, 3]$, a policy that schedules queues 1, 5 and 7 or 2, 5 and 7 is MER while a policy that schedules queues 1, 7 only, is not MER.

Lemma 4.1 in [6] defines a class of policies that is more restrictive than MER as follows:

- 1) In every slot, the policy should serve the largest number of nonempty queues subject to the interference constraints, i.e., it *should be MER*, and
- 2) the policy should prioritize “inner”, more constrained, queues over “outer” queues while breaking ties. For example, in Fig. 1, queues 2 and 3 must be prioritized over queues 1 and 4.

We will see later, that the second condition helps reduce delay by prioritizing “inner” queues, i.e., those other than Queue 1 or Queue N . We will see that in several interference graphs, scheduling based on the occupancy vector $\mathbf{i}(t)$ is sufficient not only for stability but also for delay-optimality.

Notation: Classes of scheduling policies

- $\Pi^{(N)}$: the class of all policies.
- $\Gamma_M^{(N)}$: the class of all MER policies.
- $\Pi_M^{(N)}$: the class of all policies that take only the occupancy vector $\mathbf{i}(t)$ as input and activate the largest number of non empty queues in every slot, i.e., MER policies that require only the empty or nonempty status of the queues in the network.
- $\tilde{\Pi}^{(N)}$: the class of all MER policies within $\Pi_M^{(N)}$ that additionally break ties in favour of inner queues (see condition 2 above).

Note that $\Pi^{(N)} \supseteq \Gamma_M^{(N)} \supseteq \Pi_M^{(N)} \supseteq \tilde{\Pi}^{(N)}$. Going back to our 7-queue example, when $\mathbf{i}(t) = [1, 1, 0, 0, 1, 1, 1]$, policies that choose $\mathbf{s}(t) = [1, 0, 0, 0, 1, 0, 1]$ can be in $\Pi_M^{(7)}$ but not in $\tilde{\Pi}^{(7)}$, while those that choose $\mathbf{s}(t) = [0, 1, 0, 0, 1, 0, 1]$ can be in $\tilde{\Pi}^{(7)}$.

IV. QUEUE LENGTH-AGNOSTIC SCHEDULING

While almost all well-known policies use full queue length-information ($\mathbf{Q}(t)$) to take scheduling decisions, e.g., MaxWeight [1], a key objective of this paper is to show that for a class of interference graphs, throughput-optimal policies can be designed that use much less information. By “queue length-agnostic policies,” we mean those that only require the knowledge of the occupancy vector⁴, i.e., $\mathbf{i}(t)$. Clearly, this contains much less information than the vector $\mathbf{Q}(t)$ that MaxWeight requires, and $\mathbf{i}(t)$ can be transmitted across the network with just 1 bit per queue per slot. The functions $\{\mu_t, t \geq 0\}$ comprising the policy are now maps of the form $\mu_t : \{0, 1\}^N \rightarrow V \subsetneq \{0, 1\}^N$, the set of all activation vectors.

Although it is well-known that fully-connected interference graphs admit throughput-optimal, queue length-agnostic scheduling algorithms (e.g., schedule any nonempty queue), it is not immediately clear how to stabilize other interference graphs with reduced state policies. In fact, Sec. III of [14] provides an excellent example of an MER policy (also called a *maximum-matching* policy) that is *not* throughput-optimal. Moreover, the delay. Throughout the paper, we use total system backlog, $\sum_{i=1}^N Q_i(t)$, as a proxy for delay. properties of such a

⁴We slightly abuse terminology: dependence on $\mathbf{i}(t)$ is not strictly agnostic of backlogs but is a very weak form of dependence.

scheduler are naturally suspect, since even MaxWeight is only known to be asymptotically delay optimal in such networks [15].

We now provide a sufficient condition that will later help construct strongly stable policies that use only $\{\mathbf{i}(t), t \geq 0, \}$, by proving a Lyapunov drift result that will be invoked often in the sequel.

Lemma 1. Consider the class of systems described in Section II, and define property \mathcal{P} as

$$D_i(t) + D_{i+1}(t) = 0 \iff Q_i(t) + Q_{i+1}(t) = 0, \quad (3)$$

for all $t \geq 0$, and for $1 \leq i \leq N - 1$. Any policy that satisfies property \mathcal{P} in every slot t , is throughput-optimal.

Remark IV.1. Note that condition (3) depends only on the reduced state $i(t)$. In words, (3) reads: “for a pair of neighboring queues, there is no departure from either of these queues *iff* both the queues are empty.” For example, with $N = 4$ and $\mathbf{i}(t) = (1, 1, 1, 1)$, $\mathbf{S}(t) = (1, 0, 1, 0)$ satisfies condition (3), but $\mathbf{S}(t) = (1, 0, 0, 1)$ does not.

The proof of Lemma 1 is based on a novel Lyapunov function, treating pairs of adjacent queues as collocated networks. Instead of showing negative drift per queue state, as is common in the analysis of several MaxWeight-style algorithms, we develop an averaging argument to show overall negative drift of the Lyapunov function, and appeal to Neely’s telescoping sum technique [13] to prove strong stability. The proof is deferred to the technical report [16] due to space constraints.

V. PATH-GRAPH INTERFERENCE MODEL WITH $N = 3$: OPTIMAL QUEUE LENGTH-AGNOSTIC SCHEDULING

In this section, we first completely characterize $\Pi_M^{(3)}$ and the subclass $\tilde{\Pi}^{(3)}$, and explore stability and delay optimality for this system. This study will provide some insights into the nature of MER policies in general and, more importantly, in this process, the policies we propose here will act as *building blocks* for policies for larger- N systems. Before we embark on this analysis, we would like to make a few observations about $\Pi^{(3)}$.

Note that with 3 queues, in any given slot t , a policy can choose either $\mathbf{S}(t) = [1, 0, 1]$ which serves queues 1 and 3, or $[0, 1, 0]$ which serves Queue 2. So, a queue length agnostic policy maps every state vector $\mathbf{i}(t)$ to one of these two activation vectors, giving us 256 queue length agnostic policies in all. We prove, however, that upon imposing the MER condition, this number reduces to 4, i.e., $|\Pi_M^{(3)}| = 4$ [16, Sec. V].

Analysis of $\Pi_M^{(3)}$: We now show that this class contains throughput optimal, delay optimal, and also *unstable* MER policies. First, some additional notation is in order. Depending on the mapping from $\mathbf{i}(t)$ to the activation vector, we denote the 4 MER policies $\pi_1^{(3)}, \pi_2^{(3)}, \tilde{\pi}^{(3)}, \pi_4^{(3)}$. The complete descriptions of all these policies are given in Table. I. To begin

Table I: Comparison of $\mathbf{S}(t)$ under $\pi_1^{(3)}, \pi_2^{(3)}, \tilde{\pi}^{(3)}$ and $\pi_4^{(3)}$

$[i_1(t), i_2(t), i_3(t)]$	$\pi_1^{(3)}$	$\pi_2^{(3)}$	$\tilde{\pi}^{(3)}$	$\pi_4^{(3)}$
000	101	101	101	101
001	101	101	101	101
010	010	010	010	010
011	010	101	010	101
100	101	101	101	101
101	101	101	101	101
110	101	010	010	101
111	101	101	101	101

with, we show that $\pi_1^{(3)}$ and $\pi_2^{(3)}$ are T.O. Both these policies will later be used as building blocks to construct T.O. policies for larger systems and are therefore very important to our study.

A. Analysis of $\pi_1^{(3)}$ and $\pi_2^{(3)}$

Theorem 2. $\pi_1^{(3)}$ and $\pi_2^{(3)}$ are both throughput-optimal.

In words, the policy $\pi_1^{(3)}$ simply reads “*prioritize Queue 1 over Queue 2, and Queue 2 over Queue 3, while scheduling all possible non-interfering queues.*” The proof of Theorem 2 uses the fact that under $\pi_1^{(3)}$, queues 1 and 2 form a priority queueing system and are stable. We then show that Queue 3 is served “sufficiently often” to ensure stability. $\pi_2^{(3)}$ simply swaps the priorities of Queues 1 and 3 and its proof proceeds mutatis mutandis. The complete proof is available in [16] due to space constraints here.

B. Analysis of $\tilde{\pi}^{(3)}$

This policy can be restated as follows. At time t :

- 1) If $Q_1(t) > 0$ and $Q_3(t) > 0$, choose $\mathbf{S}(t) = [1, 0, 1]$.
- 2) Else, if $Q_2(t) > 0$, choose $[0, 1, 0]$.
- 3) Else choose $[1, 0, 1]$.

In [6], it has been asserted without formal proof that $\tilde{\pi}^{(3)}$ is sum-queue optimal. We begin analysing the policy by proving that it is T.O.

Theorem 3. $\tilde{\pi}^{(3)}$ is throughput-optimal.

The proof of this result involves showing that $\tilde{\pi}^{(3)}$ satisfies property \mathcal{P} in Lem. 1 and is therefore T.O. The proof is available in [16].

We next turn to the delay performance of the policy $\tilde{\pi}^{(3)}$. Tassiulas and Ephremides [6, Theorem 4.2] define a projection operator $L : \Pi^{(N)} \rightarrow \Gamma_M^{(N)}$ that takes any policy $\pi \in \Pi^{(N)}$ and produces an MER policy, $L(\pi)$. They then show that each policy π is dominated by its “MER-improvement” policy $L(\pi)$ in the sense of lower sum-queue lengths. Specifically, if $\mathbf{Q}^\pi(t)$ denotes the backlog induced by some policy π , then Theorem 4.2 in [6] shows that when the systems upon which

π and $L(\pi)$ are started out in the same initial state and the arrivals have the same statistics, then

$$\sum_{i=1}^N Q_i^{L(\pi)}(t) \leq^{st} \sum_{i=1}^N Q_i^{\pi}(t), \quad \forall t \geq 0, \quad (4)$$

where st denotes stochastic ordering.

It has also been asserted in [6, Remark 2, pp. 353] without formal proof that the policy $\tilde{\pi}^{(3)}$ is sum-queue- or delay-optimal. We prove the following result about the delay-optimality of this policy.

Theorem 4. For any policy $\pi \in \Pi^{(3)}$, let the system backlog vector at time t be denoted by $\mathbf{Q}^{\pi}(t)$ and the backlog with $\tilde{\pi}^{(3)}$ be denoted by $\mathbf{Q}^{\tilde{\pi}^{(3)}}(t)$. Also let $\mathbf{Q}^{\pi}(0) = \mathbf{Q}^{\tilde{\pi}^{(3)}}(0)$. Then,

$$\sum_{i=1}^3 Q_i^{\tilde{\pi}^{(3)}}(t) \leq^{st} \sum_{i=1}^3 Q_i^{\pi}(t), \quad \forall t \geq 0, \quad (5)$$

where “ st ” denotes stochastic ordering.

The proof technique is essentially the same as that of Theorem 4.2 in [6], except that we make the observation that a key step in that proof has more general applicability. It involves constructing a sequence of policies each of which shows better delay than its predecessor and than a general policy π . The limit of this sequence of policies is then shown to uniquely be $\tilde{\pi}^{(3)}$. The proof is deferred to the technical report [16] owing to space constraints.

C. Analysis of $\pi_4^{(3)}$

This policy prioritizes the outer queues and can be restated as follows.

At time t

- 1) If either $Q_1(t) > 0$ or $Q_3(t) > 0$, choose $(1, 0, 1)$.
- 2) Else choose $(0, 1, 0)$.

It turns out, analogous to the observation by McKeown et al [14] that this MER policy is, in fact, not throughput-optimal.

Proposition 5 (An MER but not throughput-optimal policy). $\pi_4^{(3)}$ is *not* throughput-optimal.

The proof of this result involves constructing an arrival rate vector for which the offered service rate to one of the queues is strictly smaller than the arrival rate. It is available in the technical report [16]. This completes the characterization of $\Pi_M^{(3)}$.

D. Policies outside $\Pi_M^{(3)}$

We now propose and analyse a policy that we denote $\pi_5^{(3)}$, and show the rather surprising result that it is T-O. despite not being MER. This policy will become important later, as a *fundamental building block* while constructing policies for larger systems.

At time t

- 1) If $Q_2(t) > 0$ choose $\mathbf{S}(t) = [0, 1, 0]$,

- 2) Else choose $\mathbf{s}(t) = [1, 0, 1]$.

Since $\mathbf{i}(t) = [1, 1, 1] \mapsto [0, 1, 0]$, this policy is not MER. However, we have

Proposition 6 (A non-MER but throughput-optimal policy). $\pi_5^{(3)}$ is throughput-optimal.

Proof: The key tool behind the proof of this result is the throughput-optimality Lem. 1. It is easily checked that $\pi_5^{(3)}$ satisfies property \mathcal{P} in every slot and thus, by Lemma 1, is throughput-optimal. ■

In the sections to follow, we will first use the policies proposed here to come up with stabilizing and delay optimal policies for larger systems, and then show how to extend the theory and scheduling algorithms to other interference graphs.

VI. EXTENSIONS TO PATH GRAPH INTERFERENCE MODELS WITH $N > 3$

We will now use the policies developed in Sec. V as building blocks to construct throughput and delay optimal policies for larger systems, specifically for $N = 4$ and 5. Though some of the policies we analyze have been proposed earlier, we identify several new T-O. policies here. We will also study the delay properties of the policies developed for both systems, and eventually show an impossibility result about obtaining uniform delay or sum-queue optimality.

A. Systems with $N = 4$ queues

Continuing along the same lines as Sec. V, with 4 queues, we have three activation vectors to choose from in any given slot, viz., $[1, 0, 1, 0]$, $[1, 0, 0, 1]$ and $[0, 1, 0, 1]$, which gives us $3^{2^4} > 43 \times 10^6$ policies that take only the occupancy vector $\mathbf{i}(t)$ as input. This reduces to 96 policies once the MER condition is imposed and the explanation for the reduction is similar to the case with 3 queues. But since this number is also inordinately large, we will restrict our study to $\bar{\Pi}^{(4)}$, which contains 4 policies, which we denote by $\{\tilde{\pi}_i^{(4)}, 1 \leq i \leq 4\}$. In what follows, we provide a complete characterization of $\bar{\Pi}^{(4)}$.

1) *Analysis of $\tilde{\pi}_1^{(4)}$ and $\tilde{\pi}_2^{(4)}$:* The policy $\tilde{\pi}_1^{(4)}$ can be stated as follows. At each time t :

- 1) If $\mathbf{i}(t) = [1, 1, 1, 0]$, $\mathbf{S}(t) = [1, 0, 1, 0]$.
- 2) Else, if $Q_2(t) > 0$, $\mathbf{S}(t) = [0, 1, 0, 1]$.
- 3) Else, if $Q_3(t) > 0$, $\mathbf{S}(t) = [1, 0, 1, 0]$.
- 4) Else, $\mathbf{S}(t) = [1, 0, 0, 1]$.

The first step of $\tilde{\pi}_2^{(4)}$ involves choosing $\mathbf{S}(t) = [0, 1, 0, 1]$ when $\mathbf{i}(t) = [0, 1, 1, 1]$. In the next steps $\tilde{\pi}_2^{(4)}$ reverses the priorities of queues 2 and 3.

Remark VI.1. We note that while both these policies have been proposed by Ji et al [7], only an informal argument regarding their stability properties has been provided therein, followed by a study of their performance under heavy traffic. Their informal argument asserts that the fraction of time for which Queue 2 is nonempty *equals* its arrival rate λ_2 , and this claim is crucial to their stability argument.

By Little’s Theorem applied to the HOL position, this assertion holds only if the mean waiting time in the HOL

position of Queue 2 is exactly 1 slot. The actual fraction of time for which Queue 2 is nonempty converges to $\lambda_2 \cdot \mathbb{E}B_2$, where $\mathbb{E}B_2$ is the mean service time of packets in Queue 2. Since Queue 2 is not always served whenever it is nonempty, $\mathbb{E}B_2 > 1$ (strict inequality), so the fraction of time left to offer service to Queue 1 is strictly smaller than $1 - \lambda_2$. Moreover, as can be seen from the definition of $\tilde{\pi}_1^{(4)}$, $\mathbf{S}(t)$ is not necessarily $[1, 0, 1, 0]$ whenever queues 1 and 3 are nonempty (take $\mathbf{i}(t) = [1, 1, 1, 1]$ for instance). It is, therefore, unclear whether Queue 1 is offered service often enough to stabilize it. We here provide a formal proof of the throughput-optimality of $\tilde{\pi}_1^{(4)}$ and $\tilde{\pi}_2^{(4)}$.

Proposition 7. $\tilde{\pi}_1^{(4)}$ and $\tilde{\pi}_2^{(4)}$ are throughput-optimal.

To prove the throughput-optimality of $\tilde{\pi}_1^{(4)}$, we first combine $\pi_1^{(3)}$ and $\pi_5^{(3)}$ to form a policy $\pi_1^{(4)}$ which is *not* MER and show that it is T.O. We then use the operator L to project $\pi_1^{(4)}$ onto $\Pi_M^{(4)}$ to get $\tilde{\pi}_1^{(4)}$ and complete the proof using a sample path-wise stochastic dominance argument (of the sort in Eqn. (4)). A similar proof holds for $\tilde{\pi}_2^{(4)}$ as well. We denote the non-MER counterpart of $\pi_1^{(4)}$ designed for $\tilde{\pi}_2^{(4)}$ by $\pi_2^{(4)}$. The complete proof is available in the technical report [16].

Remark VI.2. While $\pi_1^{(3)}$ is a policy that prioritizes “outer” queues (Queue 1), $\pi_5^{(3)}$ prioritizes “inner” queues, i.e., Queue 2. Since, in the four queue system, Queue 2 becomes the outer and inner queue for these two policies respectively, both subsystems are simultaneously stabilized. This theme of stitching together policies designed for smaller systems to construct policies for larger systems will recur through the rest of this article.

2) *Analysis of $\tilde{\pi}_3^{(4)}$ and $\tilde{\pi}_4^{(4)}$:* $\tilde{\pi}_3^{(4)}$ differs from $\tilde{\pi}_1^{(4)}$ only when $\mathbf{i}(t) = [1, 1, 1, 1]$ and $\tilde{\pi}_3^{(4)}$ serves queues 1 and 3 while $\tilde{\pi}_1^{(4)}$ serves 2 and 4. Similarly with $\tilde{\pi}_2^{(4)}$ and $\tilde{\pi}_4^{(4)}$.

Proposition 8. $\tilde{\pi}_3^{(4)}$ and $\tilde{\pi}_4^{(4)}$ are throughput-optimal.

The idea behind the proof of this result involves showing that the total system backlogs under $\tilde{\pi}_1^{(4)}$ and $\tilde{\pi}_3^{(4)}$ are the same at all times when started in the same initial conditions. Similarly with $\tilde{\pi}_2^{(4)}$ and $\tilde{\pi}_4^{(4)}$. Since $\tilde{\pi}_1^{(4)}$ and $\tilde{\pi}_2^{(4)}$ are T.O., the other 2 policies are, as well. The proof is available in [16].

B. Systems with $N = 5$ queues: Analysis of $\tilde{\pi}^{(5)}$

With 5 queues, Queue 3 is the “most constrained” queue (informally, the one that is offered service in fewer activation vectors than other queues) and hence, the proposed policy will prioritize it. The policy $\tilde{\pi}^{(5)} \in \tilde{\Pi}^{(5)}$ is stated as follows.

At time t ,

- 1) If $\mathbf{i}(t) = [0, 1, 1, 1, 0]$, $\mathbf{S}(t) = [0, 1, 0, 1, 0]$.
- 2) If $\mathbf{i}(t) = [1, 1, 1, 1, 0]$ or $[0, 1, 1, 1, 1]$, $\mathbf{S}(t) = [0, 1, 0, 1, 0]$.
- 3) Else, if $Q_3(t) > 0$, $\mathbf{S}(t) = [1, 0, 1, 0, 1]$.
- 4) Else, if $Q_2(t) > 0$ and $Q_4(t) > 0$, $\mathbf{S}(t) = [0, 1, 0, 1, 0]$.
- 5) Else, if $Q_2(t) > 0$ or $Q_4(t) > 0$,
 - a) $\mathbf{S}(t) = [0, 1, 0, 0, 1]$ if $Q_2(t) > 0$ and

b) $\mathbf{S}(t) = [1, 0, 0, 1, 0]$ if $Q_4(t) > 0$.

6) Else, $\mathbf{S}(t) = [1, 0, 1, 0, 1]$.

Proposition 9. $\tilde{\pi}^{(5)}$ is throughput-optimal.

Proof: Our analysis of this policy begins with a non delay optimal but MER version of $\tilde{\pi}^{(5)}$, which we call $\pi_M^{(5)}$. We establish the throughput-optimality of $\pi_M^{(5)}$ using a non MER policy, $\pi^{(5)}$, projecting $\pi^{(5)}$ onto $\Pi_M^{(5)}$ to get $\pi_M^{(5)}$ and using the fact that L cannot worsen sum queue lengths (Eqn (4)). Finally, we show that $\tilde{\pi}^{(5)}$ is T.O. by observing that it satisfies the 2^{nd} MER condition in Sec. III, and proving that this ensures that the sum queue length of $\tilde{\pi}^{(5)}$ is never (stochastically) larger than that of $\pi_M^{(5)}$. See [16] for details. ■

C. Analysis of Delay in $\Pi^{(4)}$ and $\Pi^{(5)}$

We now show that $\Pi^{(4)}$ **does not contain** any queue length agnostic policy that is *uniformly* delay optimal over the entire set Λ_4^o . This is unlike the case with $N = 3$, where $\tilde{\pi}_4^{(3)}$ produced the lowest possible delay regardless of the arrival rate. We first prove in Prop. 19 in [16] that $\tilde{\Pi}^{(4)}$ does not contain any uniformly delay optimal policy. Next, in Prop. 11 in [16] we show that policies in $\tilde{\Pi}^{(4)}$ show better delay performance than those in $\Pi_M^{(4)}$. We already know (Eqn.(4)) that the delay of any policy in $\Pi^{(N)}$ can be improved by projecting it onto $\Pi_M^{(N)}$. Now, Prop. 11 in [16], along with Eqn.(4), shows that delay optimal policies, when they exist, must necessarily lie in $\tilde{\Pi}^{(N)}$. This observation, along with the nonexistence of delay optimal policies in $\tilde{\Pi}^{(4)}$ prove that

Theorem 10. For all $N \geq 4$, there does not exist any policy in $\Pi_M^{(N)}$ that is uniformly delay optimal over all of Λ_N^o .

Proof: See Thm. 12 in [16]. ■

So in essence, Thm. 10 shows that while *throughput* optimality in these interference graphs only requires knowledge of queue occupancy (i.e., $\mathbf{i}(t)$), *delay* optimality could require more information from the history \mathcal{H}_t . We conclude this section with a quick observation about $\Pi_M^{(5)}$.

Proposition 11. For every policy $\pi \in \Pi_M^{(5)} \setminus \tilde{\Pi}^{(5)}$, there exists a policy $\pi' \in \tilde{\Pi}^{(5)}$ such that

$$\sum_{i=1}^5 Q_i^{\pi'}(t) \stackrel{st}{\leq} \sum_{i=1}^5 Q_i^{\pi}(t), \quad \forall t \geq 0. \quad (6)$$

Proof: This proof uses the fact that while $\pi_M^{(5)}$ is only MER, $\tilde{\pi}^{(5)}$ also satisfies the 2^{nd} MER condition. See [16] for details. ■

The systems studied until now were represented by *Path graphs*, i.e., their interference graphs were straight lines. We now extend the theory developed, to more general classes of graphs that better represent sensor networks.

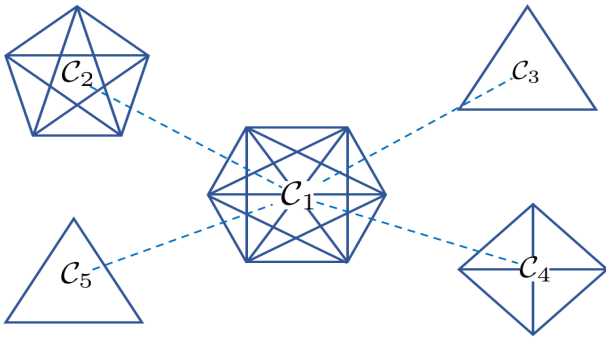


Figure 3: Interference graph corresponding to the network in Fig. 2. A dotted line connecting cliques \mathcal{C}_i and \mathcal{C}_j means that transmissions in the two cliques cannot take place simultaneously.

VII. BEYOND PATH INTERFERENCE GRAPHS

In this section, we extend the results obtained thus far to a non-trivial class of networks whose interference graphs are not necessarily Path graphs.

A. The Star-of-Cliques Interference Model

Consider an interference graph consisting of a central fully connected subgraph surrounded by $N - 1$ fully connected subgraphs (see Fig. 3). In the remainder of this section, we will call each of these fully connected subgraphs “cliques.” The graph under consideration hence consists of N cliques denoted $\mathcal{C}_1, \dots, \mathcal{C}_N$ and clique \mathcal{C}_i consists of \mathcal{N}_i vertices. *Cliques can have arbitrarily many queues.* Transmissions in \mathcal{C}_1 interfere with those in all other cliques while the transmissions in $\mathcal{C}_i, i \geq 2$ interfere with those in \mathcal{C}_1 only. Such a situation could arise in an in-building wireless sensor network comprising clusters of collocated sensors, each with its own sink, all operating on the same channel, and located such that all the peripherally located clusters interfere with a centrally located one, but not with each other.

Our aim is to propose queue length-agnostic throughput-optimal policies for such graphs. In the literature, such network topologies are a special case of what are known as “Snowflake” topologies [10, Chapters 5 and 6]. In studying these networks we are, in fact, providing natural extensions to our earlier work [17] in which we studied mean delay optimal scheduling in collocated networks using only the occupancy status of the queues.

From the preceding discussion, it is now clear how Fig. 3 is the interference graph associated with the network in Fig. 2. As before, a queue is associated with each vertex and Queue j in Clique i is fed by an exogenous Bernoulli packet arrival process of rate $\lambda_{i,j}$, i.e., $P(A_{i,j}(t) = 1) = \lambda_{i,j}, \forall t \geq 1$. At time $t+$, $Q_{i,j}(t)$ is the backlog of Queue j in \mathcal{C}_i . The queue length, arrival and departure processes are embedded as in Sec. II. Let $\mathcal{N} \equiv \sum_{i=1}^N \mathcal{N}_i$ denote the total number of queues in the system.

Remark VII.1. Notice that the same description also models the path graph with $N = 3$ with iid *batch* arrivals. Setting $\mathcal{N} = 3$ in the above description, we see that the \mathcal{N}_i arrivals processes

to *Clique* i can be treated as a single iid batch arrival process to a single queue in each clique. Hence, the policies that we present in the sequel can be used to schedule transmissions in systems with batch arrivals, generalizing the system model presented in Sec. II.

B. Scheduling Policies

We will now show that some of the scheduling policies developed for path graphs extend in a natural manner to policies for star-of-cliques graphs.

Consider the following queue-length agnostic scheduling policy $\tilde{\phi}$, an extension of the 3-node path graph policy $\tilde{\pi}^{(3)}$.

At each time t :

- 1) If $\sum_{m=2}^N (\sum_{l \in \mathcal{C}_m} i_l(t)) > 0$ serve any nonempty queue in every clique $\{\mathcal{C}_m, m \geq 2\}$ having nonempty queues.
- 2) Else, if $\sum_{l \in \mathcal{C}_1} i_l(t) > 0$, serve any nonempty queue in \mathcal{C}_1 .
- 3) Else, serve one nonempty queue (if it exists) in each of $\{\mathcal{C}_m, m \geq 2\}$.

Proposition 12. $\tilde{\phi}$ is throughput-optimal.

The main idea behind the proof of this proposition is to prove a more general version of Property \mathcal{P} that can be applied to *cliques* of arbitrary sizes rather than just queues and use Lem. 1 to prove strong stability. See [16] for details.

Extending another policy, $\pi_5^{(3)}$ (see Sec. V-D), gives us a second queue length-agnostic policy ϕ_5 for this system⁵. We define the policy as follows.

At time t ,

- 1) If $\sum_{l \in \mathcal{C}_1} i_l(t) > 0$, serve any nonempty queue in \mathcal{C}_1 .
- 2) Else, serve one nonempty queue (if it exists) in each of $\{\mathcal{C}_m, m \geq 2\}$.

Proposition 13. ϕ_5 is throughput-optimal.

Once again, the proof of this result rests on proving the new version of Property \mathcal{P} for this policy, followed by Lyapunov analysis. The reader is referred to the technical report [16] for the complete details.

We end this section with some remarks about delay performance and implementation. Prop. 20 in the technical report [16] proves that $\tilde{\phi}$ cannot be worse in delay (sum-queue length) than ϕ_5 . In Sec. IX, we will provide some preliminary ideas about how ϕ_5 and $\tilde{\phi}$ can be implemented in a decentralized fashion with very little (if any) information exchange between queues.

VIII. NUMERICAL RESULTS

In this section we numerically compare the performance of the various policies and protocols we have proposed. Recall that Thm. 10 in Sec. VI asserts that $\tilde{\Pi}^{(N)}$ cannot contain policies which are uniformly delay optimal in systems for $N \geq 4$. Table II, shows the results of simulating $\tilde{\pi}_1^{(4)}$ and $\tilde{\pi}_2^{(4)}$ and comparison with our benchmark policies MaxWeight (column 3) and $L(MW\alpha)$ (column 4). $L(MW\alpha)$ is an MER

⁵Note that the subscript 5 is for consistency; there are no ϕ_1, \dots, ϕ_4 .

Table II: Path graph interference model $N = 4$. Comparison of sum queue length under the proposed $\tilde{\pi}_1^{(4)}$, $\tilde{\pi}_2^{(4)}$ policies, MW and $L(MW\alpha)$, $\alpha = 0.01$.

λ	$\tilde{\pi}_1^{(4)}$	$\tilde{\pi}_2^{(4)}$	MW	$L(MW\alpha)$
[0.2, 0.79, 0.2, 0]	23.060	32.711	32.912	29.478
[0 , 0.049, 0.95, 0.049]	89.741	60.234	87.516	70.579
[0.49, 0.49, 0.49, 0.49]	45.963	45.924	57.302	43.508

Table III: Path graph interference model $N = 5$. Comparison of sum queue length under the proposed $\tilde{\pi}^{(5)}$ policies, MW and $L(MW\alpha)$, $\alpha = 0.01$

λ	$\tilde{\pi}^{(5)}$	MW	$L(MW\alpha)$
[0.15, 0.049 , 0.95 , 0.049 , 0.15]	61.537	88.243	75.642
[0.049, 0.95 , 0.049 , 0.95 , 0.049]	128.842	176.688	129.358

policy, obtained by using the operator L (see Sec. V-C) to project a modification of MaxWeight (MW) called $MW\alpha$ onto $\Gamma_M^{(4)}$. The $MW\alpha$ policy, studied in [18] and [11], is essentially MW with all queue lengths raised to their α -th powers, with $\alpha > 0$. This policy has been observed to show smaller sum queue lengths (than MW) with smaller α [19]. We do not simulate $\tilde{\pi}_3^{(4)}$ and $\tilde{\pi}_4^{(4)}$ since they have the same sum queue length performance as $\tilde{\pi}_1^{(4)}$ and $\tilde{\pi}_2^{(4)}$, respectively (see proof of Prop. 8).

As the first two rows of table show, different policies perform better for different arrival rates. Finally, Row 3 of the table shows an arrival rate vector for which neither of the queue length-agnostic policies does well and $L(MW\alpha)$ shows the smallest sum queue length. In a similar manner Table III shows that our proposed policy $\tilde{\pi}^{(5)}$ performs much better than both MW and $L(MW\alpha)$ under a range of arrival rates.

We move on to simulations of the policies proposed for *non path* networks. The chosen network, shown in Fig. 4, consists of 4 cliques and a total of 6 queues. Table IV shows the result of simulating $\tilde{\phi}$, ϕ_5 and MW on this network. Observe that whether the central clique, i.e., \mathcal{C}_1 is the most heavily loaded (row 1 in the table) or the least loaded (row 2), $\tilde{\phi}$ performs the best among the policies tested. This is interesting, since one expects that situations may arise wherein only two of the three peripheral cliques and \mathcal{C}_1 are nonempty. In such a case, $\tilde{\phi}$ would serve \mathcal{C}_1 , giving up the chance to serve both the peripheral nonempty cliques simultaneously and remove 2 packets from the system in a single slot, which is what

Table IV: Comparison of sum queue length under the proposed $\tilde{\phi}$, ϕ_5 policies, and MW acting on the network in Fig. 4.

λ	$\tilde{\phi}$	ϕ_5	MW
[0.1, 0.1, 0.1, 0.65 , 0.3, 0.3]	10.539	13.846	14.506
[0.3, 0.3, 0.3, 0.09 , 0.9, 0.9]	45.535	53.766	57.861

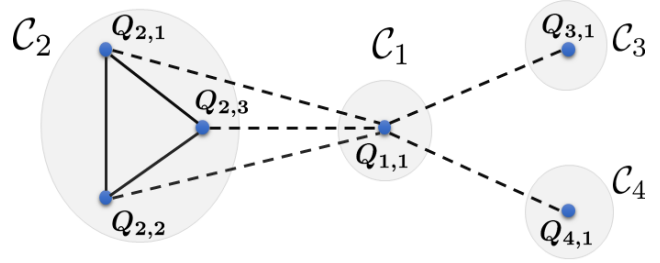


Figure 4: The network used to study the performance of ϕ_5 and $\tilde{\phi}$.

MW might have attempted, if the queues therein were large enough. If, for example, in some slot t , \mathcal{C}_2 is empty, while $Q_{1,1}(t) = 1$, $Q_{3,1}(t) = 5$ and $Q_{4,1}(t) = 2$, $\tilde{\phi}$ still serves only $Q_{1,1}$ (1 packet transmitted) while MaxWeight serves both $Q_{3,1}$ and $Q_{4,1}$ (2 packets transmitted). Why $\tilde{\phi}$ still performs better requires more investigation and will be a focus of our future work.

IX. SOME REMARKS ON DISTRIBUTED IMPLEMENTATION

We now provide some preliminary ideas on how the policies developed in Sec. VII-B can be used to design MAC protocols suited to sensor networks possessing the structure in Fig. 2. Our ultimate aim is to develop *decentralized* scheduling protocols, that require little explicit exchange of state information or control signals. The policies developed in this paper until now, seem particularly suited to distributed implementation since they rely only on occupancy information (reduced state information) rather than queue lengths. To facilitate distributed implementation, we first describe the following transmission sensing mechanism.

Transmission sensing: We assume that there exist activity-sensing intervals at the beginning of every slot called **minislots** [20]. These are used to sense whether or not a queue scheduled to transmit is empty-nonempty, by averaging power over the minislot (in a manner similar to the clear channel assessment or CCA mechanism [21]).

To begin with, consider a clique, say \mathcal{C}_1 , in isolation. Suppose the nodes in \mathcal{C}_1 could determine the backlog of a node in the clique each time it transmitted a packet⁶ and keep track of the number of slots since node i was allowed to transmit. Then, at the beginning of slot t , the information common to all nodes in \mathcal{C}_i would consist of the number of slots $V_i(t)$ since node i last transmitted⁷ and its backlog $Q_i(t - V_i(t))$ at that instant. With this partial information structure, it has been proved in [17] that *exhaustively* serving a nonempty queue minimizes delay.

This is encouraging, since, with exhaustive service, $Q_i(t - V_i(t))$ is always 0, which obviates the need to transmit queue lengths. When the queue under service, called the *incumbent* in the sequel, becomes empty the next queue to be scheduled

⁶The backlog information could be quantized and contained in the packet header, for example.

⁷If the node were empty at this instant, it wouldn't have actually transmitted anything. The others can infer its "emptiness" by sensing no power in a minislot.

for transmission can be determined by another result from [17]. It is shown therein that for this information structure (i.e., all queues in the clique keeping track of $\mathbf{V}(t)$) that scheduling node $\arg \max_{j \in \mathcal{C}_1} V_j(t)$ is throughput-optimal, and under certain conditions, also mean delay optimal.

Now, returning to the star of cliques model, one could imagine an extension of policy ϕ_5 , in which clique \mathcal{C}_i uses minislots to determine if the incumbent queue is empty. From the interference model, we already know that nodes in \mathcal{C}_1 and $\mathcal{C}_j, j \geq 2$ (the peripheral cliques) can hear all transmissions in the two cliques but for all $2 \leq j, l \leq N$, nodes in \mathcal{C}_j cannot hear the transmissions in \mathcal{C}_l and vice versa. Following ϕ_5 , the first minislot could be dedicated to determining whether the incumbent of \mathcal{C}_1 is empty or not. If it is not, this node is allowed to transmit, following the exhaustive service result in the discussion above. If no power is sensed in this minislot, a new queue $\arg \max_{j \in \mathcal{C}_1} V_j(t)$ is chosen (and served until empty). If, however, this queue is also empty (sensed by lack of power in a *second* minislot), each peripheral clique $\mathcal{C}_j, j \geq 2$ allows its incumbent to transmit, and if that queue is empty (sensed by lack of power in a *third* minislot), $\arg \max_{j \in \mathcal{C}_i} V_j(t)$ is allowed to transmit. If this queue is empty, the slot is wasted.

Using minislots to sense transmissions (or lack of transmissions) will, thus, allow us to implement the policies we have derived for the star-of-cliques network *without* the occupancy vector having to be shared across the network. The stability and delay properties of this protocol need to be studied in detail and will form part of our future work. Observe, however, that the number of minislots *does not* scale with the number of queues in the system, which is encouraging.

X. CONCLUSION AND FUTURE WORK

In this paper, in the setting of the classical work of Tassioulas and Ephremides [1] we have studied the possibility of throughput optimal and low delay scheduling, with just the knowledge of the empty or non-empty state of each queue in the network. The necessity for such policies naturally arises in low-power IoT networks where disseminating queue length information across the network consumes too much energy. We have demonstrated in the setting of path interference graphs, with iid arrivals to queues such reduced state schedulers do exist. We have shown that even with limited state information these policies attain stability and low sum queue length (and hence, delay). We have also shown that there cannot exist a uniformly delay optimal policy that uses only occupancy vectors for scheduling for $N \geq 4$.

Motivated by these policies, we then proposed and analysed scheduling policies for wireless sensor networks possessing the star-of-cliques structure that show good stability and delay properties while requiring only occupancy information. This is a natural extension of our earlier work [17] in which we studied mean delay optimal scheduling in collocated networks using only the occupancy status of the queues. Our present study has thrown up several questions like the performance of these policies under heavy traffic and stabilizing path graphs for general N , which will form part of our future

work. We also hope that our study opens the doors for more detailed investigation of the proposed reduced-state algorithms and protocols, towards the utilization of this theory for the development of decentralized medium access protocols.

REFERENCES

- [1] L. Tassioulas 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] S. Sanghavi, D. Shah, and A. S. Willsky, "Message passing for maximum weight independent set," *IEEE Transactions on Information Theory*, vol. 55, no. 11, pp. 4822–4834, 2009.
- [3] 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.
- [4] L. Jiang and J. Walrand, "Approaching throughput-optimality in distributed csma scheduling algorithms with collisions," *Networking, IEEE/ACM Transactions on*, vol. 19, no. 3, pp. 816–829, 2011.
- [5] S. Rajagopalan, D. Shah, and J. Shin, "Network adiabatic theorem: an efficient randomized protocol for contention resolution," in *ACM SIGMETRICS Performance Evaluation Review*, vol. 37, no. 1. ACM, 2009, pp. 133–144.
- [6] L. Tassioulas and A. Ephremides, "Dynamic scheduling for minimum delay in tandem and parallel constrained queueing models," *Annals of Operations Research*, vol. 48, no. 4, pp. 333–355, 1994.
- [7] T. Ji, E. Athanasopoulou, and R. Srikant, "On optimal scheduling algorithms for small generalized switches," *IEEE/ACM Transactions on Networking*, vol. 18, no. 5, pp. 1585–1598, 2010.
- [8] A. Taghavi, A. Vem, J.-F. Chamberland, and K. R. Narayanan, "On the design of universal schemes for massive uncoordinated multiple access," in *Information Theory (ISIT), 2016 IEEE International Symposium on*. IEEE, 2016, pp. 345–349.
- [9] K. R. Narayanan and H. D. Pfister, "Iterative collision resolution for slotted aloha: An optimal uncoordinated transmission policy," in *Turbo Codes and Iterative Information Processing (ISTC), 2012 7th International Symposium on*. IEEE, 2012, pp. 136–139.
- [10] S. Yasukawa, A. Farrel, and O. Komolafe, "An analysis of scaling issues in mpls-te core networks," Tech. Rep. RFC 5439, February, 2009.
- [11] D. Shah *et al.*, "Heavy traffic analysis of optimal scheduling algorithms for switched networks," *Annals of Applied Probability*, 2007.
- [12] A. Kumar, D. Manjunath, and J. Kuri, *Wireless networking*. Morgan Kaufmann, 2008.
- [13] 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.
- [14] N. McKeown, A. Mekkittikul, V. Anantharam, and J. Walrand, "Achieving 100% throughput in an input-queued switch," *IEEE Transactions on Communications*, vol. 47, no. 8, pp. 1260–1267, 1999.
- [15] S. T. Maguluri, R. Srikant, and L. Ying, "Heavy traffic optimal resource allocation algorithms for cloud computing clusters," *Arxiv preprint arXiv:1206.1264*, 2012.
- [16] A. Mohan, A. Gopalan, and A. Kumar, "Reduced-state, optimal medium access control for wireless data collection networks," Tech. Rep., 2017. Link: <https://1drv.ms/b/s!AnNnKfWPpFPjBbRkgLrk2iREPbBQ>.
- [17] A. Mohan, A. Chattopadhyay, and A. Kumar, "Hybrid MAC protocols for low-delay scheduling," in *Mobile Ad Hoc and Sensor Systems (MASS), 2016 IEEE 13th International Conference on*. IEEE, 2016, pp. 47–55.
- [18] A. L. Stolyar *et al.*, "Maxweight scheduling in a generalized switch: State space collapse and workload minimization in heavy traffic," *The Annals of Applied Probability*, vol. 14, no. 1, pp. 1–53, 2004.
- [19] I. Keslassy and N. McKeown, "Analysis of scheduling algorithms that provide 100% throughput in input-queued switches," in *Proceedings of the Annual Allerton Conference on Communication Control and Computing*, vol. 39, no. 1. The University; 1998, 2001, pp. 593–602.
- [20] I. Rhee, A. Warriar, M. Aia, J. Min, and M. L. Sichitiu, "Z-MAC: a hybrid MAC for wireless sensor networks," *IEEE/ACM Transactions on Networking (TON)*, vol. 16, no. 3, pp. 511–524, 2008.
- [21] P. Kinney, "The 802.15.4 CCA method," *Project: IEEE P802.15 Working Group for Wireless Personal Area Networks (WPANs)*, Submitted: Nov 14, 2001.